МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАКЛАД ВИЩОЇ ОСВІТИ
«ПОДІЛЬСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ»
ФАКУЛЬТЕТ ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій,
фізико-математичних та безпекових дисциплін

Курс лекцій

з навчальної дисципліни:
«Теорія інформаційних систем»
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю 126 «Інформаційні системи та технології»

Кам’янець-Подільський
2025

УДК: 004.02

Укладач:

МУШЕНИК Ірина Миколаївна

кандидат економічних наук, доцент, доцент кафедри інформаційних технологій, фізико-математичних та безпекових дисциплін.

Рекомендовано до друку науково-методичною радою
Закладу вищої освіти «Подільський державний університет»
протокол № 2 від 27 березня 2025 р.

Рецензенти:

ЗЕЛЕНСЬКИЙ Олексій Віталійович,
кандидат фізико-математичних наук, доцент кафедри математики Кам´янець-Подільського національного університету імені Івана Огієнка.

ПАНЦИР Юрій Іванович
декан інженерно-технічного факультету Закладу вищої освіти «Подільський державний університет», кандидат технічних наук, доцент, доцент кафедри електротехніки, електромеханіки і електротехнологій.

М 50

Курс лекцій з навчальної дисципліни «Теорія інформаційних систем» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 126 «Інформаційні системи та технології» / І. М. МУШЕНИК. Кам'янець-Подільський: ЗВО «ПДУ», 2025. 254 с.

Курс лекцій охоплює такі важливі теми, як загальні відомості про інформаційні системи, їхні основні компоненти, моделювання та аналіз інформаційних процесів, управління базами даних, принципи кібербезпеки та адміністрування ІС. Це дозволяє здобувачам набувати навичок проектування, розробки та оцінки ефективності інформаційних систем, що є ключовими в сучасних технологічних процесах. Вивчення основ алгоритмізації, принципів функціонування комп'ютерних мереж та систем підтримки прийняття рішень також сприяє розвитку аналітичного мислення та здатності вирішувати задачі, пов'язані з оптимізацією інформаційних потоків, автоматизацією бізнес-процесів та впровадженням новітніх інформаційних технологій.

ЗМІСТ

ВСТУП

Курс "Теорія інформаційних систем" спрямований на формування у студентів ґрунтовних знань щодо принципів побудови та функціонування сучасних інформаційних систем. В умовах швидкого розвитку цифрових технологій інформаційні системи відіграють ключову роль у різних сферах діяльності, зокрема в бізнесі, науці, освіті та державному управлінні. Даний курс допоможе студентам зрозуміти основи інформаційних процесів, моделювання даних, принципи кібербезпеки та адміністрування ІС.

Окрім теоретичних знань, здобувачі набудуть практичних навичок аналізу, розробки та оптимізації інформаційних систем, що дозволить їм ефективно працювати з базами даних, інформаційними потоками та програмним забезпеченням. Особлива увага приділяється питанням управління ІС, оцінці їх ефективності та розгляду новітніх технологій, таких як хмарні обчислення, штучний інтелект і блокчейн.

Завдяки цьому курсу здобувачі зможуть глибше зрозуміти сучасні тенденції у сфері інформаційних технологій, що сприятиме їх професійному розвитку та адаптації до швидко змінюваних умов цифрового середовища. Вивчення реальних кейсів та практичних ситуацій дозволить їм застосовувати отримані знання у вирішенні актуальних завдань у сфері ІТ.

Завдяки інтеграції теоретичного матеріалу та практичних завдань студенти зможуть навчитися застосовувати інструменти для автоматизації бізнес-процесів, створення інтелектуальних інформаційних систем і розв’язання складних технологічних завдань. Курс сприятиме формуванню у студентів аналітичного мислення, необхідного для ефективного проектування, оцінки та впровадження інформаційних систем у різних галузях.

Завершення курсу передбачає виконання підсумкового проєкту, де студенти матимуть змогу застосувати набуті знання для розробки власної інформаційної системи або її елемента, що дозволить їм отримати цінний досвід роботи з реальними проблемами та сучасними технологіями.

Тема 1.
ОСНОВНІ ПОНЯТТЯ ТЕОРІЇ ІНФОРМАЦІЙНИХ СИСТЕМ

Мета: ознайомлення здобувачів з поняттям інформаційної системи, її структурою, функціями та класифікацією, що дозволить краще розуміти принципи її функціонування та застосування.

План лекції

1. Визначення інформаційної системи.
2. Структура та компоненти інформаційної системи.
3. Основні функції інформаційної системи.
4. Класифікація інформаційних систем.

1. Визначення інформаційної системи

У сучасних умовах інформаційні системи (ІС) є невід'ємною складовою функціонування підприємств, організацій, установ і суспільства в цілому. Вони забезпечують ефективну обробку, зберігання та передачу даних, що відіграє важливу роль у процесах управління, прийняття рішень і оптимізації робочих процесів. Завдяки швидкому розвитку цифрових технологій інформаційні системи стають все більш інтегрованими, автоматизованими та адаптивними до змін у зовнішньому середовищі.

Використання інформаційних систем охоплює широкий спектр галузей – від бізнесу та фінансів до медицини, освіти та державного управління. Вони сприяють підвищенню продуктивності, забезпечують доступ до великих масивів даних, покращують комунікацію між структурними підрозділами організацій та мінімізують людський фактор у виконанні рутинних завдань [3].

Буде розглянуто сутність інформаційних систем, їхню класифікацію, основні елементи та функції, а також їхнє значення в різних сферах діяльності. Окрему увагу приділено сучасним тенденціям розвитку інформаційних систем, зокрема впровадженню штучного інтелекту, хмарних технологій, Інтернету

речей (IoT) та кібербезпеки, які значно розширюють можливості їх застосування [15].

Інформаційна система являє собою структуровану сукупність елементів, які забезпечують процеси збору, обробки, зберігання, аналізу та передачі інформації з метою підвищення ефективності управління, підтримки прийняття рішень та оптимізації операційної діяльності. Вона може функціонувати як у межах окремого підприємства, так і на міжорганізаційному чи глобальному рівні, сприяючи взаємодії між користувачами, системами та технологіями.

Інформаційні системи поділяються на кілька типів залежно від їхнього призначення та функціональності. Зокрема, існують такі види інформаційних систем:

- ***Операційні системи обробки даних*** (TPS – Transaction Processing Systems) – автоматизують рутинні операції, такі як облік товарів, обробка замовлень або банківські транзакції.
- ***Системи управління базами даних*** (DBMS – Database Management Systems) – забезпечують організацію, зберігання та пошук даних.
- ***Системи підтримки прийняття рішень*** (DSS – Decision Support Systems) – допомагають у аналізі даних і виборі оптимальних рішень.
- ***Корпоративні інформаційні системи*** (ERP – Enterprise Resource Planning Systems) – інтегрують управлінські процеси в межах компанії.
- ***Системи бізнес-аналітики*** (BI – Business Intelligence Systems) – аналізують великі масиви даних для стратегічного планування.
- ***Інтелектуальні інформаційні системи*** – використовують технології штучного інтелекту для автоматизації складних завдань.

Розвиток інформаційних систем відіграє ключову роль у цифровій трансформації суспільства, сприяючи автоматизації, підвищенню продуктивності, удосконаленню комунікацій та підвищенню конкурентоспроможності організацій. Завдяки сучасним технологіям, таким як хмарні обчислення, великі дані (Big Data), блокчейн і штучний інтелект,

інформаційні системи стають дедалі потужнішими та адаптивнішими до змінних умов бізнесу та суспільного життя.

Інформаційні системи мають широкий спектр застосувань у різних галузях економіки, науки, освіти, охорони здоров'я та державного управління. Їх використання значно підвищує ефективність діяльності організацій, сприяє оптимізації бізнес-процесів, покращує якість прийняття рішень та підвищує рівень автоматизації.

Застосування інформаційних систем у різних сферах

1. **Бізнес та управління**
    - Автоматизація бухгалтерського обліку та фінансового менеджменту (наприклад, системи 1С, SAP, Oracle ERP).
    - Управління ланцюгами постачання (SCM) для оптимізації логістики та товарообігу.
    - CRM-системи (Customer Relationship Management) для покращення роботи з клієнтами.
2. **Освіта**
    - Електронні навчальні платформи (Moodle, Google Classroom, Coursera) для дистанційного навчання.
    - Автоматизовані системи управління закладами освіти (наприклад, електронні журнали та рейтингові системи).
    - Використання штучного інтелекту для персоналізації навчання та аналізу успішності студентів.
3. **Охорона здоров’я**
    - Електронні медичні карти (EMR) та системи управління лікарнями (HIS – Hospital Information Systems).
    - Телемедицина та мобільні додатки для дистанційного моніторингу стану пацієнтів.
    - Використання великих даних для прогнозування епідемій та персоналізованої медицини.
4. **Державне управління та громадські послуги**

- Електронний документообіг та цифрові сервіси для громадян (e-Government).
- Автоматизовані системи управління податковими та митними службами.
- Розвиток "розумних міст" (Smart City) із застосуванням IoT (Інтернету речей) та штучного інтелекту.

5. **Промисловість та виробництво**
   - Автоматизовані системи управління виробництвом (MES – Manufacturing Execution Systems).
   - Використання роботизованих систем і штучного інтелекту для оптимізації виробничих процесів.
   - Аналіз великих даних для прогнозування несправностей обладнання та мінімізації простоїв.

Сучасні тенденції розвитку інформаційних систем

Розвиток інформаційних технологій призводить до постійної еволюції інформаційних систем. Серед найактуальніших трендів варто виділити:

- **Хмарні обчислення (Cloud Computing)** – забезпечують гнучкий доступ до обчислювальних ресурсів без необхідності утримання власної інфраструктури.
- **Великі дані (Big Data)** – технології аналізу та обробки величезних обсягів інформації для виявлення закономірностей та прийняття стратегічних рішень.
- **Штучний інтелект та машинне навчання (AI & ML)** – використовуються для автоматизації складних завдань, аналізу поведінки користувачів та створення інтелектуальних систем підтримки прийняття рішень.
- **Інтернет речей (IoT – Internet of Things)** – сприяє підключенню та взаємодії між "розумними" пристроями для автоматизованого управління в різних сферах (логістика, медицина, міська інфраструктура).

- **Кібербезпека** – у зв’язку зі зростанням загроз кіберзлочинності активно розвиваються системи захисту інформації, шифрування та блокчейн-технології.
- **Блокчейн** – використовується не лише у фінансовому секторі, а й у логістиці, документообігу, охороні здоров’я для забезпечення прозорості та безпеки операцій [9].

Інформаційні системи відіграють ключову роль у сучасному суспільстві, забезпечуючи ефективний обмін інформацією, автоматизацію процесів та підтримку прийняття рішень. Їхнє впровадження дозволяє організаціям підвищувати продуктивність, знижувати витрати та адаптуватися до динамічних змін ринку. Майбутній розвиток інформаційних систем буде пов’язаний із подальшим удосконаленням технологій штучного інтелекту, аналізу даних та цифрової безпеки, що відкриває нові можливості для інновацій та прогресу.

2. Структура та компоненти інформаційної системи

Інформаційна система (ІС) є складним комплексом взаємопов’язаних елементів, які забезпечують ефективний збір, обробку, збереження, аналіз, передачу та використання інформації. Вона відіграє ключову роль у функціонуванні підприємств, установ та організацій, сприяючи автоматизації процесів, підвищенню продуктивності та оптимізації управлінських рішень.

Для повноцінного функціонування інформаційна система повинна мати чітку структуру, що включає різні компоненти, які взаємодіють між собою та забезпечують її ефективність.

2. 1. Основні компоненти інформаційної системи

Структуру інформаційної системи можна розділити на кілька ключових компонентів, кожен з яких виконує певні функції та є важливим для її роботи.

1.1. Апаратне забезпечення (Hardware)

Апаратне забезпечення становить матеріальну основу інформаційної системи, яка забезпечує обробку, зберігання та передачу даних. До нього належать:

- **Обчислювальні пристрої** – комп'ютери, сервери, мобільні пристрої, що виконують процеси обробки даних.
- **Пристрої зберігання даних** – жорсткі диски, твердотільні накопичувачі (SSD), хмарні сховища, що використовуються для збереження інформації.
- **Мережеве обладнання** – маршрутизатори, комутатори, модеми, що забезпечують комунікацію між користувачами та серверами.
- **Пристрої введення та виведення інформації** – клавіатури, миші, сканери, монітори, принтери, сенсорні панелі тощо.

1.2. Програмне забезпечення (Software)

Програмне забезпечення є інструментом для керування апаратною частиною та виконання обчислювальних операцій. Воно поділяється на:

- **Операційні системи (OS)** – Windows, Linux, macOS, Android, iOS, які забезпечують взаємодію користувачів з апаратним забезпеченням.
- **Системи управління базами даних (СУБД)** – MySQL, PostgreSQL, Oracle, що відповідають за організацію, збереження та обробку інформації.
- **Прикладне програмне забезпечення** – текстові редактори, електронні таблиці, CRM- та ERP-системи, бізнес-аналітичні програми.
- **Мережеве програмне забезпечення** – веб-браузери, VPN-клієнти, хмарні платформи для забезпечення комунікації.

1.3. Інформаційне забезпечення (Data & Information)

Цей компонент включає всю інформацію, що обробляється системою:

- **Бази даних** – структуровані сховища інформації, які містять записи про користувачів, транзакції, бізнес-процеси.
- **Документація та електронні архіви** – нормативні документи, звіти, реєстраційні дані.
- **Аналітичні дані** – статистика, бізнес-інтелект (BI), прогнозні моделі.

1.4. Людський фактор (Humanware)

Користувачі є невід'ємною частиною інформаційної системи. Їх можна класифікувати наступним чином:

- **Оператори системи** – співробітники, що безпосередньо працюють з ІС (менеджери, аналітики, бухгалтери).
- **ІТ-спеціалісти** – адміністратори баз даних, програмісти, системні інженери, які підтримують і розвивають ІС.
- **Кінцеві користувачі** – працівники, клієнти або партнери, що використовують дані та послуги системи.

1.5. Організаційне забезпечення

Цей компонент містить правила та методології використання системи:

- **Політики безпеки** – заходи для захисту інформації та конфіденційних даних.
- **Стандарти управління інформаційними системами** – ISO 27001, GDPR, які регламентують захист та обробку інформації.
- **Процедури експлуатації** – інструкції та регламенти використання ІС співробітниками.

1.6. Телекомунікаційна інфраструктура

Забезпечує передачу інформації між різними елементами системи:

- **Локальні (LAN) та глобальні (WAN) мережі –** забезпечують обмін даними між серверами та клієнтами.
- **Протоколи передачі даних – TCP/IP, HTTP, FTP,** що визначають правила обміну інформацією.
- **Хмарні технології – Amazon AWS, Google Cloud, Microsoft Azure,** що дозволяють зберігати та обробляти інформацію віддалено.

2. 2. Функціональна структура інформаційної системи

Окрім фізичних компонентів, інформаційні системи мають логічну структуру, яка визначає, як саме відбувається обробка та використання інформації.

Рівні обробки інформації:

- **Рівень збору даних –** процес введення інформації у систему (сканування, введення через форми, API-запити).
- **Рівень збереження та обробки даних –** управління базами даних, фільтрація та структурування інформації.
- **Рівень аналізу інформації** – використання аналітичних інструментів, машинного навчання, бізнес-аналітики.
- **Рівень управління інформацією –** підтримка прийняття рішень та автоматизація бізнес-процесів.
- **Рівень взаємодії користувачів –** відображення інформації через графічний інтерфейс або API.

2. 3. Типи інформаційних систем за структурою

Інформаційні системи можуть мати різну архітектуру залежно від потреб користувачів та організацій:

- **Централізовані системи** – всі дані та обчислення виконуються на одному сервері (наприклад, банки).
- **Розподілені системи** – обробка інформації відбувається на різних серверах або пристроях (хмарні сервіси, блокчейн).
- **Клієнт-серверні системи** – розподіл функцій між сервером і кінцевими пристроями (веб-застосунки, ERP-системи).
- **Хмарні системи** – дані зберігаються та обробляються в хмарному середовищі (Google Drive, Dropbox, Microsoft OneDrive).

Декомпозиція інформаційної системи за функціональною ознакою (рис. 1.1) полягає у виокремленні її окремих частин, які мають назву функціональних підсистем (далі -ПС) (функціональні модулі, бізнес-додатки), що реалізують систему функцій управління. Функціональною ознакою зумовлюється призначення підсистеми, тобто те, для якої сфери діяльності вона призначена і які основні цілі, завдання та функції вона виконує. Функціональні підсистеми істотно залежать від предметної області (сфери застосування) інформаційних систем.

Рис. 1.1. Структура інформаційної системи

Вибір складу функціональних завдань функціональних підсистем управління здійснюється звичайно з урахуванням основних фаз управління: планування; обліку, контролю і аналізу; регулювання (виконання).

Планування - це управлінська функція, що забезпечує формування планів, відповідно до яких буде організовано функціонування об'єкта управління. Традиційно виокремлюють перспективне (5-Ю років), річне (1 рік) та оперативне (доба, тиждень, декада, місяць) планування.

Облік, контроль і аналіз — функції, що забезпечують одержання даних про стан керованої системи за певний проміжок часу, визначення факту і причини відхилення стану об'єкта управління від його планованого стану, а також виявлення розмірів цього відхилення. Облік ведеться за показниками

плану у вибраному діапазоні планування (оперативний, середньостроковий тощо).

Регулювання (виконання) — це функція, що забезпечує порівняння планованих та фактичних показників функціонування об'єкта управління і реалізацію необхідних керівних впливів за наявності відхилень від запланованих у заданому діапазоні (відрізку). Відповідно до виокремлених функціональних підсистем та з урахуванням вимог управління визначається склад завдань функціональних підсистем. Наприклад, інформаційна система управління персоналом підприємства може містити такі функціональні підсистеми:

· планування чисельності персоналу підприємства;

· розрахунок фонду заробітної плати персоналу;

· планування та організація навчання персоналу;

· управління кадровими переміщеннями;

· статистичний облік і звітність;

· довідки за запитом.

Вибір та обґрунтування складу функціональних завдань є одним із найважливіших елементів створення інформаційних систем. Аналіз функціональних завдань показує, що вони по-різному реалізуються в умовах використання інформаційних систем. Одне й те саме завдання може бути розв'язане (реалізоване) різними математичними методами, моделями й алгоритмами. Іноді цю функціональну підсистему називають підсистемою математичного забезпечення.

Серед багатьох варіантів реалізації є, як правило, найкращий, зумовлений можливостями обчислювальної системи і системи опрацювання даних у цілому.

У сучасних системах автоматизації проектування інформаційних систем цей компонент входить до складу так званих банків моделей і алгоритмів, з яких під час розробки інформаційних систем вибираються найефективніші для конкретного об'єкта управління.

Інформаційна система – це складна структура, яка складається з апаратного, програмного, інформаційного, людського, організаційного та комунікаційного забезпечення. Вона виконує важливі функції зі збору, збереження, обробки та аналізу даних, підтримуючи ефективне прийняття рішень та автоматизацію бізнес-процесів. Завдяки розвитку технологій, таких як штучний інтелект, великі дані, хмарні обчислення та Інтернет речей, інформаційні системи продовжують еволюціонувати, підвищуючи свою продуктивність, гнучкість та безпеку.

3. Основні функції інформаційної системи

Інформаційні системи (ІС) відіграють ключову роль у сучасному світі, забезпечуючи ефективне управління інформаційними потоками, оптимізацію бізнес-процесів та підтримку прийняття рішень. Їхня функціональність охоплює широкий спектр завдань – від збору та обробки даних до аналізу, передачі інформації, автоматизації процесів і забезпечення безпеки.

Успішне функціонування ІС залежить від її здатності виконувати комплекс взаємопов’язаних функцій, які забезпечують ефективну взаємодію між користувачами, програмним та апаратним забезпеченням, а також інформаційними ресурсами. Нижче наведено основні функції інформаційних систем, їх особливості та значення в управлінні організаціями та підприємствами.

1. Функції збору та введення даних

Початковий етап роботи інформаційної системи полягає у збиранні та реєстрації інформації для подальшого аналізу та обробки.

- **Автоматизований збір даних** – отримання інформації з різних джерел, таких як датчики IoT, сканери штрих-кодів, електронні форми, CRM-системи, API-запити.
- **Ручне введення даних** – внесення інформації користувачами через інтерфейс системи (наприклад, форми, документи, електронні таблиці).

- **Перевірка та валідація інформації** – автоматичне або ручне визначення коректності та точності введених даних для уникнення помилок та дублювання.
- **Стандартизація та формалізація інформації** – приведення даних до уніфікованого вигляду для зручності обробки та інтеграції з іншими системами.

2. Функції зберігання та управління даними

Збереження та організація інформації є фундаментальними завданнями ІС, що забезпечують надійний доступ до даних у будь-який момент часу.

- **Централізоване та розподілене зберігання** – використання серверів, хмарних технологій або баз даних для ефективного збереження інформації.
- **Ієрархічна організація даних** – структура, що дозволяє швидко знайти необхідні записи (файлова система, база даних, архіви).
- **Механізми резервного копіювання та відновлення** – створення копій важливої інформації для захисту від втрати через апаратні збої, хакерські атаки або інші форс-мажорні обставини.
- **Контроль версій** – збереження змін у документах та записах, що дозволяє повернутися до попередніх версій інформації у разі потреби.
- **Керування доступом до інформації** – розмежування прав доступу для різних категорій користувачів (звичайні користувачі, адміністратори, аналітики).

3. Функції обробки та аналізу інформації

Обробка даних є основною складовою функціонування інформаційної системи, яка дозволяє отримувати корисні знання з великих обсягів інформації.

- **Фільтрація та класифікація даних** – автоматичне групування інформації за певними критеріями (наприклад, сегментація клієнтів у CRM-системі).

- **Обчислювальні операції** – виконання розрахунків, аналіз фінансових показників, прогнозування тенденцій.
- **Аналіз великих даних (Big Data Analytics)** – обробка та інтерпретація масивів інформації для виявлення закономірностей та прийняття стратегічних рішень.
- **Візуалізація інформації** – графічне представлення даних у вигляді діаграм, графіків, аналітичних звітів для спрощення сприйняття.
- **Машинне навчання та штучний інтелект** – використання алгоритмів AI для автоматичного аналізу та прогнозування на основі історичних даних.

4. Функції передачі та обміну інформацією

Передача даних є важливим етапом у роботі інформаційної системи, який забезпечує швидкий доступ до інформації між користувачами та системами.

- **Мережевий обмін даними** – використання локальних (LAN) та глобальних (WAN) мереж для передачі інформації між пристроями та серверами.
- **API та інтеграція із зовнішніми системами** – обмін інформацією між різними інформаційними системами через програмні інтерфейси.
- **Автоматизоване оновлення інформації** – регулярна синхронізація даних між різними компонентами системи.

5. Функції підтримки прийняття рішень

Інформаційні системи допомагають керівникам та аналітикам у виборі оптимальних рішень на основі актуальних даних.

- **Генерація звітів** – автоматичне формування фінансової, виробничої, аналітичної та іншої документації.
- **Інтерактивний аналіз даних** – можливість проводити глибоке дослідження інформації за допомогою аналітичних панелей та інструментів.

- **Моделювання сценаріїв розвитку** – оцінка можливих наслідків управлінських рішень у різних ситуаціях.
- **Рекомендаційні алгоритми** – використання штучного інтелекту для надання персоналізованих рекомендацій (наприклад, в електронній комерції).

6. Функції автоматизації бізнес-процесів

Автоматизація є одним із головних завдань інформаційних систем, що дозволяє значно зменшити ручну працю та підвищити ефективність діяльності організацій.

- **Оптимізація робочих процесів** – автоматизація бухгалтерського обліку, управління персоналом, логістики.
- **Управління задачами та проєктами** – інтеграція із системами контролю виконання завдань (наприклад, Trello, Jira, Asana).
- **Роботизація процесів (RPA)** – використання програмних роботів для виконання рутинних операцій.

7. Функції безпеки та захисту інформації

Безпека є критично важливою складовою будь-якої інформаційної системи, яка забезпечує конфіденційність, цілісність та доступність даних.

- **Аутентифікація користувачів** – використання паролів, біометричних даних, двофакторної авторизації для захисту доступу.
- **Моніторинг активності користувачів** – ведення логів та аудит дій для виявлення аномальної активності.
- **Шифрування інформації** – захист даних під час зберігання та передачі за допомогою криптографічних методів.
- **Захист від кіберзагроз** – використання антивірусного ПЗ, міжмережевих екранів (firewall) та систем виявлення атак [17].

Інформаційні системи виконують широкий спектр функцій, спрямованих на обробку, аналіз, передачу, захист та використання інформації. Завдяки постійному вдосконаленню технологій, вони стають більш інтегрованими,

інтелектуальними та безпечними, забезпечуючи ефективну роботу підприємств та організацій у цифрову епоху.

4. Класифікація інформаційних систем

Інформаційні системи (ІС) охоплюють широкий спектр застосувань у різних сферах діяльності, і їхня класифікація дозволяє чітко визначити особливості, можливості та функціональність кожного типу. В залежності від сфери застосування, рівня управління, типу оброблюваних даних та інших критеріїв, інформаційні системи поділяються на різні види [8].

У цьому розділі розглянемо основні підходи до класифікації інформаційних систем та детально проаналізуємо їхні особливості.

1. Класифікація за рівнем управління

Різні інформаційні системи використовуються на різних рівнях управління організацією – від оперативного до стратегічного.

- **Операційні інформаційні системи (Transaction Processing Systems, TPS)**
    - Виконують рутинні, повторювані операції, такі як введення, збереження та обробка транзакцій.
    - Приклади: касові апарати, банківські системи, системи реєстрації замовлень.
- **Системи управління знаннями (Knowledge Management Systems, KMS)**
    - Використовуються для збору, зберігання та розповсюдження корпоративних знань.
    - Приклади: корпоративні вікі, бази знань, навчальні платформи.
- **Системи підтримки прийняття рішень (Decision Support Systems, DSS)**
    - Аналізують дані та надають рекомендації для прийняття управлінських рішень.

- Приклади: фінансові аналітичні системи, CRM-аналітика.

- **Інформаційно-аналітичні системи (Executive Information Systems, EIS)**
  - Використовуються керівниками для стратегічного аналізу та планування.
  - Приклади: системи бізнес-аналітики, інструменти прогнозування ринку.

2. Класифікація за функціональним призначенням

Інформаційні системи поділяються за сферою застосування та специфікою функціональних можливостей.

- **Операційні системи обробки даних (Operational Information Systems)**
  - Призначені для обліку та контролю виконання стандартних процесів у компанії.
  - Приклад: бухгалтерські програми (1С, SAP).
- **Системи управління підприємством (Enterprise Resource Planning, ERP)**
  - Інтегровані системи для комплексного управління бізнесом.
  - Приклад: Oracle ERP, Microsoft Dynamics.
- **Системи управління взаємовідносинами з клієнтами (Customer Relationship Management, CRM)**
  - Використовуються для управління продажами, маркетингом та підтримкою клієнтів.
  - Приклад: Salesforce, HubSpot.
- **Системи управління ланцюгами постачання (Supply Chain Management, SCM)**
  - Автоматизують процеси постачання, складування та логістики.
  - Приклад: SAP SCM, IBM Sterling.
- **Системи бізнес-аналітики (Business Intelligence, BI)**
  - Використовуються для аналізу бізнес-даних та побудови прогнозів.

- Приклад: Power BI, Tableau, Google Data Studio.

3. Класифікація за типом оброблюваних даних

Інформаційні системи можуть працювати з різними типами даних, що визначає їхню архітектуру та методи аналізу.

- **Текстові інформаційні системи** – обробляють текстові документи та бази знань (Google Docs, Wikipedia).
- **Числові інформаційні системи** – використовуються для фінансового аналізу, бухгалтерії (Excel, SAP).
- **Графічні інформаційні системи** – працюють із зображеннями та мультимедіа (AutoCAD, Adobe Photoshop).
- **Геоінформаційні системи (GIS)** – аналізують геопросторові дані (Google Maps, ArcGIS).

4. Класифікація за способом реалізації та доступу

Залежно від технології реалізації, інформаційні системи поділяються на кілька категорій:

- **Локальні інформаційні системи**
  - Функціонують у межах однієї організації, без підключення до зовнішніх ресурсів.
  - Приклад: внутрішня бухгалтерська система компанії.
- **Мережеві інформаційні системи**
  - Підключені до локальних або глобальних мереж для обміну інформацією.
  - Приклад: корпоративні портали, ERP-системи.
- **Хмарні інформаційні системи (Cloud-Based Systems)**
  - Дані та обчислювальні процеси розміщуються у хмарному середовищі.
  - Приклад: Google Drive, Microsoft Azure.
- **Мобільні інформаційні системи**
  - Оптимізовані для використання на смартфонах та планшетах.
  - Приклад: мобільний банкінг, мобільні CRM.

5. Класифікація за галузевою спеціалізацією

Деякі інформаційні системи призначені для використання у конкретних галузях економіки.

- **Фінансові інформаційні системи** – забезпечують облік фінансів, управління бюджетом та аналіз ризиків (SAP Financials, QuickBooks).
- **Медичні інформаційні системи** – підтримують електронні медичні картки та телемедицину (MedSAP, DrChrono).
- **Освітні інформаційні системи** – управління навчальним процесом та дистанційним навчанням (Moodle, Google Classroom).
- **Юридичні інформаційні системи** – забезпечують аналіз та обробку юридичних документів (LexisNexis, Судова Влада України).
- **Транспортні та логістичні інформаційні системи** – моніторинг і управління перевезеннями (Uber Freight, DHL Logistics).

6. Класифікація за рівнем інтеграції

Інформаційні системи можуть бути ізольованими або інтегрованими з іншими системами.

- **Автономні інформаційні системи** – працюють самостійно, без взаємодії з іншими системами.
- **Інтегровані інформаційні системи** – об’єднують декілька підсистем для комплексного управління організацією.

Класифікація інформаційних систем дозволяє визначити їхню функціональність, сферу застосування та особливості роботи. Вони можуть бути розраховані як на виконання вузькоспеціалізованих завдань, так і на комплексне управління бізнес-процесами в організації [20].

Сучасні тенденції розвитку ІС спрямовані на інтеграцію технологій штучного інтелекту, хмарних сервісів, великих даних та кібербезпеки, що робить їх ще більш ефективними та адаптивними до змін у бізнес-середовищі.

Зокрема, штучний інтелект відіграє ключову роль у автоматизації процесів, аналізі великих обсягів інформації та прогнозуванні майбутніх тенденцій. Використання машинного навчання та алгоритмів обробки

природної мови дозволяє покращити якість прийняття рішень та персоналізувати взаємодію з користувачами.

Хмарні технології забезпечують гнучкість, масштабованість та доступність даних у будь-який час та з будь-якого місця, що сприяє оптимізації бізнес-процесів. Великий обсяг даних (Big Data) дозволяє компаніям отримувати цінні аналітичні висновки, що допомагають у стратегічному плануванні та підвищенні конкурентоспроможності.

Кібербезпека, у свою чергу, стає невід'ємною частиною розвитку інформаційних систем, оскільки зростання цифрових загроз вимагає вдосконалення механізмів захисту даних, впровадження сучасних методів автентифікації та систем моніторингу безпеки.

Контрольні питання

1. Що таке інформаційна система та які її основні компоненти?
2. Які функції виконує інформаційна система та як вони впливають на управління даними?
3. Які види інформаційних систем існують за рівнем управління та функціональним призначенням?
4. У чому полягає різниця між операційними, аналітичними та підтримуючими інформаційними системами?
5. Яка роль апаратного та програмного забезпечення у функціонуванні інформаційної системи?
6. Як забезпечується безпека та збереження інформації в інформаційних системах?
7. Які особливості має автоматизація бізнес-процесів за допомогою інформаційних систем?
8. Які відмінності між локальними, хмарними та розподіленими інформаційними системами?
9. Як інформаційні системи підтримують процес прийняття рішень у бізнесі та управлінні?

Тема 2.
ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Мета: Метою лекції є ознайомлення здобувачів з основними поняттями, структурою та функціями інформаційних систем, а також їхнім впливом на розвиток сучасного суспільства та бізнесу.

План лекції

1. Загальні поняття інформаційних технологій.
2. Види інформаційних систем.
3. Сучасні ІТ-технології та інновації.

1. Загальні поняття інформаційних технологій

Технологія з грецького означає: «*logos*» – поняття, вчення, «*techne*» – мистецтво, майстерність, уміння, процес. Під процесом потрібно розуміти певну сукупність дій, які скеровані на досягнення поставленої мети. Процес повинен визначатися вибраною стратегією і реалізуватися за допомогою сукупності різних засобів і методів.

Інформація є одним з найцінніших ресурсів суспільства поруч з традиційними матеріальними видами ресурсів, як нафта, метал, корисні копалини тощо, тому, процес переробки інформації, подібно до процесів переробки матеріальних ресурсів можна сприймати як технологію. Інформаційна технологія передбачає вміння грамотно працювати з інформацією і обчислювальною технікою [11].

Інформаційні технології (ІТ) – це сукупність методів, процесів, програмного та апаратного забезпечення, що використовуються для збирання, збереження, обробки, аналізу, передавання та захисту інформації з метою підвищення ефективності діяльності організацій, автоматизації бізнес-процесів, підтримки прийняття рішень та розвитку цифрового суспільства.

Основна мета інформаційної технології полягає у тому, щоб перетворити інформаційний ресурс на високоякісний інформаційний продукт, який відповідає вимогам користувача. Методи, що застосовуються в рамках інформаційної технології, включають різноманітні способи обробки і передачі даних. Засоби реалізації цих методів можуть бути математичними, програмними, інформаційними, технічними тощо.

Інформаційна технологія, в такому розумінні, виступає як цілісна технічна система, що спрямована на створення, передачу, зберігання і представлення інформаційного продукту (будь то дані, знання чи ідеї) із мінімальними витратами. Вона функціонує з урахуванням соціального середовища, в якому ця технологія використовується і розвивається.

Практично методи і засоби обробки даних можуть бути різними, тому доцільно виділити *глобальну, базову та конкретну інформаційні технології* (Рис.2.1):

Рис.2.1. Поділ інформаційних технологій

- **Глобальні інформаційні технології:** охоплюють великомасштабні системи, які використовуються на міжнародному рівні для забезпечення глобального доступу до інформації.

Приклади глобальних інформаційних технологій:

1. Інтернет і його основні сервіси (веб-браузери, пошукові системи, електронна пошта, хмарні сховища даних), які забезпечують доступ до інформації для мільйонів користувачів у всьому світі.

2. Глобальні платформи обміну інформацією, такі як соціальні мережі (Facebook, Twitter, LinkedIn), що дозволяють обмінюватися знаннями, ідеями та даними незалежно від місцезнаходження.

3. Глобальні інформаційно-аналітичні системи, такі як платформи для моніторингу клімату чи міжнародної економіки (наприклад, World Bank Data, Google Earth Engine).

- **Базові інформаційні технології:** забезпечують фундаментальні методи обробки даних і лежать в основі функціонування інформаційних систем.

Приклади базових інформаційних технологій:

1. Системи управління базами даних (SQL, Oracle), які забезпечують структурування, зберігання і обробку великих масивів даних.

2. Операційні системи (Windows, Linux, macOS), які створюють середовище для функціонування програм і керування апаратними ресурсами.

3. Алгоритми обробки інформації, такі як шифрування даних (AES, RSA), які гарантують безпеку передачі та зберігання даних.

- **Конкретні інформаційні технології:** реалізують специфічні завдання в окремих галузях або проектах, адаптуючи базові методи та засоби під конкретні потреби користувачів.

Приклади конкретних інформаційних технологій:

1. Автоматизовані системи управління виробничими процесами (SCADA), що використовуються в промисловості для моніторингу і контролю виробничих ліній.

2. Медичні інформаційні системи (EHR/EMR), які допомагають лікарям вести облік пацієнтів, розробляти плани лікування та стежити за ефективністю терапії.

3. Інформаційно-аналітичні системи для фінансового моніторингу, які виявляють підозрілі транзакції та допомагають боротися з шахрайством у банківській сфері.

Інформаційна технологія є найбільш важливою складовою процесу використання інформаційних ресурсів суспільства. До теперішнього часу вона пройшла кілька еволюційних етапів, зміна яких визначалася розвитком науково-технічного прогресу та появою нових технічних засобів переробки інформації [12].

Інформаційно-комунікаційні технології (ІКТ) – це сукупність методів, засобів і процесів, які забезпечують створення, зберігання, передачу, обробку та управління інформацією. Термін охоплює всі технології, які застосовуються для спілкування, роботи з даними, автоматизації процесів і полегшення доступу до знань.

До ІКТ належать як традиційні форми комунікацій (телефонія, телебачення), так і новітні цифрові рішення, такі як хмарні обчислення, мобільні додатки, інтернет речей, системи дистанційного навчання та онлайн-конференції. Наприклад, сучасні відеоконференц-системи дозволяють проводити зустрічі у реальному часі, незалежно від географічного розташування учасників. Інтернет-платформи надають можливість обмінюватися документами, співпрацювати над проектами в хмарних сервісах, а також швидко знаходити потрібну інформацію за допомогою пошукових систем.

Крім того, ІКТ сприяють розширенню доступу до освіти та підвищенню рівня цифрової грамотності населення. З їх допомогою люди можуть отримувати нові навички через онлайн-курси, вебінари та інтерактивні тренінги. У бізнесі ІКТ застосовуються для оптимізації внутрішніх процесів,

управління проектами, автоматизації фінансових операцій, а також для покращення взаємодії з клієнтами через CRM-системи та соціальні мережі.

Інформаційно-комунікаційні технології постійно вдосконалюються, і їх роль у житті суспільства з кожним днем зростає. Вони стають невід'ємною частиною нашого повсякденного життя, значно змінюючи спосіб, яким ми спілкуємося, працюємо, навчаємося та отримуємо доступ до інформації.

Сучасні інформаційні технології сприяють швидкому обміну даними, впровадженню новітніх цифрових рішень та інтеграції технологій у різні сфери життя – від фінансів і медицини до освіти та державного управління.

Інформаційні технології складаються з таких ключових компонентів:

- **Апаратне забезпечення (Hardware)** – фізичні пристрої, що забезпечують функціонування інформаційних систем, такі як сервери, комп'ютери, мережеве обладнання, мобільні пристрої та периферійні пристрої введення-виведення.
- **Програмне забезпечення (Software)** – операційні системи, прикладні програми, антивірусні системи та спеціалізовані рішення для автоматизації діяльності (ERP, CRM, SCM, BI-системи).
- **Дані та бази даних (Data & Databases)** – структурована та неструктурована інформація, яка зберігається, аналізується і використовується для прийняття рішень.
- **Мережеві технології та комунікації (Networking & Communication)** – технології передавання інформації через локальні та глобальні мережі (Інтернет, Wi-Fi, 5G, VPN).
- **Кібербезпека (Cybersecurity)** – методи та засоби захисту інформаційних систем від загроз, включаючи шифрування, системи аутентифікації, фаєрволи, антивірусне ПЗ та політики безпеки.
- **Інтелектуальні технології (AI & ML)** – системи штучного інтелекту, машинного навчання, аналізу великих даних (Big Data), які дозволяють здійснювати прогнозування, автоматизацію та прийняття рішень на основі даних.

- **Хмарні технології (Cloud Computing)** – сервіси для обробки та зберігання даних у віддалених дата-центрах (AWS, Google Cloud, Microsoft Azure).

Інформаційні технології виконують широкий спектр функцій, що сприяють ефективному управлінню даними та автоматизації процесів. Основними функціями є:

Обробка та аналіз даних – трансформація інформації у зручний для користувачів формат за допомогою алгоритмів та програмного забезпечення.

Автоматизація процесів – мінімізація людської участі у виконанні рутинних завдань через використання інформаційних систем.

Передача та збереження інформації – забезпечення швидкого доступу до даних у локальних та хмарних середовищах.

Захист інформації – контроль доступу, шифрування даних, резервне копіювання та антивірусні рішення для гарантування безпеки інформаційних активів.

Оптимізація бізнес-процесів – покращення ефективності організацій через інтеграцію цифрових технологій у внутрішні процеси.

Розвиток штучного інтелекту та автоматизованого прийняття рішень – використання алгоритмів машинного навчання для обробки інформації та прогнозування трендів.

Інформаційні технології змінили підхід до ведення бізнесу, освіти, охорони здоров’я, державного управління та інших сфер. Основні напрями впливу IT:

- **Бізнес та економіка** – впровадження автоматизованих систем управління підприємствами (ERP, CRM), онлайн-платежів, електронної комерції (e-commerce).
- **Освіта** – розвиток дистанційного навчання (Moodle, Google Classroom), адаптивних навчальних платформ та онлайн-курсів.
- **Охорона здоров'я** – електронні медичні картки, телемедицина, діагностика на основі штучного інтелекту.
- **Державне управління** – цифровізація адміністративних послуг, електронний документообіг, концепція «розумного міста» (Smart City).

- **Соціальна сфера** – соціальні мережі, мобільні додатки для комунікації, цифрові платформи для управління персональними фінансами.

Розвиток інформаційних технологій продовжується стрімкими темпами, і сьогодні визначаються такі ключові тенденції:

Штучний інтелект та машинне навчання – використання нейромереж для автоматизації складних завдань та прийняття рішень.

Хмарні обчислення – зростання популярності хмарних сервісів для зберігання та обробки даних.

Інтернет речей (IoT) – взаємодія пристроїв у мережі для підвищення рівня автоматизації виробництва, транспорту, енергетики.

Кібербезпека та цифровий суверенітет – розвиток технологій захисту даних та національних IT-інфраструктур.

Блокчейн та фінансові технології (FinTech) – децентралізовані платформи для проведення транзакцій, електронні платежі, криптовалюти.

5G та нові мережеві технології – збільшення швидкості передачі даних і розширення можливостей мобільного зв'язку.

Доповнена та віртуальна реальність (AR/VR) – застосування у розважальній індустрії, медицині, освіті та архітектурі.

Попри значний прогрес, інформаційні технології стикаються з низкою викликів:

- **Кібербезпека та конфіденційність** – зростання ризиків зламів, витоку даних, шахрайства у цифровому середовищі.
- **Етичні питання використання IT** – проблеми приватності, використання ШІ, маніпуляція даними.
- **Доступність та цифровий розрив** – нерівний доступ до сучасних технологій у різних регіонах світу.
- **Енергоспоживання та екологічний вплив** – питання ефективного використання ресурсів у дата-центрах.
- **Автоматизація та зміни на ринку праці** – скорочення традиційних професій через роботизацію.

Інформаційні технології є основою цифрової трансформації суспільства, сприяючи розвитку бізнесу, науки, медицини, освіти та інших галузей. Вони продовжують еволюціонувати, пропонуючи нові можливості для автоматизації, аналізу даних, комунікації та управління. У майбутньому зростаюча інтеграція ШІ, Інтернету речей, блокчейну та хмарних сервісів зробить інформаційні технології ще більш потужними та доступним.

Інформаційна технологія обробки даних слугує інструментом для автоматизації завдань, які мають чітку структуру і передбачуваний алгоритм виконання. Вона базується на використанні стандартних процедур та відомих методів опрацювання, які застосовуються для отримання бажаних результатів із наявних вхідних даних. Основна мета такої технології — автоматизувати рутинні, повторювані операції, що зазвичай виконуються в рамках управлінської діяльності [20].

На рівні операційної діяльності такі технології допомагають вирішувати наступні завдання:

- **Обробка даних про транзакції компанії:** наприклад, автоматизація запису продажів у касових апаратах, формування рахунків-фактур або обробка платежів у банківських системах.
- **Формування регулярних звітів:** створення стандартних контрольних звітів про обіг коштів, залишки на складах, продуктивність відділів або співробітників.
- **Відповіді на поточні запити:** оперативна підготовка відповідей на запити керівництва, надання аналітичних довідок чи формування запитуваної документації для клієнтів у форматі електронних звітів або друкованих документів.

Приклад:

Банківська система автоматично обробляє щоденні транзакції клієнтів, формує звітність для центрального офісу, а також відповідає на запити клієнтів про баланс на рахунках, надходження коштів чи графік погашення кредиту. Усе

це здійснюється за допомогою чітко визначених алгоритмів і встановлених процедур, що виключає необхідність ручного втручання в кожну операцію.

Інформаційна технологія обробки даних використовується не лише в банківському секторі, але й у роздрібній торгівлі, складській логістиці, страховій справі, державних установах тощо. Наприклад, супермаркети застосовують системи для обліку продажів, автоматичного поповнення запасів товарів і створення щотижневих звітів про виручку. Страхові компанії використовують такі технології для автоматизованого обліку договорів, нарахування страхових премій і формування аналітичних звітів для акціонерів.

Відтак, інформаційні технології обробки даних є критично важливими для забезпечення безперебійної операційної роботи, економії часу та підвищення точності управлінських рішень.

2. Види інформаційних систем

Відомо, що інформаційні системи, включаючи бази та банки даних, являють собою складні структури. Вони мають багаторівневу архітектуру, в якій окремі частини можуть бути недоступними для кінцевого користувача або навіть для адміністратора системи. Таким чином, інформація, що зберігається в цих системах, має різний ступінь доступності залежно від рівня її розгляду.

Наприклад, дані в базі можуть бути представлені на рівні користувацьких запитів, де доступні лише вибрані поля чи записи. Водночас адміністратор може мати доступ до фізичної структури таблиць, індексів та механізмів їхньої підтримки. Але навіть адміністратор може не знати, як саме дані фізично розташовані в пам’яті сервера чи які алгоритми використовуються для шифрування, якщо це було налаштовано іншою стороною.

Така розподілена структура доступу дозволяє з одного боку забезпечити конфіденційність даних, з іншого – зручність роботи для кінцевих користувачів, яким не потрібно знати всі технічні деталі. Ступінь «приватності» інформації може залежати як від політик доступу в організації, так і від обраної технології.

Наприклад, сучасні реляційні бази даних часто підтримують рольову модель доступу, що дозволяє налаштувати, які саме дані може бачити кожен користувач чи група користувачів. Це забезпечує не лише безпеку, а й ефективну роботу з великими обсягами даних.

Коли створюються або класифікуються інформаційні системи, неминуче постає питання формалізації — точного математичного та алгоритмічного опису розв'язуваних задач. Рівень такої формалізації має значний вплив на загальну ефективність роботи системи, а також на ступінь її автоматизації. Вищий рівень автоматизації досягається, коли людина мінімально залучена до прийняття рішень, які базуються на отриманій інформації [5].

Більш точний і деталізований математичний опис задачі дозволяє значно підвищити можливості комп'ютерної обробки даних. Це, у свою чергу, зменшує необхідність втручання людини, що веде до зростання рівня автоматизації. Наприклад, в автоматизованих системах управління складом або логістикою точний алгоритмічний опис процесів, таких як відстеження запасів або маршрутизація доставки, дає змогу значно скоротити ручну працю. Завдяки цьому система може самостійно обробляти дані про залишки товарів, прогнозувати попит та оптимізувати запаси, а оператор лише контролює результати та вносить коригування в разі потреби.

Отже, чим більш формалізованими та чіткими є математичні моделі та алгоритми, тим вища точність і швидкість обробки даних, а також зростає можливість автоматизації складних процесів. Це зменшує витрати часу і ресурсів, підвищуючи загальну ефективність інформаційної системи.

Інформаційні системи розробляються для вирішення різних типів задач, які поділяються на:

- **структуровані (формалізовані)** — задачі, де всі елементи та їх взаємозв'язки добре визначені і відомі;
- **неструктуровані (неформалізовані)** — задачі, для яких неможливо чітко виділити елементи і встановити між ними зв'язки;

- **частково структуровані** — задачі, які мають як формалізовані, так і неформалізовані елементи.

Структуровані задачі можна представити у вигляді математичних моделей із відомими алгоритмами розв’язання. Наприклад, облік залишків на складі, розрахунок заробітної плати працівників, складання фінансової звітності — це типові структуровані задачі. Вони мають рутинний характер, вирішуються багаторазово та підлягають повній автоматизації, зводячи участь людини до мінімуму. Завдяки інформаційній системі такі задачі виконуються швидше, зменшується ймовірність помилок і значно підвищується ефективність.

Неструктуровані задачі не мають чітких алгоритмів розв’язання, і їхнє вирішення значною мірою залежить від інтуїції, досвіду та знань людини. Наприклад, стратегічне планування розвитку компанії, прийняття рішення щодо інвестицій у новий ринок або розробка концепції маркетингової кампанії — це типові приклади неструктурованих задач. У таких випадках інформаційна система може лише надавати допоміжні дані, які спеціаліст використовує для аналізу та прийняття рішень.

Частково структуровані задачі містять елементи обох попередніх категорій. Наприклад, розподіл ресурсів у межах проекту може бути частково структурованим: існують певні стандартизовані процедури для оцінки витрат і термінів, але остаточне рішення залежить від аналізу ситуації і досвіду керівника проекту. У таких задачах інформаційна система допомагає автоматизувати рутинну частину роботи (наприклад, обчислення оптимального розподілу ресурсів), залишаючи людині функції остаточного аналізу та ухвалення рішень.

Розширення і приклади:

- Для структурованих задач, таких як розрахунок податкових відрахувань чи створення стандартних звітів, система автоматично виконує всі операції без втручання людини.

- Для неструктурованих задач, як-от визначення стратегії просування нового продукту, система може надати доступ до аналітичних даних, але остаточне рішення приймає людина.
- Частково структуровані задачі, наприклад, планування логістичних маршрутів, можуть бути частково автоматизовані: інформаційна система розрахує найбільш економічно вигідний маршрут, але вибір залежить від додаткових чинників, які враховує фахівець.

Таким чином, інформаційні системи адаптуються до типу задачі, допомагаючи досягти максимальної ефективності як у рутинних, так і в аналітичних процесах.

Інформаційні системи, призначені для розв'язання частково структурованих задач, можна умовно поділити на два типи:

- **Системи, які генерують управлінські звіти та зосереджені на обробці даних.** Вони виконують операції пошуку, сортування, фільтрації тощо, забезпечуючи керівників необхідною інформацією для самостійного прийняття рішень.
- **Системи, які пропонують можливі варіанти вирішення задач.** У цьому випадку система формує декілька альтернатив, а остаточне рішення людина приймає шляхом вибору найбільш підходящого варіанту.

Інформаційні системи, які генерують варіанти рішень, поділяються на модельні та експертні.

Модельні інформаційні системи забезпечують користувача різноманітними математичними, статистичними, фінансовими та іншими моделями. Ці моделі допомагають у розробці та оцінці альтернативних варіантів вирішення проблеми. Завдяки взаємодії з моделлю в інтерактивному режимі користувач може отримати необхідну інформацію для прийняття обґрунтованого рішення.

Основними функціями модельної інформаційної системи є:

- **Використання стандартних математичних моделей.** Це дає змогу вирішувати типові завдання моделювання, наприклад, "що буде, якщо

збільшити виробництво на 10%?" або "як досягти бажаного рівня доходу, враховуючи поточні витрати?". Система дозволяє проводити аналіз чутливості, визначаючи, як зміна одного чи кількох параметрів впливає на загальний результат.

- **Швидка й точна інтерпретація результатів моделювання.** Отримані дані легко аналізувати і використовувати для прийняття управлінських рішень.
- **Оперативна підготовка й коригування вхідних параметрів та обмежень.** Це означає, що користувач може швидко внести зміни у вихідні дані, наприклад, змінити рівень витрат або додати нові обмеження, і система автоматично перерахує результати.
- **Графічне відображення динаміки моделі.** Система візуалізує зміни параметрів і їх вплив на кінцеві результати у вигляді графіків, діаграм або таблиць.
- **Пояснення кроків формування і роботи моделі.** У разі потреби система може надати користувачеві рекомендації або коментарі щодо того, як модель була побудована, які параметри використовуються і як найкраще інтерпретувати результати.

Приклади:

- Фінансове планування: система моделює різні сценарії бюджетування компанії, наприклад, що станеться з прибутком, якщо підняти зарплати або скоротити витрати на маркетинг.
- Логістика: за допомогою модельної системи можна прорахувати оптимальний маршрут доставки, враховуючи витрати на паливо, час доставки та наявність складів на шляху.
- Управління запасами: моделюється, як зміняться витрати на зберігання і ризик дефіциту, якщо збільшити обсяги закупівель або змінити графік поповнення складів.

Ці функції та можливості роблять модельні інформаційні системи незамінними для багатьох галузей, де потрібно швидко і точно оцінювати різні сценарії розвитку подій.

Експертні інформаційні системи використовуються для вироблення і оцінки можливих варіантів рішень шляхом обробки знань за допомогою спеціалізованих експертних систем. Вони надають користувачеві експертну підтримку у прийнятті рішень на двох рівнях.

На першому рівні експертна підтримка базується на концепції «типових управлінських рішень». Згідно з цією концепцією, більшість ситуацій, які регулярно виникають у процесі управління, можна звести до певних типових класів рішень. Це означає, що на основі попереднього досвіду і зібраних даних формується стандартний набір альтернативних варіантів, які можуть бути використані в схожих обставинах. Для цього створюється спеціальний інформаційний фонд, де зберігаються, систематизуються і аналізуються ці типові альтернативи.

Приклади:

1. **Управління виробництвом:**

У системах планування виробництва типові рішення можуть включати стандартні алгоритми розподілу ресурсів, графіки технічного обслуговування обладнання або типові методики усунення відхилень у виробничих процесах. Наприклад, якщо конвеєр зупиняється через певну несправність, система надає кілька варіантів стандартних дій: перенаправлення ресурсів, переналаштування обладнання або тимчасове збільшення виробничих змін.

2. **Фінансовий менеджмент.**

У сфері управління фінансами експертні системи можуть рекомендувати типові варіанти оптимізації бюджету, скорочення витрат або вибору джерел фінансування. Наприклад, якщо компанія стикається з нестачею оборотних коштів, система може запропонувати такі типові альтернативи, як використання короткострокового кредиту, затримка виплат постачальникам або перегляд інвестиційних планів.

3. **Медичні інформаційні системи:**

У медичних експертних системах на основі типових діагностичних моделей і протоколів лікування можна запропонувати найбільш імовірні альтернативи терапії. Наприклад, у разі певних симптомів система може надати перелік стандартних методів діагностики і лікування, що застосовуються в аналогічних випадках.

Загалом експертні інформаційні системи на першому рівні допомагають швидко і обґрунтовано обирати з типових варіантів, зменшуючи час на прийняття рішень і мінімізуючи ймовірність помилок.

Окрім класифікації інформаційних систем за ступенем структурованості розв'язуваних задач, існують інші підходи до їх поділу, які широко застосовуються на практиці. Одним із таких критеріїв є **ступінь автоматизації** інформаційних процесів. Відповідно до цього критерію, інформаційні системи можна розділити на:

- **ручні**,
- **автоматичні**,
- **автоматизовані.**

Ручні інформаційні системи повністю залежать від людського втручання: усі дії, включаючи збір, обробку, аналіз і збереження даних, виконуються людиною без використання сучасних технічних засобів. *Наприклад*, у невеликій компанії, де немає комп'ютерів, менеджер може вручну заповнювати звітність, вести записи у журналах та таблицях на папері, а для зведення інформації використовувати лише калькулятор або звичайні письмові інструменти.

Автоматичні інформаційні системи здійснюють всі операції обробки інформації самостійно, без втручання людини. *Наприклад*, автоматичні системи управління технологічними процесами на заводах можуть самостійно регулювати температуру в печах, контролювати потік матеріалів або запускати виробничі лінії згідно з наперед заданими параметрами. Людина в такому

випадку виконує лише функції моніторингу і реагує лише у разі виникнення проблем.

Автоматизовані інформаційні системи поєднують людську участь з використанням технічних засобів, переважно комп'ютерів. Людина взаємодіє із системою, виконує аналітичні функції, приймає остаточні рішення, тоді як комп'ютери відповідають за виконання складних обчислень, збереження великих обсягів даних, генерування звітів та інших стандартних процедур. *Наприклад,* автоматизовані системи управління складом дозволяють оператору вводити дані про надходження товарів, але розрахунок залишків, формування заявок на поповнення запасів і складання звітів здійснюється комп'ютером.

Таким чином, залежно від ступеня автоматизації, інформаційні системи варіюються від повністю ручних, через автоматичні, до автоматизованих, де комп'ютери виконують більшість рутинної роботи, а людина зосереджується на прийнятті рішень і контролі процесів.

Широке впровадження автоматизованих інформаційних систем у процеси управління призвело до появи численних модифікацій таких систем. Їх можна класифікувати за різними критеріями, зокрема за характером використання інформації та сферою застосування.

Класифікація за характером використання інформації поділяє інформаційні системи на:

- **Інформаційно-пошукові системи** — їх основна функція полягає у введенні, організації, зберіганні та видачі даних на запит користувача. Такі системи не виконують складних трансформацій інформації. Наприклад, до цього типу належать системи пошуку книг у бібліотеках або системи бронювання та продажу квитків на залізничні та авіарейси.
- **Інформаційно-вирішальні системи** — вони обробляють дані відповідно до заданих алгоритмів. Цей клас систем поділяється на два підтипи:
    - **Керуючі системи**: використовуються для розрахункових завдань, обробки великих обсягів даних, а також для підготовки інформації, на основі якої людина ухвалює рішення. Прикладами таких систем

є короткострокове планування випуску продукції на виробництві, бухгалтерські облікові системи або системи розрахунку заробітної плати.

- **Дорадчі системи**: орієнтовані на аналіз і представлення інформації, яка не обов'язково перетворюється на безпосередні дії. Вони володіють вищим рівнем інтелекту, оскільки обробляють не лише дані, але й знання. Наприклад, система прогнозування економічних ризиків чи система надання рекомендацій щодо інвестицій.

Приклади:

- У транспортній сфері інформаційно-пошукова система може забезпечувати пошук маршрутів та розкладів у громадському транспорті, надаючи пасажирам актуальну інформацію без складних обчислень.
- У фінансовій галузі керуючі системи можуть розраховувати оптимальні кредитні ставки чи обробляти щоденні транзакції банків, а дорадчі системи — формувати аналітичні звіти про економічну ситуацію, які допомагають у прийнятті стратегічних рішень.

Таким чином, залежно від характеру використання інформації, інформаційні системи виконують різні функції: одні з них забезпечують зручний доступ до даних, інші — допомагають автоматизувати прийняття рішень або надають рекомендації, що використовуються для подальшого аналізу і планування.

Класифікація за сферою застосування дозволяє виділити кілька основних категорій інформаційних систем (ІС) залежно від їх функцій і цілей.

1. ІС організаційного управління.

Ці системи призначені для автоматизації адміністративних і управлінських завдань. Їх використовують у різних сферах, включаючи промислові підприємства, готелі, банки, торговельні компанії та інші організації. Основні функції таких систем: контроль і регулювання діяльності, ведення обліку та аналізу даних, короткострокове і довгострокове планування, управління збутом і постачанням, бухгалтерський облік.

Приклад:

- ERP-системи (наприклад, SAP, Oracle ERP Cloud), які інтегрують фінансовий облік, управління ланцюгами поставок і людськими ресурсами в єдину платформу.
- CRM-системи (Salesforce, Microsoft Dynamics CRM), що допомагають оптимізувати управління відносинами з клієнтами, забезпечуючи зручний доступ до даних клієнтів, історії угод і аналітики.

2. ІС управління технологічними процесами

Ці системи автоматизують функції виробничого персоналу та забезпечують контроль технологічних процесів у реальному часі. Вони широко застосовуються в таких галузях, як металургія, машинобудування, енергетика і хімічна промисловість.

Приклад:

- SCADA-системи (Supervisory Control and Data Acquisition) для моніторингу та управління виробничими процесами. Наприклад, SCADA-системи можуть використовуватися для контролю температури і тиску в трубопроводах на нафтопереробному заводі.
- DCS-системи (Distributed Control Systems), що забезпечують контроль і автоматизацію складних виробничих процесів, таких як виробництво сталі або складання автомобілів.

3. ІС автоматизованого проектування (САПР)

САПР (системи автоматизованого проектування) допомагають автоматизувати проектувальні, інженерні, конструкторські та дизайнерські завдання. Вони використовуються при створенні нових технічних пристроїв, механізмів, будівель або технологій.

Приклад:

- AutoCAD або SolidWorks для створення тривимірних моделей деталей і компонентів.

- BIM-системи (Building Information Modeling) на зразок Autodesk Revit, які дозволяють проектувати будівлі з урахуванням архітектурних, конструктивних і інженерних характеристик.

4. Інтегровані (корпоративні) ІС

Ці системи охоплюють всі аспекти діяльності підприємства: від проектування продукту до його збуту, включаючи управління виробництвом, логістикою, маркетингом і фінансами. Їх реалізація є досить складною і часто передбачає перебудову організаційних процесів.

Приклад:

- Система SAP S/4HANA, яка інтегрує фінансові, логістичні, виробничі та маркетингові модулі, дозволяючи компанії відстежувати весь цикл створення і реалізації продукції.
- Microsoft Dynamics 365, що поєднує в собі можливості управління фінансами, виробництвом і відносинами з клієнтами на єдиній платформі.

Класифікація ІС за сферою застосування дозволяє побачити, як інформаційні системи адаптуються до конкретних потреб галузей і компаній. Кожна з цих категорій виконує свої функції, допомагаючи автоматизувати процеси, підвищувати продуктивність і покращувати якість управлінських та виробничих рішень.

Центральним елементом банку даних є база даних, і більшість характеристик, які використовуються для її класифікації, стосуються саме способів представлення інформації. Залежно від форми, в якій подається інформація, можна виділити кілька типів систем:

- **Візуальні системи:** інформація представлена у вигляді зображень. Це можуть бути текстові дані, статичні графічні об’єкти (малюнки, креслення), фотографії, географічні карти або відеоматеріали.
- **Аудіосистеми:** дані зберігаються й подаються у вигляді звукових файлів.
- **Мультимедійні системи:** інформація комбінує кілька форм подання — наприклад, текст, графіку, звук і відео.

Ця класифікація показує, як саме дані зберігаються в базі даних і як вони можуть бути отримані кінцевими користувачами. Зображення, у цьому контексті, трактується в широкому сенсі: це може бути текстовий контент, графічні матеріали (наприклад, креслення), фотографії або навіть анімовані чи відео-об'єкти.

Таким чином, різноманітність форм представлення інформації в базах даних створює окрему проблематику класифікації, яка вимагає детального розгляду.

Рис. 2.2. Класифікація баз даних

Бази даних, залежно від характеру організації даних, можна поділити на три основні категорії: неструктуровані, частково структуровані й структуровані. Ця класифікація стосується переважно інформації, представленої у символьному вигляді.

Неструктуровані бази даних — це такі бази, де дані не мають чіткої схеми або фіксованої структури. Вони можуть бути організовані, наприклад, у формі семантичних мереж, що дозволяє моделювати зв’язки між різними елементами без жорсткого задання їхньої взаємозалежності.

Приклад: текстові документи у форматі JSON, які зберігаються в NoSQL базах, таких як MongoDB, де немає чіткої схеми і кожен документ може містити свій набір атрибутів.

Частково структуровані бази даних зазвичай містять певний рівень організації, але не настільки жорсткий, як у повністю структурованих БД. Дані в таких базах можуть бути представлені у вигляді звичайного тексту або гіпертексту, що дозволяє організовувати інформацію у вигляді пов'язаних між собою текстових блоків.

Приклад: файли XML, де дані мають теги для структуризації, але формат дозволяє значну гнучкість у визначенні структури і зв'язків.

Структуровані бази даних вимагають ретельного проектування та чіткого опису своєї структури. Дані в таких БД впорядковані у вигляді таблиць із фіксованими стовпцями та рядками, а їх введення можливе лише після визначення структури бази.

Приклад: реляційні бази даних, такі як MySQL чи PostgreSQL, де дані організовані у вигляді таблиць зі строго визначеними колонками і типами даних.

Загалом, вибір між неструктурованими, частково структурованими та структурованими базами даних залежить від потреб конкретного проекту, а також від вимог до гнучкості, швидкості доступу та обсягу даних, що обробляються.

Структуровані бази даних класифікуються за типом моделі даних, що використовується, зокрема, на ієрархічні, мережні, реляційні, змішані та мультимодельні. Ця класифікація застосовується як до самих баз даних, так і до систем управління базами даних (СУБД), які працюють із цими моделями.

У структурованих базах даних розрізняють кілька рівнів інформаційних одиниць, які ієрархічно входять одна в одну. Ці рівні можуть варіюватися навіть у межах одного класу систем. Найпоширеніші рівні, що підтримуються більшістю структурованих баз даних, включають:

- **Поле**: найменша семантична одиниця інформації.

- **Запис**: сукупність полів або інших, більш складних одиниць (залежно від можливостей СУБД).
- **Файл**: сукупність однотипних записів, які утворюють певний набір даних.

Більшість сучасних СУБД також підтримують рівень самої бази даних як сукупності взаємопов'язаних файлів, що дозволяє створювати комплексні системи з багаторівневою структурою.

Приклади використання різних типів структурованих баз даних:

- **Ієрархічна модель:**
  - **Приклад:** IMS (Information Management System) від IBM — використовується в банківських і страхових компаніях для зберігання даних у вигляді ієрархій, наприклад, клієнтські дані, пов'язані з обліковими записами та транзакціями.
- **Мережна модель:**
  - **Приклад:** IDMS (Integrated Database Management System) — застосовується в великих організаціях для управління складними взаємозв'язками між даними, наприклад, управління ланцюгами постачань або великомасштабними проектами.
- **Реляційна модель:**
  - **Приклад:** MySQL або PostgreSQL — широко використовуються в веб-розробці для управління даними користувачів, замовлень, продуктів, що зберігаються в таблицях із чітко визначеними зв'язками між ними.
- **Мультимодельні бази даних:**
  - **Приклад:** ArangoDB — дозволяє працювати одночасно з графовими, документними та реляційними даними, що зручно для складних додатків, які потребують різних підходів до структурування даних.

Вибір моделі бази даних залежить від специфіки задачі: якщо потрібна жорстка ієрархія — використовують ієрархічну модель; для складних зв'язків між даними підходить мережна модель; для стандартних додатків із чітко визначеними таблицями і зв'язками найкраще підходить реляційна модель.

Мультимодельні бази надають додаткову гнучкість, дозволяючи комбінувати підходи залежно від потреб проекту.

У **ієрархічній** (рис. 2.3) та *мережній* (рис. 2.4) моделях можуть бути встановлені зв'язки між різними інформаційними одиницями (ІО), тобто записами з різних файлів. Як видно на відповідних схемах, графічне представлення ієрархічної моделі нагадує структуру дерева. Така модель має єдину вершину, яка виступає коренем дерева і забезпечує вхід до всієї структури. Усі інші вершини, окрім кореня, мають лише одну вхідну вершину, а кількість породжених вершин у загальному випадку не обмежується.

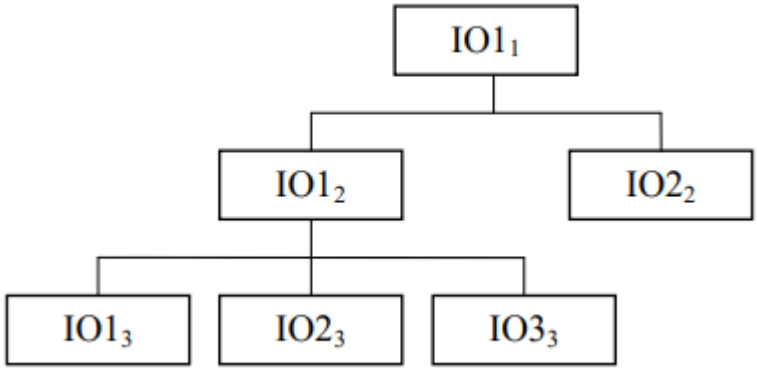

Рис. 2.3. Ієрархічна модель бази даних

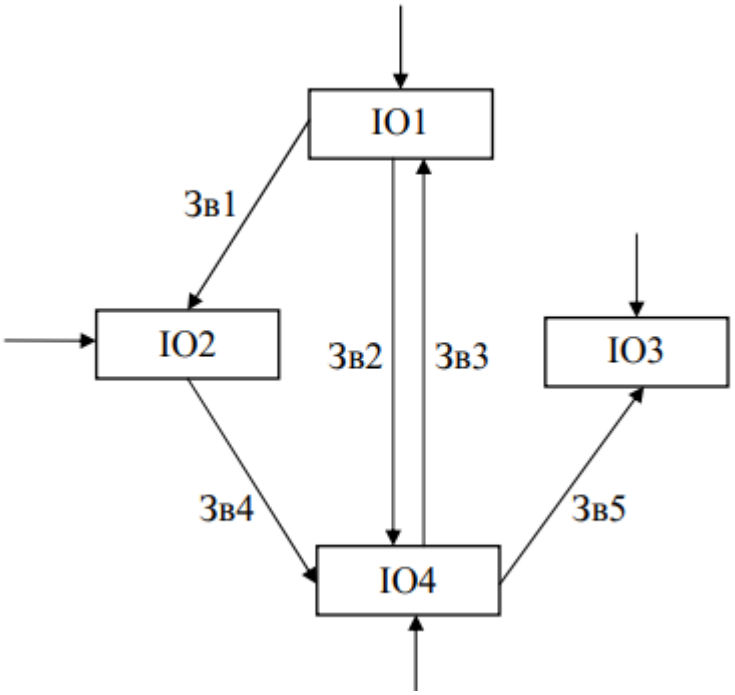

Рис. 2.4. Мережна модель бази даних

Мережна модель даних графічно зображується у вигляді «мережевого» графа, де будь-яка вершина може слугувати точкою входу. Кожна вершина має можливість мати кілька породжених вершин і кілька вихідних. Між окремими вершинами можна визначати кілька зв'язків. У більшості систем управління базами даних (СУБД) реалізовано підтримку простих мережевих структур. На відміну від ієрархічної моделі, в мережній моделі напрямок і характер зв'язків не завжди очевидні, тому при зображенні бази даних слід обов'язково вказувати напрямки зв'язків.

У **реляційних моделях,** на відміну від ієрархічних і мережних, зв'язки між файлами бази даних встановлюються безпосередньо під час виконання запиту. Ці зв'язки базуються на порівнянні значень відповідних полів (полів зв'язку) у зв'язаних файлах. Ще однією важливою особливістю реляційної моделі є те, що її записи мають лінійну структуру й можуть складатися лише з адресних або простих полів. Саме ці особливості визначають ключові підходи до проектування структури бази даних.

3. Сучасні ІТ-технології та інновації

Інформаційні технології (ІТ) є рушійною силою сучасного світу, проникаючи у всі сфери життя – від науки, освіти та медицини до промисловості, фінансів і розваг. Завдяки стрімкому розвитку ІТ з'являються нові інструменти та рішення, що підвищують ефективність бізнесу, автоматизують складні процеси та покращують якість життя людей.

Інновації в ІТ визначаються рядом ключових тенденцій, що формують цифрове майбутнє людства. До таких належать штучний інтелект, хмарні технології, Інтернет речей, квантові обчислення, блокчейн, доповнена і віртуальна реальність, а також кібербезпека. Усі ці технології не тільки вдосконалюються, а й створюють нові можливості для подальшого розвитку світової економіки та суспільства.

Сучасні інформаційні технології та системи являють собою комплекс технічних засобів і програмного забезпечення, призначених для збору, обробки, зберігання, передачі та візуалізації даних. Вони виступають ключовим інструментом для автоматизації процесів, спрощують доступ до інформації та сприяють підвищенню ефективності виконання повсякденних завдань.

Ключові напрями сучасних IT-технологій

1. Штучний інтелект (AI) та машинне навчання (ML)

Штучний інтелект (Artificial Intelligence) та машинне навчання (Machine Learning) є основою сучасної цифрової трансформації. Вони забезпечують аналіз великих обсягів даних, автоматизацію процесів і створення нових продуктів на основі самообучуваних алгоритмів.

Основні напрямки використання AI:

- Автоматизовані голосові асистенти (*Siri, Alexa, Google Assistant*).
- Комп'ютерний зір для розпізнавання облич, тексту та об'єктів.
- AI-генерація текстів, зображень і відео (*ChatGPT, DALL·E, Midjourney*).
- Персоналізовані рекомендації в маркетингу та електронній комерції.
- Автоматизація обслуговування клієнтів через чат-боти.
- Медична діагностика на основі штучного інтелекту.

Прогнозується, що AI продовжить трансформувати галузі, включаючи фінанси, юриспруденцію, освіту та виробництво, допомагаючи швидше та точніше ухвалювати рішення.

2. Хмарні обчислення (Cloud Computing)

Хмарні технології змінюють підходи до зберігання даних, управління інформаційними системами та розробки програмного забезпечення.

Ключові моделі хмарних обчислень:

- **IaaS (Infrastructure as a Service)** – оренда віртуальної інфраструктури (AWS, Google Cloud, Microsoft Azure).
- **PaaS (Platform as a Service)** – середовище для розробки додатків без управління інфраструктурою (Heroku, IBM Cloud).

- **SaaS (Software as a Service)** – використання готових програм через інтернет (Google Drive, Dropbox, Microsoft 365).

Розвиток **хмарних обчислень** сприяє зменшенню витрат на IT-інфраструктуру, підвищенню рівня безпеки даних і гнучкості бізнес-процесів.

3. Інтернет речей (IoT – Internet of Things)

Інтернет речей – це глобальна мережа пристроїв, що обмінюються даними без участі людини. IoT розширює можливості автоматизації у промисловості, розумних містах, медицині та побуті.

Сфери застосування IoT:

- **Розумні будинки** – автоматизовані системи освітлення, клімат-контроль, безпека (системи освітлення, безпеки, термостати).
- **Промисловий IoT (IIoT)** – моніторинг виробничих процесів та оптимізація витрат (моніторинг виробництва та автоматизація підприємств.)
- **Медицина** – носимі пристрої для моніторингу стану здоров'я пацієнтів.
- **Автомобільна індустрія** – автономні транспортні засоби, GPS-системи.

IoT продовжує розвиватися, створюючи "розумні" екосистеми, що підвищують комфорт та ефективність повсякденного життя.

4. Кібербезпека та захист даних

Зі зростанням кіберзагроз компанії все більше інвестують у системи безпеки для захисту конфіденційної інформації.

Сучасні підходи до кібербезпеки:

- Багатофакторна аутентифікація (*MFA, Biometric Security*).
- Використання блокчейну для безпеки фінансових транзакцій.
- Штучний інтелект для виявлення загроз у реальному часі.
- Антивірусні системи нового покоління (*EDR, XDR*).

Кібербезпека стає критично важливим аспектом для підприємств та організацій, оскільки хакерські атаки та витоки даних можуть завдавати значних збитків.

5. Блокчейн та криптовалюти

Блокчейн – це технологія розподіленого реєстру, яка забезпечує безпеку, прозорість і незмінність транзакцій без необхідності централізованого управління. Вона ґрунтується на децентралізованих механізмах верифікації, що дозволяють захищати дані від несанкціонованих змін та зменшують ризик шахрайства.

Завдяки своїй архітектурі блокчейн не тільки підтримує криптовалюти, але й знаходить застосування у широкому спектрі галузей, включаючи фінанси, охорону здоров'я, логістику та навіть урядові структури.

Де застосовується блокчейн?

1. Криптовалюти та фінансові технології

Блокчейн став основою для цифрових валют, які функціонують без центрального банку або фінансового регулятора.

- **Bitcoin (BTC), Ethereum (ETH), Solana (SOL), Ripple (XRP)** – цифрові активи, що використовуються для фінансових операцій.
- **DeFi (Децентралізовані фінанси)** – фінансові сервіси на блокчейні, які забезпечують кредитування, страхування та обмін без посередників (*Uniswap, Aave, Compound*).
- **Стейблкоїни (USDT, USDC, DAI)** – цифрові активи, прив'язані до традиційних валют, що зменшують волатильність крипторинку.

2. Захист авторських прав та цифрової власності

Блокчейн дає змогу захищати інтелектуальну власність, зберігати авторські права та забезпечувати підтвердження власності у цифровому просторі.

- **NFT (невзаємозамінні токени)** – цифрові сертифікати власності на мистецтво, музику, відео, ігрові предмети (*OpenSea, Rarible*).
- **Смарт-контракти** – автоматизовані контракти, що виконуються заздалегідь визначеними умовами без участі третіх осіб (*Ethereum, Cardano, Polkadot*).

➢ **Цифрова ідентифікація** – можливість створення децентралізованих цифрових посвідчень особи, що зменшує ризик підробок та крадіжок особистих даних.

3. Банківська та платіжна сфера

➢ **Миттєві міжнародні платежі** – блокчейн дозволяє здійснювати швидкі, безпечні та дешеві грошові перекази без участі банківських посередників (*Ripple, Stellar*).

➢ **Токенізація активів** – переведення фізичних активів у цифрові токени (нерухомість, акції, облігації).

➢ **Фінансовий аудит та звітність** – прозорість блокчейну дозволяє автоматизувати фінансовий контроль, запобігаючи шахрайству та відмиванню коштів.

4. Логістика та управління ланцюгами постачання

➢ Прозорість поставок – блокчейн дозволяє відстежувати товари від виробника до кінцевого споживача, мінімізуючи ризики підробок та покращуючи контроль якості (IBM Food Trust, VeChain).

➢ Контроль за екологічністю – відстеження походження сировини та оцінка впливу на навколишнє середовище.

➢ Оптимізація документобігу – цифрові контракти та сертифікація без потреби у паперовій документації.

5. Державне управління та цифрова ідентифікація

➢ Голосування на блокчейні – захист виборчих процесів від підробки та забезпечення прозорості результатів.

➢ Децентралізовані реєстри власності – запис прав на землю, нерухомість і корпоративні активи в блокчейн для спрощення операцій.

➢ Автоматизовані податкові звіти – блокчейн може допомогти в боротьбі з ухиленням від сплати податків, автоматично фіксуючи транзакції.

6. Охорона здоров'я та збереження медичних даних

➢ Захист персональних медичних записів – пацієнти можуть контролювати доступ до своїх медичних даних.

- Лікарські поставки – перевірка автентичності медикаментів та відстеження поставок для боротьби з підробками.
- Телемедицина та децентралізовані діагностичні системи – лікарі можуть отримувати захищений доступ до історії хвороб пацієнтів у глобальному масштабі.

Переваги та виклики блокчейн-технологій

Переваги блокчейну:

- Децентралізація – відсутність центрального органу управління робить систему менш вразливою до зловживань.
- Безпека – криптографічний захист гарантує цілісність даних.
- Прозорість – всі транзакції записуються у відкритий реєстр, що ускладнює шахрайство.
- Автоматизація – смарт-контракти дозволяють виконувати угоди без участі посередників.

Виклики та обмеження:

- ✓ Висока енергозатратність – майнінг криптовалют потребує значних обчислювальних ресурсів.
- ✓ Правове регулювання – у багатьох країнах правовий статус блокчейну та криптовалют залишається невизначеним.
- ✓ Масштабованість – деякі блокчейн-мережі мають обмежену пропускну здатність, що уповільнює транзакції.
- ✓ Втрата доступу до активів – якщо користувач загубить ключі доступу до криптовалютного гаманця, відновити кошти буде неможливо.

Перспективи розвитку блокчейну

Попри виклики, блокчейн-технології продовжують вдосконалюватися. Серед основних напрямків їхнього розвитку можна виділити:

- **Перехід до екологічних механізмів підтвердження транзакцій** (Proof of Stake замість Proof of Work, як у новій версії Ethereum 2.0).
- **Інтеграція блокчейну в урядові структури** – для захисту державних реєстрів та спрощення бюрократії.

➢ **Розвиток CBDC (Central Bank Digital Currency)** – державні цифрові валюти (цифровий долар, євро, юань).

➢ **Спрощення взаємодії між блокчейнами** – створення платформ для швидкого обміну цифровими активами між різними мережами (Polkadot, Cosmos).

➢ **Збільшення рівня безпеки та конфіденційності** – впровадження нових алгоритмів шифрування для збереження анонімності користувачів.

Блокчейн – це не лише основа для криптовалют, а й революційна технологія, яка змінює фінансову, медичну, юридичну та державну сфери. Він сприяє підвищенню прозорості, безпеки та ефективності бізнес-процесів.

Попри певні виклики, розвиток блокчейну продовжується, і ця технологія має потенціал стати ключовою складовою цифрової економіки майбутнього.

6. Розвиток 5G і майбутніх технологій зв’язку

Впровадження 5G дозволяє значно збільшити швидкість передачі даних і знизити затримку в мережах, що відкриває нові можливості для індустрії.

Очікувані переваги:

- Поліпшена якість відеозв'язку та онлайн-стрімінгу.
- Підтримка автономного транспорту (безпечне управління автопілотом).
- Розвиток IoT та розширені можливості для "розумних" міст.

Впровадження технології 5G (п’ятого покоління мобільного зв’язку) кардинально змінює способи обміну даними, роблячи зв’язок швидшим, стабільнішим і безпечнішим. Швидкість передачі даних у 5G-мережах може досягати 10 Гбіт/с, що у десятки разів перевищує можливості 4G.

Завдяки цьому 5G відкриває нові можливості для інноваційних галузей, таких як автономний транспорт, розумні міста, Інтернет речей (IoT), дистанційна медицина та індустрія розваг.

Приклади практичного застосування 5G

1. Відеозв’язок, онлайн-стрімінг та розваги

5G забезпечує миттєве завантаження контенту та відсутність затримок під час відеозв’язку та стрімінгу.

➢ **YouTube, Netflix, Twitch** – якісне потокове відео у форматі **8K UHD** без буферизації**.**

➢ **VR та AR-додатки** – висока швидкість передачі даних дозволяє переглядати **360° відео** без затримок (*Oculus, Apple Vision Pro*).

➢ **Хмарний геймінг (Cloud Gaming)** – сервіси на основі 5G, такі як **Google Stadia, GeForce Now, Xbox Cloud Gaming**, дозволяють запускати потужні ігри без консолей або ПК.

2. Автономний транспорт і автомобільна індустрія

Технологія 5G дозволяє бездротово обмінюватися даними в режимі реального часу, що критично важливо для автономних транспортних засобів.

➢ **Tesla, Waymo, Mercedes-Benz** – системи автопілотування, які використовують 5G для зниження часу реакції автомобіля.

➢ **Розумний транспортний рух** – 5G допомагає інтегрувати **V2X (Vehicle-to-Everything)** – обмін інформацією між автомобілями, світлофорами, камерами спостереження.

➢ **Дистанційне керування транспортом** – у деяких країнах уже тестуються **автономні автобуси та вантажівки**, які працюють без водія (*Baidu, Einride*).

3. Розвиток Інтернету речей (IoT) та "розумних" міст

Інфраструктура 5G сприяє підключенню мільярдів пристроїв до єдиної мережі, дозволяючи створювати розумні міста та автоматизовані промислові рішення.

➢ **"Розумне освітлення"** – інтелектуальні вуличні ліхтарі, які автоматично регулюють яскравість залежно від часу доби та руху транспорту.

➢ **Системи контролю якості повітря** – датчики, що передають інформацію про рівень забруднення в реальному часі (проєкти в **Сінгапурі, Лондоні, Стокгольмі**).

➢ **Розумні будівлі та енергосистеми** – система 5G дозволяє **автоматизувати електропостачання**, оптимізуючи витрати енергії.

➢ **IoT-кампанії, такі як Huawei, Ericsson, Nokia** – вже впроваджують 5G для управління міською інфраструктурою.

4. Дистанційна медицина та телемедицина

5G значно покращує швидкість і стабільність медичних послуг, відкриваючи нові можливості для дистанційного лікування.

➢ **Хірургічні операції на відстані** – 5G забезпечує **наднизьку затримку сигналу**, що дозволяє лікарям проводити **роботизовані операції віддалено** (*Da Vinci Surgical System*).

➢ **Телемедицина** – пацієнти можуть проходити консультації **онлайн у режимі реального часу** без втрати якості зв'язку (*Mayo Clinic, Ping An Good Doctor*).

➢ **Медичні датчики та пристрої** – носимі гаджети (Apple Watch, Fitbit) можуть передавати **дані про серцевий ритм, рівень кисню в крові, артеріальний тиск** лікарям без затримок.

5. Вплив 5G на виробництво та автоматизацію

У промисловості 5G сприяє індустрії 4.0, покращуючи управління виробничими процесами та автоматизацію підприємств.

➢ **Роботизовані заводи** – підприємства, такі як **BMW, Tesla, Siemens**, впроваджують **5G-мережі для керування роботами** бездротово.

➢ **Предиктивне обслуговування обладнання** – датчики у виробничих лініях повідомляють про необхідність технічного обслуговування **до появи поломки**.

➢ **Розширена реальність (AR) на виробництві** – інженери можуть використовувати **AR-окуляри** для дистанційного ремонту або навчання персоналу (*Microsoft HoloLens, Magic Leap*).

6. Освіта та цифрове навчання

Технологія 5G кардинально змінює підходи до навчання, роблячи освіту більш доступною та інтерактивною.

➢ **Віртуальні класи та лекції у форматі VR** – використання 5G для забезпечення інтерактивного навчання у віртуальних середовищах.

➢ **Доповнена реальність у навчанні** – використання **AR-додатків** для вивчення біології, фізики, хімії у візуальному форматі.

➢ **Швидке завантаження освітніх матеріалів** – учні та студенти можуть отримувати **онлайн-доступ до високоякісних курсів без затримок** (*Coursera, Khan Academy*).

7. Розширена (AR) та віртуальна реальність (VR)

Технології розширеної (Augmented Reality – AR) та віртуальної реальності (Virtual Reality – VR) стають важливою складовою різних галузей, виходячи далеко за межі ігрової індустрії. Вони допомагають покращити взаємодію з інформацією, створювати реалістичні симуляції та значно підвищувати ефективність роботи у сфері освіти, медицини, будівництва, дизайну та промисловості.

Приклади застосування AR та VR у різних сферах

1. Освіта та навчання

Розширена і віртуальна реальність кардинально змінюють традиційні методи навчання, роблячи процес інтерактивним та захопливим.

➢ **Віртуальні тури** – студенти можуть відвідувати музеї, історичні місця або навіть інші планети у VR (*Google Expeditions, Louvre VR*).

➢ **Інтерактивні лабораторії** – **Labster, zSpace** надають студентам можливість проводити експерименти в хімії, фізиці та біології без реальних лабораторних ризиків.

➢ **Доповнена реальність у підручниках** – використання **AR-додатків**, які дозволяють вивчати 3D-моделі серця, ДНК або механізмів роботи машин (*Curiscope, Merge Cube*).

2. Охорона здоров'я та медицина

Медична галузь активно використовує AR і VR для навчання, діагностики та лікування пацієнтів.

➢ **VR-симуляції для лікарів** – програми **Osso VR, Touch Surgery** дозволяють хірургам практикувати складні операції у віртуальному середовищі перед реальним втручанням.

➢ **Доповнена реальність у хірургії** – технології **Microsoft HoloLens** та **Augmedics xVision** допомагають лікарям бачити 3D-візуалізацію органів під час операції.

➢ **VR-терапія для пацієнтів** – лікування посттравматичного синдрому (PTSD) через занурення в контрольовані сценарії (*Bravemind, Psious*).

➢ **Реабілітація після травм** – VR-системи **MindMaze, VRHealth** допомагають пацієнтам відновлювати рухові функції.

3. Будівництво та архітектура

AR та VR кардинально змінюють підходи до проєктування та будівництва будівель, дозволяючи бачити кінцевий результат ще до початку роботи.

➢ **Проєктування будівель** – **Autodesk Revit + VR**, **SketchUp VR** дозволяють архітекторам створювати 3D-моделі будівель, які можна переглядати у віртуальній реальності.

➢ **Доповнена реальність у будівництві** – **Trimble Connect AR, Magic Leap** використовують AR для візуалізації інженерних креслень на реальному будівельному майданчику.

➢ **Спільна робота над дизайном** – інженери, дизайнери та клієнти можуть взаємодіяти з 3D-моделями у реальному часі.

4. Роздрібна торгівля та електронна комерція

➢ **Віртуальні примірочні** – покупці можуть "приміряти" одяг або косметику через AR (*Sephora Virtual Artist, IKEA Place, Nike Fit*).

➢ **Доповнена реальність у магазинах** – **Zara, L'Oréal** використовують AR-дисплеї для демонстрації товарів у магазинах без фізичного контакту.

➢ **Віртуальні шоуруми** – автомобільні бренди, такі як **Audi, Tesla**, використовують VR для демонстрації моделей авто в цифровому середовищі.

5. Розвиток індустрії розваг та спорту

➢ **Ігри у VR та AR** – такі ігри, як **Half-Life: Alyx, Beat Saber, Pokémon GO, The Walking Dead: Saints & Sinners**, створюють новий рівень занурення.

➢ **Спортивні тренування** – платформи **STRIVR, Rezzil** використовують VR для тренування спортсменів у футболі, баскетболі, боксі.

➢ **Концерти та заходи у VR** – платформи, як-от **Horizon Venues (Meta), WaveXR**, дозволяють організовувати віртуальні виступи та зустрічі.

6. Промисловість та виробництво

➢ **VR-навчання для робітників** – **Siemens, Boeing** використовують віртуальну реальність для навчання персоналу без реальних ризиків.

➢ **Віддалене управління обладнанням** – **Microsoft HoloLens** використовується для візуалізації роботи складних машин та діагностики несправностей.

➢ **Ремонтні роботи з AR** – компанії, як **Bosch, General Electric**, використовують доповнену реальність для обслуговування техніки та ремонту складного обладнання.

7. Автомобільна галузь та транспорт

➢ **Проєктування автомобілів** – **Ford, BMW, Volvo** використовують VR для моделювання авто до їх виробництва.

➢ **Тренування водіїв та пілотів** – **VR-симулятори** використовуються у школах водіння та авіаційних тренажерах (Boeing, Airbus).

➢ **AR-головні дисплеї у транспорті** – сучасні автомобілі оснащуються доповненою реальністю у проекційних дисплеях (Mercedes-Benz MBUX, Tesla).

Перспективи розвитку AR та VR

– **Поєднання з штучним інтелектом (AI)** – інтеграція AI в AR/VR для покращення персоналізації досвіду.

– **Збільшення використання у будівництві** – AR-дисплеї для інженерів та архітекторів.

– **Розвиток метавсесвіту** – поява віртуальних світів для роботи, навчання та соціальної взаємодії (*Meta Horizon Worlds, Microsoft Mesh*).

– **Більш реалістичні медичні симуляції** – навчання лікарів на VR-моделях з високою деталізацією.

– **Віртуальний шопінг** – подальше впровадження AR в онлайн-магазини для інтерактивного вибору товарів.

AR та VR стають не просто розважальними технологіями, а потужними інструментами для бізнесу, науки, освіти та медицини. Вони покращують навчальний процес, допомагають у медицині, оптимізують виробництво та відкривають нові можливості для споживчого ринку.

У майбутньому розширена і віртуальна реальність стануть ще більш інтегрованими у наше життя, надаючи унікальні способи взаємодії з інформацією та навколишнім світом.

Сучасні IT-технології є рушійною силою цифрової трансформації, що змінює світ навколо нас. Вони сприяють автоматизації бізнесу, покращенню якості життя, ефективнішому використанню ресурсів і відкривають нові горизонти для інновацій.

Подальший розвиток інформаційних технологій забезпечить ще більше можливостей для штучного інтелекту, безпечного зберігання даних, швидшого обміну інформацією та цифровізації ключових галузей. Світ продовжить рухатися в напрямку більш розумних, екологічно чистих та ефективних рішень, що змінять наш спосіб життя.

Контрольні питання

1. Що таке інформаційні технології та які їх основні складові?
2. Які основні критерії використовуються для класифікації інформаційних систем?
3. Які типи інформаційних систем існують за ступенем структурованості завдань?
4. У чому полягає різниця між інформаційно-пошуковими та інформаційно-вирішальними системами?
5. Які інноваційні IT-технології ви знаєте, і як вони впливають на розвиток бізнесу та суспільства?
6. Як розподілені системи змінюють підхід до обробки та зберігання даних?
7. Які переваги і виклики пов’язані з використанням хмарних технологій у сучасних інформаційних системах?

Тема 3.
АРХІТЕКТУРА ІНФОРМАЦІЙНИХ СИСТЕМ

Метою лекції є ознайомлення з принципами побудови, структурними компонентами та підходами до розробки інформаційних систем. Здобувачі розглянуть основні архітектурні моделі, зокрема клієнт-серверну, розподілену та хмарну, а також особливості корпоративних та веб-орієнтованих систем. Вивчення переваг і недоліків різних архітектур дозволить обґрунтовано обирати оптимальні рішення для конкретних завдань. У результаті студенти здобудуть розуміння архітектурного проєктування та його ролі в ефективному функціонуванні інформаційних систем.

План лекції

1. Загальні поняття та класифікація архітектур інформаційних систем.
2. Основні компоненти архітектури інформаційних систем.
3. Типи архітектур інформаційних систем.
4. Переваги та недоліки різних архітектурних підходів.

1. Загальні поняття та класифікація архітектур інформаційних систем

Сучасний світ характеризується стрімким розвитком інформаційних технологій, що призводить до зростання потреби у ефективних та надійних інформаційних системах. Вони відіграють ключову роль у функціонуванні підприємств, державних установ, фінансових організацій, медичних закладів та багатьох інших сфер діяльності.

Інформаційні системи є комплексом апаратних і програмних засобів, що забезпечують збір, обробку, зберігання, аналіз та передачу інформації. Важливим аспектом їх ефективного функціонування є **архітектура**, яка визначає структуру, взаємодію та логіку роботи всіх компонентів системи. Саме правильний вибір архітектурного підходу дозволяє оптимізувати

продуктивність, масштабованість, безпеку та зручність використання інформаційної системи [12].

Залежно від завдань та сфери застосування існують різні архітектурні моделі, такі як **однорівневі, багаторівневі, клієнт-серверні, хмарні та мікросервісні системи**. Вибір архітектури залежить від вимог до швидкодії, обсягу оброблюваних даних, можливості інтеграції з іншими системами та рівня безпеки.

Розглянемо основні принципи побудови архітектури інформаційних систем, їх класифікація, особливості взаємодії компонентів, а також сучасні тенденції розвитку, що визначають майбутнє цифрових технологій. Розуміння архітектурних підходів є важливим для фахівців у сфері інформаційних технологій, оскільки це дозволяє розробляти ефективні рішення для автоматизації бізнес-процесів та впровадження інновацій у різних сферах діяльності.

Архітектура інформаційних систем – це сукупність концептуальних і технічних рішень, що визначають структуру, функціональність, взаємодію та управління компонентами інформаційної системи. Вона охоплює організацію апаратного, програмного, інформаційного та мережевого забезпечення, а також методи обробки, зберігання, передавання та аналізу даних.

Архітектура інформаційних систем визначає логічну і фізичну організацію ресурсів, механізми їхньої взаємодії, а також забезпечує ефективність, масштабованість, продуктивність і безпеку системи. Вона є основою для планування, розробки, впровадження та супроводу інформаційних систем, що застосовуються у бізнесі, промисловості, науці, медицині та державному управлінні.

Рівень розвитку сучасних технологій настільки високий, що дозволяє побудувати інформаційну систему будь-якого масштабу, складності й функціональності. Однак, з огляду на вимоги бізнесу, засновані на показниках різних бізнес-оцінок, виникають додаткові складнооі, вирішення яких зводиться до забезпечення раціонального підходу до процесу проектування,

реалізації й подальшій експлуатації інформаційних систем. Виходячи із цього, можна однозначно вважати обрану архітектуру одним з основних показників ефективності створюваної інформаційної системи, а, отже, і успішності бізнесу.

У результаті, архітектуру інформаційної системи можна описати як концепцію, що визначає модель, структуру, виконувані функції й взаємозв'язок компонентів інформаційної системи.

Процедура вибору архітектури для проектованої інформаційної системи, у ринкових умовах, зводиться до визначення вартості володіння нею. Вартість володіння інформаційною системою складається із планових витрат і вартості ризиків. Планові витрати містять у собі вартість технічного обслуговування, модернізації, зарплату обслуговуючого персоналу й т.д.

Сукупна вартість ризиків визначається з вартості всіх типів ризиків, їхніх імовірностей і матрицею відповідності між ними. Сама ж матриця відповідності визначається обраною архітектурою інформаційної системи.

Можна виділити найбільш важливі типи ризиків:

– проектні ризики (ризики при створенні системи);

– ризики розробки (помилки, недостатня оптимізація);

– технічні ризики (простої, відмови, втрата даних);

– бізнес-ризики (виникають через технічні ризики й пов'язані з експлуатацією системи);

– невизначеності (пов'язані з варіативністю бізнесів-процесів і складаються з необхідності внесення змін у систему й неоптимальну процедуру функціонування);

– операційні (мають на увазі невиконання набору операцій, можуть виникати через технічні ризики й бути ініціаторами бізнесів-ризиків).

Ключові аспекти архітектури інформаційних систем

1. **Функціональна структура** – визначає, які компоненти входять до складу системи та які функції вони виконують.
2. **Рівні обробки даних** – від фізичного зберігання до рівня бізнес-логіки та користувацького інтерфейсу.

3. **Взаємодія компонентів** – правила обміну даними між серверами, клієнтськими пристроями та базами даних.
4. **Типи архітектур** – однорівнева, двохрівнева, багаторівнева, клієнт-серверна, мікросервісна, хмарна тощо.
5. **Забезпечення продуктивності та безпеки** – методи оптимізації швидкодії, стійкість до збоїв, управління доступом та захист інформації.

Розглядаючи архітектуру великих організацій, прийнято використовувати поняття «корпоративна архітектура».

Її можна представити у вигляді сукупності декількох типів архітектур:

– бізнес архітектура (Business architecture);

– IT-архітектура (Information Technology architecture);

– архітектура даних (Data architecture); програмна архітектура (Software architecture);

– технічна архітектура (Hardware architecture).

Модель корпоративної архітектури представлена на рис. 3.1

Рис. 3.1. Модель корпоративної інформаційної системи

Джерело: побудовано на основі [23]

Технічна архітектура є першим рівнем архітектури інформаційної системи. Вона описує всі апаратні засоби, що використовуються при виконанні заявленого набору функцій, а також включає засобу забезпечення мережної взаємодії й надійності. У технічній архітектурі вказуються периферійні

пристрої, мережні комутатори й маршрутизатори, жорсткі диски, оперативна пам’ять, процесори, сполучні кабелі, джерела безперебійного живлення й т.п.

Програмна архітектура являє собою сукупність комп’ютерних програм, призначених для рішення конкретних завдань. Даний тип архітектури призначений для опису додатків, що входять до складу інформаційної системи. На даному рівні описують програмні інтерфейси, компоненти й поведінку. Архітектура даних поєднує в собі як фізичні сховища даних, так і засоби керування даними. Крім того, до неї входять логічні сховища даних, а при орієнтованості розглянутої компанії на роботу зі знаннями, може бути виділений окремий рівень – архітектура знань (Knowledge architecture). На цьому рівні описуються логічні й фізичні моделі даних, визначаються правила цілісності, складаються обмеження для даних.

Слід особливо виділити рівень IT-архітектури, оскільки він є сполучним. На ньому формується базовий набір сервісів, які використовуються як на рівні програмної архітектури, так і на рівні архітектури даних. Якщо будь-яка особливість функціонування для цих двох рівнів не була передбачена, то сильно зростає ймовірність збоїв у роботі, а, отже, втрат для бізнесу. У деяких випадках неможливо розділити IT-архітектуру й архітектуру окремого додатка. Таке можливо при високому ступені інтеграції додатків.

Прикладом IT архітектури може служити SharePoint від компанії Microsoft. Цей продукт надає сервіси для спільної роботи й зберігання інформації, що є дуже важливим аспектом функціонування будь якої компанії. Його базові системні модулі відносяться до IT-архітектури, а користувальницькі – до програмного. Основною функцією IT-архітектури є забезпечення функціонування важливих бізнесів-додатків для досягнення позначених бізнесів-цілей. Якщо деяка функція потрібна відразу в декількох додатках, то її доцільно перенести на рівень IT-архітектури, тим самим підвищивши інтеграцію системи й знизити складність архітектури додатків.

Останнім в ієрархії є рівень бізнес-архітектури або архітектури бізнесів-процесів. На цьому рівні визначаються стратегії ведення бізнесу, способи

керування, принципи організації й ключові процеси, що представляють для бізнесу величезну важливість.

Тенденції розвитку платформних архітектур

У сучасних інформаційних системах можна виокремити три основні напрями еволюції платформних архітектур:

1. **Автономні** – системи, що функціонують незалежно, без підключення до зовнішніх ресурсів.
2. **Централізовані** – архітектури, де всі обчислення та управління даними здійснюються на одному основному сервері або кластері серверів.
3. **Розподілені** – моделі, у яких обробка даних та виконання завдань розподіляється між декількома вузлами мережі, що підвищує масштабованість і відмовостійкість системи.

Автономна архітектура передбачає розміщення всіх функціональних компонентів системи на одному фізичному пристрої, наприклад, персональному комп'ютері, без необхідності підключення до зовнішніх ресурсів. Такі системи функціонують самостійно та не потребують інтеграції з іншими інформаційними середовищами. Прикладами можуть бути системні утиліти, текстові редактори та прості корпоративні додатки. Варто зазначити, що у процесі створення корпоративної інформаційної системи зазвичай не передбачається ізольованих модулів або вузлів, оскільки це може ускладнювати загальне управління. Проте у певних випадках автономні рішення використовуються для підвищення рівня безпеки або надійності роботи окремих компонентів.

Централізована архітектура передбачає виконання всіх ключових завдань на єдиному обчислювальному вузлі, потужностей якого достатньо для обслуговування всіх підключених користувачів. У такій моделі функціональні компоненти системи розподіляються між основним обчислювальним сервером (мейнфреймом) і термінальними станціями, що використовуються кінцевими користувачами. Основне навантаження – обробка даних, управління ресурсами та виконання прикладних програм – покладається на сервер, тоді як термінали

виконують лише функцію виводу інформації без додаткових обчислювальних можливостей. Такий підхід забезпечує централізований контроль за даними, високий рівень безпеки та ефективність управління системою.

Розвиток розподілених архітектур став можливим завдяки стрімкому прогресу в галузі апаратного та програмного забезпечення. У таких системах функціональні компоненти інформаційної системи розподіляються між різними вузлами відповідно до їхніх завдань та ролей. Це дозволяє підвищити ефективність обробки даних, забезпечити масштабованість та підвищити надійність системи.

Основними характеристиками розподілених архітектур є:

- **Спільне використання ресурсів** – доступ до апаратних і програмних ресурсів здійснюється декількома вузлами одночасно.
- **Відкритість** – можливість розширення системи шляхом додавання нових ресурсів та технологій.
- **Паралельність** – підтримка одночасного виконання процесів на різних вузлах із можливістю їхньої взаємодії.
- **Масштабованість** – здатність системи адаптуватися до зростаючих обсягів даних та додавати нові функціональні можливості.
- **Відмовостійкість** – здатність системи зберігати працездатність навіть при виході з ладу окремих компонентів за рахунок дублювання даних і резервування ресурсів.

Розподілені архітектури широко застосовуються у складних інформаційних системах, таких як хмарні сервіси, блокчейн-технології, великі корпоративні мережі та високонавантажені онлайн-платформи.

З розвитком інформаційних технологій архітектура інформаційних систем стає дедалі складнішою, включаючи штучний інтелект, аналіз великих даних (Big Data), Інтернет речей (IoT) та блокчейн. Інноваційні підходи дозволяють створювати більш адаптивні та ефективні системи, що відповідають вимогам сучасного цифрового середовища.

2. Основні компоненти архітектури інформаційних систем

Інформаційні системи є невід'ємною частиною сучасного бізнесу та управління. Вони дозволяють організаціям ефективно обробляти дані, підтримувати ключові процеси, взаємодіяти зі споживачами та партнерами, а також забезпечувати прийняття обґрунтованих рішень. Важливо розуміти основні компоненти, які формують архітектуру інформаційних систем, адже саме вони визначають, наскільки злагоджено і стабільно працює система, наскільки легко її можна масштабувати чи адаптувати до нових умов. У цій статті розглянемо основні елементи архітектури інформаційних систем, їхні функції та значення.

Архітектура інформаційних систем зазвичай складається з кількох основних компонентів, які забезпечують злагоджену роботу системи та ефективну взаємодію між її частинами. До ключових компонентів належать:

1. **Інфраструктурний рівень:**

Це апаратні та мережеві ресурси, що забезпечують основу для функціонування інформаційної системи. Сюди входять сервери, сховища даних, мережеве обладнання, а також засоби віртуалізації та хмарні сервіси.

2. **База даних та сховища даних:**

Включають системи управління базами даних (СУБД) та структури, де зберігаються, організовуються та керуються дані. Вони дозволяють ефективно зберігати та швидко отримувати інформацію для підтримки бізнес-процесів.

3. **Додатки та програмне забезпечення:**

Сюди входять прикладні програми, які виконують конкретні функції, такі як обробка транзакцій, аналітика даних, управління документами, а також бізнес-логіка, що реалізує основні процеси організації.

4. **Інтерфейси користувача та взаємодії:**

Засоби, за допомогою яких користувачі взаємодіють із системою. Це можуть бути веб-інтерфейси, мобільні додатки, настільні програми чи спеціалізовані панелі управління.

5. **Сервіси інтеграції:**

Компоненти, що забезпечують взаємодію між різними модулями, системами чи сторонніми сервісами. Включають API, сервісно-орієнтовану архітектуру (SOA), мікросервіси чи посередницьке програмне забезпечення для інтеграції.

6. **Безпека та управління доступом:**

Включають механізми автентифікації, авторизації, шифрування даних, а також моніторинг та виявлення загроз. Цей рівень гарантує, що система захищена від несанкціонованого доступу та зломів [10].

Кожен з цих компонентів відіграє важливу роль у забезпеченні надійності, продуктивності та функціональності інформаційної системи.

Слід відмітити, що розподіл функціональних компонентів системи при використанні клієнтсерверної архітектури може виконуватися декількома способами, зображеними на рис. 3.2.

Рис.3.2. Приклади розподілу функціональних компонентів

Джерело: побудовано на основі [23]

Архітектура web-додатків або архітектура web-сервісів має на увазі надання деякого сервісу, доступного в мережі Internet, через спеціальний додаток. Основою для надання таких послуг служать відкриті стандарти й протоколи SOAP, UDDI й WSDL. SOAP (Simple Object Access Protocol) визначає

формат запитів до web-сервісів. Дані між клієнтом і сервісом передаються в SOAP-конвертах (envelops). WSDL (Web Service Description Language) служить для опису інтерфейсу надаваного сервісу. Перед розгортанням web-додатка потрібно скласти його опис, указати адреса, список підтримуваних протоколів, перелік припустимих операцій, а так само формати запитів і відповідей. UDDI (Universal Description, Discovery and Integration) являє собою протокол пошуку web-сервісів у мережі Internet. Пошук здійснюється за їхніми описами, які розташовані в спеціальному реєстрі.

Архітектура таких сервісів схожа за концепцією з багатоланкової клієнт-серверною, однак, сервера додатків і баз даних розташовуються в мережі Internet.

Більшість процесів, пов’язаних із проєктуванням інформаційних систем, ґрунтуються на використанні досвіду попередніх схожих проєктів. Труднощі з’являються лише в окремих випадках, коли неможливо застосувати вже існуючі рішення чи напрацювання. Архітектурний стиль, у цьому контексті, можна описати як спільні принципи й підходи до виконання завдань, засновані на набутому досвіді. Він визначає складові системи, а також способи й умови їхньої взаємодії. Водночас, незважаючи на численні спроби, загальноприйнятих стандартних мов для опису архітектур досі немає. Архітектурні стилі, як правило, поділяють на п’ять основних категорій (рис. 3.3):

- Потоки даних (Data Flow Systems)
- Виклик із поверненням (Call-and-Return Systems)
- Незалежні компоненти (Independent Component Systems)
- Централізовані дані (Data-Centric Systems)
- Віртуальні машини (Virtual Machines).

Системи потоків даних, у свою чергу, поділяються на:

- системи пакетно-послідовної обробки (Batch Sequential Systems);
- системи типу конвеєри й фільтри (Pipe and Filter Architecture).

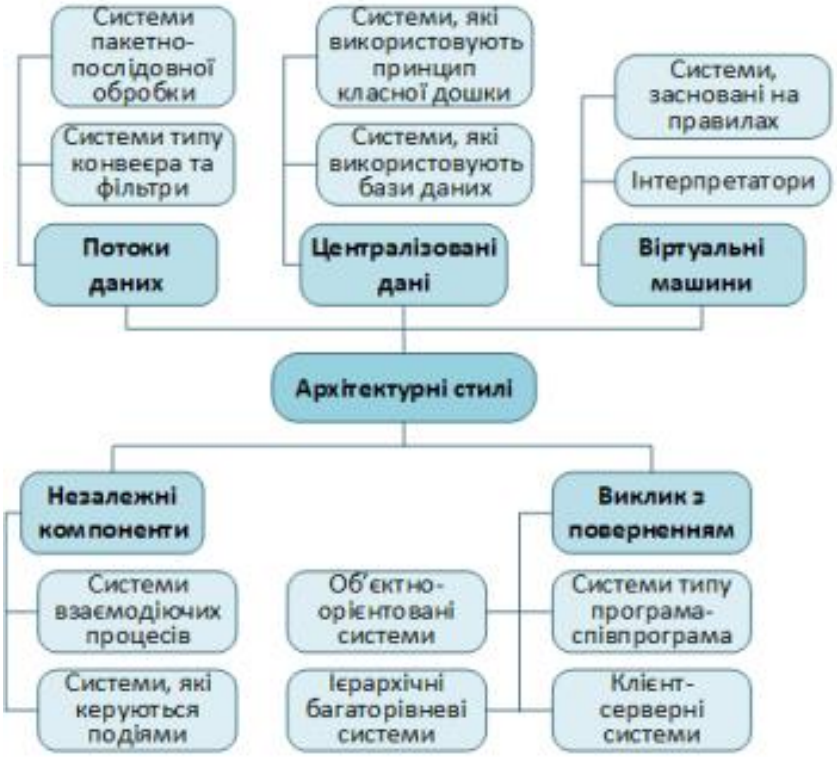

Рис.3.3. Класифікація архітектурних стилів

Джерело: побудовано на основі [23]

Таким чином, представлені на рисунку архітектурні стилі охоплюють широкий спектр підходів до проєктування інформаційних систем. Вони демонструють різноманіття способів організації компонентів, їхньої взаємодії та управління даними. Кожна категорія має свої особливості, сильні сторони й обмеження, що дозволяє розробникам обирати оптимальний стиль залежно від специфіки проєкту та поставлених завдань. Відповідний вибір архітектурного стилю забезпечує стабільність, масштабованість та ефективність інформаційної системи.

3. Типи архітектур інформаційних систем

Архітектура інформаційної системи визначає, як саме організовані її компоненти, як вони взаємодіють між собою та з користувачами, а також як зберігаються та обробляються дані. Вибір архітектури є одним із ключових рішень у процесі проєктування, оскільки вона впливає на масштабованість,

продуктивність, зручність обслуговування та стабільність роботи системи. Основні типи архітектур, які часто використовуються на практиці, включають:

1. **Монолітна архітектура:**

У цьому підході вся логіка системи інтегрована в єдину велику програму або застосунок. Такий варіант зручний для невеликих проєктів, адже забезпечує відносну простоту розробки та швидкість впровадження. Однак, у міру зростання системи моноліт стає важчим у підтримці, оскільки будь-яка зміна в одному модулі може вплинути на весь проєкт.

2. **Клієнт-серверна архітектура:**

Передбачає чітке розділення системи на клієнтську частину, яка відповідає за взаємодію з користувачем, і серверну частину, що забезпечує зберігання даних та виконання бізнес-логіки. Це один із найпоширеніших підходів, що дозволяє спрощувати модернізацію, додавання нових функцій і розподіл навантаження.

3. **Трирівнева (N-рівнева) архітектура:**

Цей підхід додає ще більше модульності, розділяючи систему на кілька рівнів:

- Рівень презентації (взаємодія з користувачем)
- Логічний рівень (обробка даних, алгоритми, бізнес-правила)
- Рівень даних (зберігання та управління базами даних). Такий розподіл дозволяє незалежно змінювати або оновлювати окремі частини системи, що особливо важливо для великих, довготривалих проєктів.

4. **Мікросервісна архітектура:**

Усе більше популярності набуває мікросервісний підхід, який передбачає розділення системи на невеликі автономні сервіси. Кожен сервіс відповідає за конкретну функцію і взаємодіє з іншими через API. Це дозволяє легко масштабувати окремі компоненти, підвищує гнучкість в розробці та підтримці, але вимагає чіткої організації інфраструктури та засобів оркестрації.

5. **Сервісно-орієнтована архітектура (SOA).**

Дозволяє об’єднувати різні системи через стандартизовані інтерфейси. Вона особливо корисна для великих підприємств, де є потреба у взаємодії кількох

незалежних програм. Хоча SOA не є такою гнучкою, як мікросервіси, вона забезпечує стабільність і стандартизацію у великомасштабних інфраструктурах.

6. **Подіє-орієнтована архітектура:**

У системах такого типу компоненти взаємодіють через події. Іншими словами, замість безпосередніх викликів між модулями вони реагують на певні тригери (наприклад, нові дані чи користувацькі дії). Такий підхід дозволяє швидко адаптувати систему до змін, полегшує інтеграцію нових компонентів і забезпечує високу гнучкість.

7. **Хмарна архітектура:**

Вона використовує ресурси, що надаються хмарними провайдерами, для зберігання даних, виконання обчислень і розподілу навантаження. Хмарна архітектура дозволяє швидко масштабувати систему, забезпечує високу доступність і знижує витрати на фізичну інфраструктуру.

Розширення типів архітектур:

Сучасні інформаційні системи дедалі частіше поєднують кілька підходів. Наприклад, мікросервіси можуть функціонувати на основі подіє-орієнтованої архітектури, а компоненти, розгорнуті в хмарі, взаємодіють через клієнт-серверний принцип. Такі комбіновані підходи допомагають оптимально використовувати ресурси, забезпечувати високу гнучкість у розробці й легко адаптуватися до нових вимог бізнесу та технологій.

Вибір відповідного архітектурного підходу залежить від багатьох факторів: масштабу проєкту, прогнозованого навантаження, очікуваного часу до виходу на ринок і доступних технологічних ресурсів. Саме тому архітектура інформаційної системи є основою її ефективності, стабільності та здатності відповідати на виклики майбутнього.

Сучасні тенденції у сфері інформаційних систем сприяють появі нових архітектурних підходів, які спрямовані на підвищення гнучкості, безпеки та швидкості впровадження інновацій. Одним із таких підходів є архітектура, побудована навколо контейнеризації та оркестрації.

Контейнерна архітектура та оркестрація:

Контейнеризація стала основою багатьох сучасних інформаційних систем, оскільки дозволяє упаковувати додатки та всі їхні залежності у стандартизовані контейнери. Це значно спрощує розгортання системи на будь-якому середовищі, полегшує масштабування, забезпечує стабільність та незалежність від платформи. Системи оркестрації контейнерів, такі як Kubernetes, додають ще один рівень ефективності, дозволяючи автоматично розподіляти ресурси, запускати резервні копії, оновлювати компоненти без простоїв і зберігати загальну цілісність системи.

Мультимодельна архітектура:

Цей підхід передбачає поєднання кількох моделей зберігання та обробки даних в межах однієї системи. Наприклад, одна й та сама інформаційна система може використовувати реляційні бази даних для транзакційних операцій, графові бази даних для аналізу зв’язків між об’єктами, а також хмарні сховища для масштабованого зберігання великих обсягів неструктурованих даних. Завдяки цьому мультимодельна архітектура підвищує продуктивність та ефективність роботи з даними, дозволяючи обирати найкращий інструмент для кожного конкретного завдання.

Архітектура з низькою затримкою:

В умовах зростаючої потреби в реальному часі до обробки даних, з'явилися підходи, що дозволяють забезпечувати наднизьку затримку під час передачі та обробки інформації. Це особливо важливо для критично важливих систем, таких як фінансові торгові платформи, системи інтернету речей (IoT), автономні транспортні засоби та стримінгові сервіси. Архітектури з низькою затримкою зазвичай орієнтовані на розподіл обчислювальних потужностей ближче до джерел даних (наприклад, використання edge-комп’ютингу), оптимізацію мережевих шляхів та застосування високопродуктивних обчислювальних платформ.

Архітектура, орієнтована на штучний інтелект:

З огляду на зростання обсягів даних і важливість аналітики, архітектура інформаційних систем дедалі частіше інтегрується з технологіями штучного

інтелекту та машинного навчання. Вона включає компоненти для автоматичного збирання даних, їхньої попередньої обробки, тренування моделей, розгортання алгоритмів у виробничих середовищах та подальшого моніторингу продуктивності цих моделей. Такий підхід дозволяє інформаційним системам не лише виконувати запрограмовані завдання, а й адаптуватися до змін і самостійно вдосконалювати свої процеси.

Загалом, архітектурні підходи до проєктування інформаційних систем продовжують розвиватися, щоб відповідати на все більш складні виклики сучасного світу. Завдяки цьому сучасні інформаційні системи стають більш гнучкими, надійними, ефективними та здатними підтримувати інноваційні бізнес-моделі.

Архітектури для обчислень на периферії (Edge Computing):

З розвитком інтернету речей (IoT) та необхідністю обробляти дані ближче до джерела їхнього виникнення, дедалі більшого значення набувають edge-архітектури. Вони дозволяють здійснювати попередню обробку даних безпосередньо на пристроях, які їх генерують, або на близьких до них серверах. Це знижує навантаження на основні сервери та центральні хмарні системи, забезпечує мінімальні затримки у передачі даних, а також підвищує загальну продуктивність та ефективність системи. Завдяки цьому підходу організації можуть швидше реагувати на зміни умов, поліпшувати якість обслуговування користувачів і зменшувати витрати на обробку даних у віддалених центрах.

Гібридні та мультихмарні архітектури:

У сучасних умовах компанії рідко обмежуються використанням лише однієї хмарної платформи. Зростає популярність мультихмарних архітектур, які дозволяють розподіляти навантаження між кількома провайдерами хмарних послуг. Гібридні архітектури, своєю чергою, поєднують локальні дата-центри з хмарними рішеннями, забезпечуючи оптимальне співвідношення продуктивності, вартості та безпеки. Це дозволяє організаціям зберігати контроль над критично важливими даними, одночасно використовуючи переваги масштабованості та гнучкості хмари.

Архітектури для розподілених реєстрів (DLT) і блокчейн:

Технологія блокчейн та інші розподілені реєстри спричинили появу нових архітектур, орієнтованих на забезпечення прозорості, надійності й незмінності даних. Такі архітектури знаходять застосування у фінансах, логістиці, охороні здоров'я та багатьох інших галузях. Вони дають змогу будувати системи, де всі учасники мають однаковий доступ до даних, а транзакції є максимально захищеними й неможливими для фальсифікації.

Зростання автоматизації та DevOps-архітектур:

Сучасні архітектури інформаційних систем усе частіше інтегруються із процесами безперервної інтеграції та безперервного розгортання (CI/CD), що сприяє швидшому впровадженню оновлень і підвищенню загальної стабільності. Використання інфраструктури як коду (IaC), інструментів автоматичного моніторингу, тестування та розгортання дозволяє компаніям реагувати на зміни в реальному часі, мінімізувати простої і забезпечувати безперервну оптимізацію своїх систем.

Розвиток архітектур для підтримки високих навантажень і глобальних користувачів:

З огляду на зростаючі потреби в обробці великих обсягів даних і підтримці мільйонів одночасних користувачів, з'являються архітектури, орієнтовані на високий рівень доступності, відмовостійкість та продуктивність. Використання технологій розподіленого кешування, балансувальників навантаження, глобально розподілених баз даних і мереж доставки контенту (CDN) допомагає створювати системи, що ефективно працюють у глобальному масштабі, забезпечуючи користувачів максимально швидким доступом до інформації.

Таким чином, архітектури інформаційних систем продовжують еволюціонувати, відповідаючи на вимоги часу. Інноваційні підходи, інтеграція новітніх технологій і використання гнучких моделей розробки дозволяють створювати потужні, масштабовані та стабільні системи, які підтримують сучасний бізнес та наукові дослідження.

4. Переваги та недоліки різних архітектурних підходів

Залежно від сфери застосування інформаційні системи можуть мати різні архітектурні підходи. Наприклад, корпоративні системи часто використовують багаторівневі архітектури для гнучкості та масштабованості, а у сфері мобільних технологій поширені хмарні рішення.

Процес проектування інформаційної системи тісним образом пов’язаний з її архітектурним описом, що відбито в деяких визначеннях терміну "архітектура". Можна виділити п’ять різних підходів до проектування:

- Календарний підхід.
- Підхід, за основу якого взятий процес керування вимогами.
- Підхід, заснований на процесі розробки документації.
- Підхід, в основі якого лежить система керування якістю. Архітектурний підхід.

Архітектурний підхід до проектування інформаційних систем можна вважати найбільш зрілим. Його ключовим аспектом є створення фреймворка, тобто каркаса, адаптація якого під потреби конкретної системи буде легко здійсненна. Відповідно до цього, завдання проектування розбивається на дві: розробки багаторазово використовуваного каркаса й створення системи на його основі. Слід зазначити, що дані підзадачі можуть вирішуватися різними групами фахівців. При використанні каркасів з’являється можливість досить швидко змінювати функціональність системи за рахунок ітеративності процесу проектування. Архітектурний підхід покликаний ліквідувати недоліки, що виникають у процесі проектування, заснованому на керуванні вимогами

Основні підходи до проєктування архітектури інформаційних систем

Проєктування архітектури інформаційних систем є одним із ключових етапів їхньої розробки, оскільки визначає структуру, принципи взаємодії компонентів, рівні обробки даних та масштабованість системи. Вибір архітектурного підходу залежить від вимог до продуктивності, безпеки, надійності та інтеграції з іншими системами.

1. Монолітний підхід

Цей підхід передбачає побудову інформаційної системи як єдиного цілісного додатка, у якому всі компоненти тісно взаємопов'язані та функціонують у рамках одного середовища.

Характеристики:

- Всі функції системи інтегровані в один модуль.
- Дані обробляються централізовано.
- Мінімальна кількість взаємодій між різними сервісами.

Переваги:

- Простота розробки та розгортання.
- Відносно низькі вимоги до інфраструктури.
- Легкість у забезпеченні безпеки через централізоване управління.

Недоліки:

- ✓ Складність у масштабуванні.
- ✓ Обмежена гнучкість у зміні або додаванні функціональності.
- ✓ Ризик збоїв через відсутність модульності.

2. Багаторівневий (клієнт-серверний) підхід

Цей підхід передбачає поділ інформаційної системи на кілька рівнів, що дозволяє розділити функції та покращити продуктивність. Найпоширенішими є **двохрівневі та трирівневі архітектури**.

Характеристики:

- Дані, бізнес-логіка та інтерфейс користувача розподілені на окремі рівні.
- Сервери забезпечують централізоване управління даними та запитами.
- Користувач взаємодіє із системою через клієнтське програмне забезпечення або веб-інтерфейс.

Переваги:

- Поліпшена продуктивність через розподіл навантаження.
- Гнучкість у зміні або розширенні функціональності.

- Можливість використання віддалених серверів для збереження та обробки даних.

Недоліки:

- ✓ Вищі вимоги до інфраструктури та адміністрування.
- ✓ Необхідність налаштування ефективних механізмів безпеки.
- ✓ Потенційна залежність від серверів (збій сервера може призвести до недоступності системи).

3. Мікросервісний підхід

Цей підхід базується на принципі розподілу системи на незалежні сервіси, кожен з яких виконує окрему функцію і взаємодіє з іншими сервісами через API.

Характеристики:

- Система складається з невеликих незалежних сервісів, які можуть працювати автономно.
- Використовується API для взаємодії між сервісами.
- Кожен мікросервіс може мати власну базу даних та інфраструктуру.

Переваги:

- Висока масштабованість – можна збільшувати потужності окремих сервісів.
- Гнучкість у розробці – різні команди можуть працювати над різними мікросервісами.
- Полегшене тестування та оновлення окремих компонентів.

Недоліки:

- ✓ Вища складність управління мережею взаємодій між сервісами.
- ✓ Вимоги до високої продуктивності API-запитів.
- ✓ Необхідність розвинених механізмів безпеки та контролю доступу між сервісами.

4. Хмарний підхід

Цей підхід передбачає використання хмарних платформ (AWS, Google Cloud, Microsoft Azure) для зберігання та обробки даних, що дозволяє зменшити потребу у локальній інфраструктурі.

Характеристики:

- Дані та обчислення виконуються у хмарному середовищі.
- Використовуються гнучкі моделі інфраструктури (IaaS, PaaS, SaaS).
- Масштабування ресурсів відбувається автоматично.

Переваги:

- Зниження витрат на власну інфраструктуру.
- Гнучкість і швидке розширення потужностей.
- Висока доступність даних та безпека резервного копіювання.

Недоліки:

- ✓ Залежність від постачальників хмарних послуг.
- ✓ Питання конфіденційності та безпеки даних у хмарному середовищі.
- ✓ Можливість додаткових витрат через зміну тарифних планів.

5. Орієнтований на дані підхід (Data-Centric Architecture)

Цей підхід базується на тому, що дані є основним активом системи, а всі бізнес-процеси навколо них організовані.

Характеристики:

- Основний фокус на моделюванні, аналізі та збереженні даних.
- Використання централізованих сховищ даних (Data Warehouse, Data Lake).
- Гнучкість у взаємодії з іншими системами та платформами.

Переваги:

- Поліпшений контроль за якістю та безпекою даних.
- Легкість інтеграції з іншими системами та аналітичними платформами.
- Покращені можливості аналітики та прогнозування.

Недоліки:

- ✓ Висока складність управління великими обсягами даних.
- ✓ Вимоги до продуктивних серверів та сховищ.
- ✓ Можливі ризики втрати даних при неправильному адмініструванні [4].

Вибір підходу до проєктування архітектури інформаційних систем залежить від потреб бізнесу, вимог до продуктивності, безпеки, масштабованості та гнучкості системи:

Монолітна архітектура підходить для невеликих проєктів із мінімальними вимогами до змін.

Багаторівневий підхід оптимальний для корпоративних систем із високим навантаженням.

Мікросервісна архітектура забезпечує модульність та гнучкість, але потребує складнішого адміністрування.

Хмарний підхід дозволяє знизити витрати на інфраструктуру та підвищити масштабованість.

Орієнтований на дані підхід використовується у проєктах, що пов'язані з Big Data та складною аналітикою.

Знання основних архітектурних підходів допомагає інженерам та аналітикам обирати оптимальні рішення для конкретних завдань, забезпечуючи надійну, ефективну та масштабовану роботу інформаційних систем.

Контрольні питання

1. Що розуміється під архітектурою інформаційної системи?
2. Які існують критерії класифікації архітектур інформаційних систем?
3. Які функції виконують бази даних у архітектурі інформаційних систем?
4. Що таке трирівнева архітектура, і які переваги вона надає?
5. Які переваги забезпечує хмарна архітектура?
6. Які недоліки мають монолітні інформаційні системи?
7. Чому вибір відповідного архітектурного підходу є критично важливим для успішності проєкту?

Тема 4.
МЕТОДИ ТА МОДЕЛІ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Метою лекції є ознайомлення здобувачів із сучасними методами та моделями створення інформаційних систем, розгляд принципів їх розробки, впровадження та оптимізації. Лекція спрямована на формування розуміння основних етапів життєвого циклу інформаційних систем, методологій проєктування, а також критеріїв вибору відповідної моделі розробки залежно від специфіки проєкту.

План лекції

1. Загальні поняття про процес створення інформаційних систем.
2. Моделі життєвого циклу розробки інформаційних систем.
3. Методи створення інформаційних систем.

1. Загальні поняття про процес створення інформаційних систем

Розробка ***інформаційної системи*** включає всі етапи, необхідні для її створення та впровадження, починаючи з аналізу вимог і закінчуючи підтримкою після введення в експлуатацію. Це комплексний, багатогранний процес, який потребує детального планування, структурного підходу до проєктування та належного тестування. Ключовими аспектами є чітке розуміння потреб користувачів, узгодження технічних вимог і забезпечення того, щоб кінцевий продукт відповідав поставленим цілям. У цьому контексті, процес створення інформаційних систем можна розглядати як керовану послідовність кроків, що ведуть до створення ефективної, функціональної та стабільної системи.

Підхід до розробки системи істотно впливає на її кінцевий успіх. Правильно обраний метод і модель визначають, наскільки швидко можна реалізувати проєкт, скільки ресурсів для цього знадобиться і якою буде якість кінцевого продукту. Наприклад, у невеликих проєктах із чітко

сформульованими вимогами може бути достатньо традиційних методів, тоді як масштабні проєкти із високим ступенем невизначеності вимагають більш гнучких підходів. Кожен метод має свої сильні сторони й обмеження, і вибір повинен залежати від специфіки проєкту, доступних ресурсів, рівня ризику, вимог до швидкості розробки та подальшого супроводу.

Крім того, вибір методу та моделі розробки має враховувати не лише поточні потреби, але й перспективи подальшого розвитку системи. Наприклад, якщо проєкт розрахований на довготривале використання і передбачає масштабування чи інтеграцію з іншими системами, доцільно обрати модель, яка дозволяє гнучко адаптуватися до змін, легко оновлювати компоненти і зберігати стабільність роботи.

Важливим аспектом є також врахування рівня кваліфікації команди розробників і технічної бази. Деякі методи вимагають значного досвіду у використанні складних інструментів чи фреймворків, тоді як інші можуть бути реалізовані навіть у невеликих командах з мінімальними ресурсами. Відповідний підхід сприяє підвищенню продуктивності розробників, зменшенню кількості помилок і забезпеченню високої якості продукту вже на етапі тестування.

У результаті, вибір методу та моделі розробки – це не лише технічне рішення, але й стратегічний крок, який визначає, наскільки ефективно команда зможе працювати, наскільки задоволені будуть кінцеві користувачі та як швидко система окупить витрати на її створення. Правильний вибір у цьому питанні – запорука успішного завершення проєкту, його підтримки та розвитку у майбутньому.

Успішний вибір методології розробки також передбачає врахування культурних і організаційних факторів. Наприклад, якщо в компанії вже існує певна традиція використання конкретних методів або інструментів, нові підходи можуть потребувати додаткового навчання, змін у внутрішніх процесах і перебудови робочого циклу. Це може суттєво вплинути на час і витрати

впровадження, навіть якщо обраний методологічний підхід є теоретично найкращим.

Крім того, варто враховувати зворотний зв'язок від майбутніх користувачів системи. Під час планування розробки важливо розуміти, наскільки швидко користувачі зможуть адаптуватися до нової системи, які функції є для них критичними, і як гнучкість обраного підходу допоможе враховувати ці потреби вже в процесі створення продукту. Чим краще методологія підтримує поступове вдосконалення ітераціями, тим більше можливостей у команди швидко реагувати на змінені вимоги чи нові бізнес-умови.

На додаток до цього, вибір відповідної методології має прямий вплив на те, як система буде підтримуватися і розвиватися після завершення початкового етапу розробки. Вибір гнучкого або модульного підходу може значно полегшити інтеграцію нових функцій, перехід на сучасніші технології чи підключення до зовнішніх платформ. Отже, правильний методологічний вибір зменшує майбутні витрати на технічну підтримку та розширення системи, що є стратегічно вигідним для будь-якої організації.

Не менш важливим є врахування галузевих стандартів і рекомендацій при виборі методології. У певних індустріях існують суворі регуляторні вимоги або стандарти якості, які можуть впливати на вибір підходів до розробки. Наприклад, у сфері фінансових технологій чи охорони здоров'я необхідно забезпечити високий рівень безпеки, аудиторський контроль, точність і відповідність регуляторним нормам. У таких випадках вибір методології, яка підтримує детальну документацію, суворі процедури тестування та простежуваність змін, стає критично важливим.

Водночас, якщо проєкт передбачає співпрацю з міжнародними командами або підрядниками, потрібно врахувати рівень координації та комунікації, який забезпечує методологія. Гнучкі моделі, такі **як Scrum або SAFe (Scaled Agile Framework),** пропонують чіткі механізми планування, прозорість прогресу та регулярний зворотний зв'язок, що особливо корисно у розподілених командах.

Ці підходи дозволяють мінімізувати ризики, пов’язані з різницею в часових поясах, культурними особливостями або мовними бар’єрами, а також забезпечують швидке виявлення та усунення проблем.

Окрім того, важливо звернути увагу на можливість масштабування обраного підходу. Якщо система розрахована на довготривалу експлуатацію, доцільно передбачити, як вона буде адаптуватися до зростання кількості користувачів, обсягу даних і вимог до продуктивності. Методології, які забезпечують поетапне впровадження змін і дозволяють легко інтегрувати нові технології або сервіси, допоможуть підтримувати високу продуктивність і конкурентоспроможність системи на довгі роки.

Можна визначити такі три покоління процесів розробки ПЗ:

1. Традиційний процес: 1960–1970-і рр., кустарне виробництво. Організації використовують кустарний інструментарій, куртарні процеси і практично всі компоненти для замовника пишуться на примітивних мовах. Результат виконання проекту був легко передбачуваний в тому сенсі, що він практично ніколи не вкладався в заздалегідь задані вартість, терміни та якість.

2. Перехідний: 1980–1990–і рр., програмна інженерія. Організації використовують відтворювані процеси і готові інструменти, а більшість створюваних компонентів (>70%) пишеться на мовах високого рівня. Деякі компоненти (30%). стають доступними в якості комерційного продукту, включаючи ОС, системи управління базами даних (СУБД), мережеве ПЗ і графічний інтерфейс користувача. У 80–і рр. XX ст. деякі організації починають досягати економії при великих масштабах, однак зі зростанням складності додатків (особливо при переході до розподілених систем) існуючі мови, методи і технології виявилися недостатніми, щоб підтримувати необхідний рівень промислового створення.

3. Сучасна практика: починаючи з 2000 р., виробництво ПЗ. Цей період характеризує застосування керованих і вимірюваних процесів, інтегрованих середовищ автоматизації і здебільшого (на 70%) готових компонентів 30 (можливо, лише 30% компонентів слід створювати на замовлення).

Використовуючи переваги технології створення ПЗ та інтегрованих середовищ розробки, можна швидко створювати системи, побудовані з компонентів.

Технології, які дозволяють автоматизувати середовище розробки, зменшити розмір ПЗ і вдосконалити процес, не є незалежними. Для кожного нового періоду часу ключовим стає деяке вдосконалення всіх технологій: наприклад, переваги нового процесу не можуть бути успішно використаними без 1) нових технологій створення компонентів, 2) підвищення ступеня автоматизації. Організації досягають більшої економії при великих масштабах протягом технологічно успішних періодів – в рамках великих проектів (системи систем), продуктів довготривалого використання і продуктових ліній, які включають в себе множину однотипних проектів.

Проектування ІС охоплює такі три основні області:

1) проектування об'єктів даних, які будуть реалізовані в базі даних;

2) проектування програм, екранних форм, звітів, які будуть забезпечувати виконання запитів до даних;

3) облік конкретного середовища або технології, а саме: топології мережі, конфігурації апаратних засобів, використовуваної архітектури (файл–сервер або клієнт–сервер), паралельної та розподіленої обробки даних тощо [17].

Відповідно до сучасної методології, процес створення ІС є процесом побудови і послідовного перетворення ряду узгоджених моделей на всіх етапах життєвого циклу (ЖЦ) ІС. На кожному етапі ЖЦ створюються специфічні для нього моделі (організації, вимог до ІС, проекту ІС, вимог до застосунків тощо). Моделі формуються робочими групами команди проекту, зберігаються і накопичуються в репозитарії проекту. Створення моделей, їх контроль, перетворення і надання в колективне користування здійснюється із використанням спеціальних програмних інструментів – CASE–засобів [1].

Отже, вибір відповідного методу та моделі розробки інформаційної системи має ключове значення для її успіху. Правильна методологія дозволяє зменшити ризики, оптимізувати витрати, скоротити терміни розробки та забезпечити високу якість кінцевого продукту. Врахування масштабу проєкту,

вимог замовника, технічних ресурсів, майбутніх змін і галузевих стандартів допомагає знайти баланс між гнучкістю, стабільністю та довговічністю системи. У результаті, продуманий підхід до розробки забезпечує не лише успішне впровадження інформаційної системи, а й її ефективну підтримку та розвиток у майбутньому.

2. Моделі життєвого циклу розробки інформаційних систем

У світі програмної інженерії ефективне управління процесом створення програмного забезпечення відіграє ключову роль у забезпеченні його якості, функціональності та відповідності вимогам замовників. Одним із основоположних концептів у цій сфері є поняття життєвого циклу програмного забезпечення (ЖЦ ПЗ), яке визначає структуру, фази і процеси, що супроводжують розробку від її початку до завершення. Розуміння цього циклу дозволяє планувати роботи, зменшувати ризики, забезпечувати прозорість і контроль на кожному етапі.

Життєвий цикл програмного забезпечення (ЖЦ ПЗ) у програмній інженерії (ПІ) являє собою певну послідовність фаз або стадій, які розпочинаються з моменту ухвалення рішення про розробку і завершуються повним вилученням програмного забезпечення з експлуатації. Кожна фаза включає виконання низки процесів, спрямованих на створення конкретних продуктів чи результатів із використанням відповідних ресурсів.

Стандарт ISO/IEC 12207:1995 "Information Technology – Software Life Cycle Processes" регламентує структуру життєвого циклу програмного забезпечення. Він описує процеси, дії та завдання, які необхідно виконати на кожному етапі. Завдяки цьому стандарту забезпечується узгодженість і прогнозованість процесів, що, своєю чергою, сприяє підвищенню якості кінцевого продукту та зменшенню витрат на його розробку та підтримку.

На практиці життєвий цикл програмного забезпечення може бути реалізований через різноманітні моделі розробки. Наприклад, класична

каскадна модель визначає послідовний перехід від однієї стадії до іншої, тоді як гнучкі методології, такі як Agile, передбачають ітераційний підхід, де окремі етапи можуть повторюватися та вдосконалюватися на основі постійного зворотного зв’язку.

Ключовим аспектом життєвого циклу є можливість контролю та управління на кожному його етапі. Завдяки чітко визначеним фазам і процесам команди розробників можуть краще оцінювати прогрес, своєчасно виявляти проблеми, коригувати вимоги та вдосконалювати підходи до тестування і впровадження.

Таким чином, концепція життєвого циклу програмного забезпечення є фундаментом для успішного управління проєктами у сфері програмної інженерії. Її розуміння дозволяє не лише знизити ризики, пов’язані з невідповідністю вимогам або якістю продукту, а й забезпечити прозору і передбачувану розробку, яка відповідає сучасним стандартам та потребам ринку.

Згідно зі стандартом, програмне забезпечення трактується як сукупність комп’ютерних програм, процедур, а також супровідної документації та даних. Процес у цьому контексті визначається як набір взаємопов’язаних дій, спрямованих на перетворення вхідних даних на вихідні результати. Такий процес структуровано на певні дії, які, у свою чергу, розбиваються на конкретні завдання. Виконання кожного елемента відбувається тоді, коли це необхідно, без чітко встановленої послідовності кроків, а ініціація нових процесів або завдань може здійснюватися іншими процесами в системі.

Усі продукти програмної інженерії являють собою різні форми описів, включаючи тексти вимог, угоди між сторонами, технічну документацію, програмний код і супровідні інструкції з експлуатації. Основними ресурсами, які визначають ефективність і успішність розробки програмного забезпечення, є час та вартість. Вони напряму впливають на якість кінцевого продукту, дотримання термінів і дотримання запланованого бюджету.

Стандарт ISO/IEC 12207 чітко розподіляє всі процеси життєвого циклу програмного забезпечення на три основні групи (рис. 4.1), що дозволяє структурувати розробку, полегшувати управління і сприяти досягненню передбачуваних результатів.

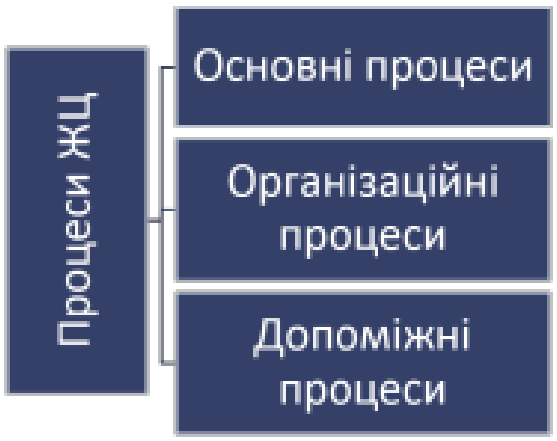

Рис. 4.1. Процеси життєвого циклу розроблення ПЗ

Джерело: побудовано на основі [23]

Основні процеси включають:

- **Процес придбання:** ініціює життєвий цикл інформаційної системи (ІС) і визначає замовника. Він охоплює виконання замовлення та передачу продукту кінцевому споживачеві.
- **Процес розроблення:** визначає дії, які здійснює організація-розробник для створення інформаційного продукту. До них належить розробка компонентів програмного забезпечення (ПЗ) відповідно до вимог, підготовка проектної та експлуатаційної документації, формування матеріалів для перевірки якості та працездатності програмних продуктів, а також забезпечення навчальних ресурсів для персоналу.
- **Процес постачання:** регламентує дії, пов’язані з передачею розробленого продукту замовнику, включаючи його приймання та офіційну передачу.

• **Процес експлуатації:** охоплює обслуговування системи під час її використання, зокрема надання консультацій користувачам, вивчення їхніх побажань і пропозицій для покращення системи.

• **Процес супроводження:** передбачає управління змінами, забезпечення актуальності та функціональності системи, установлення нових версій та вилучення застарілих версій у користувача.

Процес розробки програмного забезпечення охоплює весь шлях від виявлення потреб замовника до передачі йому готового продукту (рис. 4.2).

Рис. 4.2. Процеси розроблення програмного забезпечення

Джерело: побудовано на основі [23]

Основні етапи цього процесу:

Аналіз і формування вимог: визначення та збір вимог замовника, їх аналіз і документування у зрозумілій формі, яка буде однаково прозора для обох сторін — замовника і виконавця.

Проєктування: перетворення сформульованих вимог у проєктні рішення, які включають розробку загальної архітектури системи, прив’язку архітектури до середовища її функціонування, деталізацію компонентів і модулів, з яких складається програмний продукт.

Реалізація: написання коду та створення програмної системи на основі прийнятих проєктних рішень.

Тестування: перевірка функціонування кожного окремого модуля, оцінка інтеграції компонентів, а також перевірка системи в цілому на відповідність поставленим вимогам. Цей етап охоплює:

Верифікацію — перевірку, чи реалізація відповідає проєктним специфікаціям.

Валідацію — підтвердження, що готовий продукт виконує поставлені функції та відповідає очікуванням замовника.

Підготовчий етап розпочинається з вибору моделі життєвого циклу програмного забезпечення (ЖЦ ПЗ), яка відповідатиме масштабам, складності та значущості проекту. Обраний підхід до розробки має бути адаптований до конкретних умов проекту. Розробник повинен визначити, які стандарти, методи і інструменти будуть використовуватися, узгодити їх із замовником і скласти детальний план виконання робіт.

Аналіз вимог до системи включає вивчення функціональних можливостей, потреб користувачів, а також вимог до надійності, безпеки та зовнішніх інтерфейсів. Усі ці вимоги мають бути оцінені за критеріями можливості їхньої реалізації та перевірки на етапі тестування.

Проектування архітектури системи спрямоване на визначення її основних компонентів: апаратного забезпечення, програмних модулів і операцій, які виконуватиме персонал. Аналіз вимог до програмного забезпечення зосереджений на таких аспектах, як:

- функціональні можливості, включаючи продуктивність і середовище функціонування;
- зовнішні інтерфейси;
- вимоги до надійності та безпеки;
- ергономічні аспекти;
- вимоги до даних, інсталяції та введення системи в експлуатацію;
- вимоги до документації користувачів, експлуатації та супроводу.

Проектування архітектури програмного забезпечення (ПЗ) охоплює такі ключові завдання для кожного його компонента:

- перетворення вимог до ПЗ на архітектурне рішення, яке визначає загальну структуру та склад компонентів;
- розробка та документування інтерфейсів для взаємодії програмних компонентів і баз даних (БД);
- створення попереднього варіанту документації для користувачів;
- підготовка та документування початкових вимог до тестування й плану інтеграції ПЗ.

На етапі детального проектування ПЗ здійснюються такі дії:

- визначення та опис компонентів ПЗ, а також інтерфейсів між ними на рівні деталізації, достатньому для їхньої самостійної реалізації та тестування;
- розробка та документування детальної схеми бази даних;
- уточнення й оновлення супровідної документації;
- складання детальних вимог до тестів і плану тестування компонентів ПЗ;
- оновлення плану інтеграції ПЗ за потреби.

Кодування та тестування ПЗ включають наступні завдання:

- створення (написання) і документування кожного окремого компонента ПЗ і бази даних, а також тестових процедур і даних для їх перевірки;
- тестування всіх компонентів ПЗ і БД для перевірки відповідності встановленим вимогам;
- документування результатів тестування;
- уточнення та оновлення користувацької документації та плану інтеграції ПЗ, якщо це необхідно.

Інтеграція програмного забезпечення (ПЗ) включає об'єднання розроблених компонентів відповідно до плану інтеграції та їх подальше тестування. Для кожного компонента створюються спеціальні тестові набори й процедури, які дозволяють перевірити виконання кваліфікаційних вимог під час майбутнього кваліфікаційного тестування. Кваліфікаційна вимога визначається

як сукупність критеріїв або умов, які мають бути виконані, щоб підтвердити відповідність програмного продукту його специфікаціям і можливість його використання в умовах експлуатації.

Кваліфікаційне тестування програмного забезпечення проводиться розробником у присутності замовника з метою підтвердження відповідності програмного продукту заявленим специфікаціям. На цьому етапі кожен компонент ПЗ перевіряється щодо всіх вимог із використанням різноманітних тестів. Окрім функціональності компонентів, також оцінюються повнота та якість технічної документації, її відповідність самій програмній системі.

Інтеграція системи, своєю чергою, передбачає поєднання всіх її складових частин, включно із програмним забезпеченням та обладнанням. Після цього проводиться кваліфікаційне тестування вже всієї системи для підтвердження відповідності всім вимогам. У цей же час готується й перевіряється повний комплект документації для всієї системи.

Встановлення програмного забезпечення виконується розробником відповідно до плану розгортання в передбаченому операційному середовищі та на обладнанні, визначеному в замовленні. Приймання ПЗ передбачає аналіз результатів кваліфікаційного тестування як програмного продукту, так і всієї системи в цілому. Цей аналіз здійснює замовник за підтримки розробника, з документуванням усіх отриманих результатів. Остаточна передача програмного забезпечення замовнику проводиться розробником згідно з умовами договору, включаючи надання необхідного навчання користувачів і підтримки.

На різних етапах розвитку галузі та у різних джерелах представлено різні списки моделей розробки програмного забезпечення, причому їх інтерпретація також змінювалася з часом. Наприклад, раніше інкрементальна модель розглядалася як процес створення системи через послідовність збірок (релізів), виконуваних згідно з попередньо підготовленими планами і незмінними, чітко визначеними вимогами. Сучасне розуміння інкрементального підходу часто пов’язане з поетапним додаванням нової функціональності до продукту, що створюється.

На перший погляд може скластися враження, що індустрія нарешті дійшла до єдиної «правильної» моделі. Проте навіть каскадна модель, яку теорія і практика багато разів оголосили застарілою, все ще використовується на практиці. Спіральна модель є прикладом еволюційного підходу до розробки, яка, крім усього іншого, є однією з небагатьох моделей, що акцентують увагу на аналізі ризиків та їхньому попередженні.

Розглянемо детальніше три моделі: інкрементальну, каскадну та спіральну, щоб охарактеризувати їхні основні особливості та сфери застосування.

Каскадна (водоспадна) модель

Ця модель (рис. 4.3) передбачає чітко визначену послідовність етапів, які виконуються лише один раз у встановленому порядку. Всі фази проєкту проходять одна за одною з детальним попереднім плануванням і фокусом на жорстко заданих або заздалегідь повністю сформульованих вимогах до програмного продукту.

Рис. 4.3. Каскадна модель життєвого циклу

Джерело: побудовано на основі [23]

На практиці ця модель виявила свою недосконалість у більшості IT-проєктів. Винятком є, хіба що, певні проєкти, пов’язані з оновленням програмного забезпечення для критично важливих систем, таких як авіаційна електроніка чи медичне обладнання. Реальний досвід свідчить, що каскадна модель не підходить для бізнес-систем, адже такі системи зазвичай

характеризуються високою динамікою змін і уточнень вимог. Неможливість чітко визначити вимоги ще до початку розробки (особливо для нових систем), а також їх часті зміни під час експлуатації роблять цей підхід неефективним.

На рисунку 4.3 наведено типові етапи життєвого циклу, передбачені каскадною моделлю, а також відповідні проєктні артефакти, які на одних фазах є вихідними матеріалами, а на інших слугують вхідними даними.

Сфера застосування каскадної моделі

Каскадну модель доцільно використовувати у випадках, коли:

- вимоги до системи чітко визначені та залишаються незмінними протягом усього життєвого циклу;
- розробка базується на знайомих, добре зрозумілих методах реалізації та перевірених технічних підходах;
- проект спрямований на створення системи або продукту аналогічного тому, який команда вже розробляла раніше;
- проект полягає у розробці нової версії вже існуючого продукту або системи;
- завдання полягає в перенесенні чинного продукту на нову платформу;
- реалізується масштабний проект із залученням кількох великих команд розробників.

Однак ефективність цієї моделі обмежується її недоліками, тому вона застосовується тільки за умов стабільних вимог, добре відпрацьованих процесів і мінімальної ймовірності змін під час реалізації.

Ітеративна й інкрементальна модель – еволюційний підхід

Ітеративна модель передбачає поділ життєвого циклу проекту на серію послідовних ітерацій. Кожна ітерація являє собою своєрідний «міні-проект», що включає всі основні фази життєвого циклу, проте фокусується на створенні окремих частин функціональності (на відміну від роботи над системою в цілому). Основною метою кожної ітерації є отримання робочої версії програмної системи, яка поєднує функціонал, розроблений на попередніх етапах, із новими можливостями, доданими в поточній ітерації. В результаті

фінальної ітерації формується продукт, що містить всю необхідну функціональність. У цьому сенсі ітеративна модель забезпечує поступове вдосконалення продукту шляхом інкрементального нарощування.

З погляду структури життєвого циклу, цю модель називають ітеративною, оскільки вона базується на циклічному повторенні етапів розробки. У той же час, з точки зору розвитку продукту, її називають інкрементальною, оскільки кожна ітерація додає новий функціонал до вже створеного. На практиці ці підходи часто поєднуються, і модель зазвичай називають ітеративною, якщо йдеться про процес, або інкрементальною, якщо мають на увазі нарощування функціональності продукту.

Еволюційна модель передбачає не лише створення працездатної версії системи (з погляду тестування), але й її розгортання в реальних умовах. Відгуки користувачів аналізуються, щоб визначити напрям і зміст наступних ітерацій. На відміну від цього, у чистій інкрементальній моделі всі ітерації здійснюються за наперед визначеним планом, без розгортання проміжних версій. У результаті користувач отримує лише кінцеву версію продукту після завершення останньої ітерації.

Еволюційний підхід, побудований на ітераціях, допомагає зменшити невизначеність з кожним циклом. Це, у свою чергу, дозволяє знижувати ризики на кожному етапі розробки. Як показано на рисунку 4.4, ітеративний підхід може застосовуватися не тільки до всього життєвого циклу в цілому (з перекривними фазами — аналізом вимог, проектуванням, розробкою тощо), але й до окремих фаз. Наприклад, кожна фаза може розбиватися на уточнюючі ітерації, пов’язані з деталізацією архітектури модулів системи або іншими аспектами декомпозиції проекту.

Завдяки ітераційному підходу команди можуть поступово поглиблювати своє розуміння продукту, що створюється. На початкових етапах фокус робиться на базових вимогах, які визначають загальний напрямок роботи. У наступних ітераціях відбувається деталізація, уточнення специфікацій та адаптація до нових умов або змін у потребах замовника.

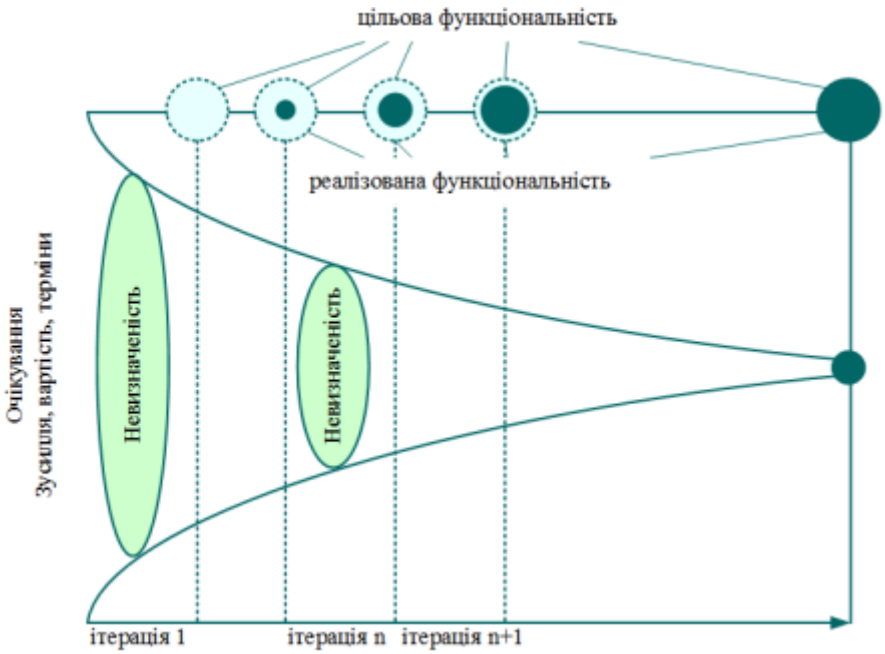

Рис. 4.4. Зниження невизначеності та інкрементальне розширення функціональності при ітеративній організації життєвого циклу

Джерело: побудовано на основі [23]

Ітераційний процес також полегшує інтеграцію нових технологій або методів у ході розробки. Це дозволяє команді експериментувати з різними рішеннями, оцінювати їхню ефективність і швидко впроваджувати найкращі практики. Такий підхід сприяє постійному вдосконаленню процесів і результатів, підвищує якість кінцевого продукту та забезпечує більш передбачувані результати.

Зниження невизначеності й розбиття проекту на дрібніші частини дозволяють досягти вищої продуктивності. Команди можуть сфокусуватися на окремих, чітко визначених завданнях, що підвищує точність прогнозів і термінів виконання, а також знижує ймовірність виникнення критичних помилок на пізніх етапах.

Спіральна модель

Спіральна модель є одним із найбільш популярних і поширених варіантів еволюційного підходу, який з часом набув статусу окремої методології з різними варіантами деталізації та сценаріями розвитку. Цей підхід, уперше

представлений Баррі Боемом у 1988 році, відрізняється особливою увагою до управління ризиками, що суттєво впливає на структуру та організацію життєвого циклу.

Однією з основних переваг спіральної моделі є її здатність інтегрувати найкращі аспекти існуючих методологій розробки програмного забезпечення. Вона добре підходить для створення інноваційних або складних систем, у випадках, коли замовник і розробник не мають чіткої картини кінцевого продукту чи впевненості в успішній реалізації проекту через високий рівень невизначеності та ризиків.

Замість створення цілісної системи відразу, ця модель передбачає поступову розробку окремих її частин. У процесі розробки можливе уточнення вимог, адаптація до нових умов або навіть припинення проекту, якщо ризики виявляються надто високими. Згідно з графічним представленням (рис. 4.5), проект може завершитися не лише після впровадження, але й на етапі аналізу ризиків, якщо подальший розвиток виявиться недоцільним.

Ця гнучкість спіральної моделі дозволяє знижувати ризики і поступово формувати більш чітке розуміння кінцевого продукту. На кожному витку «спіралі» команда отримує можливість переглянути початкові рішення, оцінити досягнуті результати, виявити потенційні проблеми та прийняти зважені рішення щодо подальшого розвитку. Завдяки цьому спіральна модель особливо ефективна для великих та складних проектів, де ризики, пов’язані з неточністю вимог або технологічними обмеженнями, можуть серйозно вплинути на успіх проекту.

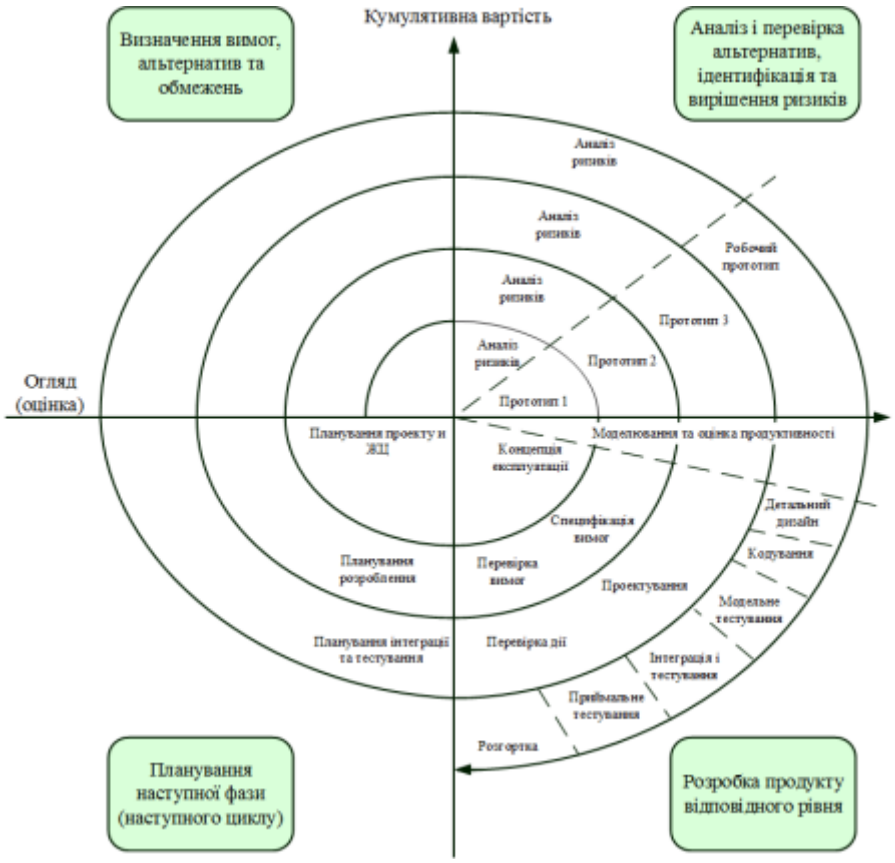

Рис. 4.5. Оригінальна спіральна модель ЖЦ, розроблена Боемом

Джерело: побудовано на основі [23]

Водночас використання спіральної моделі вимагає від команди високої кваліфікації та досвіду. Успіх проекту залежить від уміння точно ідентифікувати ризики, належно планувати ітерації, забезпечувати якісну комунікацію між розробниками та замовником, а також ретельно документувати всі зміни і результати. При правильному застосуванні спіральна модель дозволяє забезпечити ефективне управління проектом навіть у складних і динамічних умовах.

3. Методи створення інформаційних систем

Розробка інформаційних систем характеризується складністю початкових етапів життєвого циклу, тоді як подальші стадії є менш трудомісткими. Проте

питання, що залишилися невирішеними на початкових етапах, можуть створити серйозні труднощі у майбутньому, ускладнюючи процес завершення проекту.

Відповідно до методів виконання аналізу та проектування, прийнято виділяти кілька методів до створення інформаційних систем:

- структурно-орієнтовані;
- об’єктно-орієнтовані;
- процесно-орієнтовані.

Структурно-орієнтовані методи дозволяють зменшити складність великих систем за рахунок розбиття їх на окремі компоненти, які можна розглядати як "чорні скриньки" та організовувати у вигляді ієрархічної структури. Перевага такого підходу полягає в тому, що користувачі "чорних скриньок" не потребують знання їх внутрішньої реалізації, їм достатньо розуміти призначення компонентів, а також їхні входи і виходи.

Об’єктно-орієнтовані методології ґрунтуються на об’єктній декомпозиції предметної області. Система розглядається як сукупність об’єктів, що взаємодіють між собою шляхом передачі інформації. Цей підхід не виключає структурного, оскільки фрагменти методологій структурного аналізу часто використовуються в об’єктно-орієнтованому аналізі для моделювання як структури, так і поведінки об’єктів.

Процесно-орієнтований метод базується на концепції реінжинірингу бізнес-процесів. Він спрямований на оптимізацію бізнес-процесів і побудову інформаційних систем, які підтримують ефективну взаємодію між різними елементами організації.

Застосування кожного з цих методів залежить від особливостей проекту та його вимог. Наприклад, структурно-орієнтовані методи добре підходять для великих систем із чітко визначеною ієрархією компонентів. Об’єктно-орієнтовані підходи, зі свого боку, стають у нагоді, коли важливо моделювати систему як сукупність взаємодіючих об’єктів, а також забезпечувати повторне використання компонентів. Процесно-орієнтовані методології найкраще

працюють у ситуаціях, де потрібно оптимізувати бізнес-процеси або змінити існуючу організаційну структуру, щоб підвищити ефективність роботи.

Крім того, сучасні методології все частіше об’єднують елементи різних методів, щоб скористатися перевагами кожного з них. Наприклад, під час проектування складних систем може бути доцільним розпочати з об’єктно-орієнтованого аналізу, а потім, для реалізації певних підсистем або окремих компонентів, використовувати структурний підхід. У свою чергу, оптимізація процесів і організація роботи команди можуть базуватися на принципах реінжинірингу, що є характерним для процесно-орієнтованого підходу.

Таким чином, успіх проекту багато в чому залежить від гнучкості обраного підходу, здатності адаптуватися до змін і забезпечити ефективну взаємодію всіх учасників розробки. Інтеграція різних методологій дає змогу створювати високоякісні інформаційні системи, які відповідають потребам замовника, швидко адаптуються до змін і легко підтримуються протягом усього життєвого циклу.

Розглянемо кілька прикладів, як комбіноване використання різних методів може впливати на успішність проектів.

1. **Проект із впровадження ERP-системи**

Для створення та впровадження комплексної ERP-системи, яка охоплює всі основні бізнес-процеси компанії, розробники часто починають із структурно-орієнтованого підходу. Це дозволяє створити загальну модель модулів (наприклад, фінанси, управління ланцюгами постачання, виробництво, HR), визначити їхню ієрархію та функціональні залежності. Потім, під час розробки кожного з модулів, може бути застосований об’єктно-орієнтований підхід, що забезпечує чітке моделювання класів (наприклад, працівників, замовлень, матеріалів) і їхньої взаємодії. У свою чергу, під час впровадження системи на підприємстві, залучення процесно-орієнтованих методів дозволяє провести реінжиніринг бізнес-процесів, скоригувати їх під нову систему і забезпечити ефективну інтеграцію з організаційною структурою.

2. **Розробка мобільного додатку для онлайн-замовлення послуг**

У такому проекті розробники часто починають з об’єктно-орієнтованого аналізу, щоб змоделювати ключові сутності: користувачі, замовлення, постачальники послуг, платежі тощо. Кожна з цих сутностей може бути представлена як клас із чітко визначеними атрибутами та методами. Потім для інтеграції цих об’єктів у цілісну систему може використовуватися структурний підхід, який допомагає створити ієрархію елементів інтерфейсу, визначити основні шляхи взаємодії між об’єктами та забезпечити їхню відповідність бізнес-логіці. Якщо проект передбачає швидке зростання кількості користувачів, на етапі тестування й масштабування розробники можуть застосувати принципи реінжинірингу процесів, щоб оптимізувати основні сценарії використання і знизити навантаження на сервери.

3. **Модернізація банківської системи обробки транзакцій**

У цьому випадку початковий етап часто передбачає структурно-орієнтований аналіз для виділення критичних компонентів системи: модуль обробки платежів, модуль верифікації клієнтів, звітність тощо. Потім під час заміни чи оновлення окремих компонентів може бути використаний об’єктно-орієнтований підхід, що полегшує реалізацію нових функцій (наприклад, введення нових типів транзакцій або впровадження об’єктів для автоматичного виявлення підозрілих операцій). Згодом процесно-орієнтовані методи дозволяють оптимізувати внутрішні процедури банку, пов’язані з обробкою великих обсягів даних, забезпечуючи мінімальні затримки в роботі і високий рівень безпеки.

У всіх наведених прикладах можна побачити, як інтеграція структурного, об’єктно-орієнтованого та процесно-орієнтованого підходів допомагає адаптувати процес розробки до специфіки проекту, знижувати ризики, покращувати якість продукту і забезпечувати його ефективну інтеграцію в існуючі бізнес-процеси.

Під час розробки інформаційних систем важливо використовувати гнучкі методи, які дозволяють адаптувати процеси під конкретні завдання та умови

проекту. Інтеграція різних методологій – структурного, об’єктно-орієнтованого та процесно-орієнтованого – забезпечує баланс між організаційною чіткістю, технічною ефективністю та бізнес-оптимізацією. Такий підхід дозволяє знизити ризики, підвищити якість кінцевого продукту та забезпечити його тривалу функціональність, що є ключовим фактором успішності сучасних інформаційних систем.

Контрольні питання

1. Які ключові етапи включає процес створення інформаційної системи?
2. У чому полягає основна мета моделювання життєвого циклу інформаційної системи?
3. Як відрізняються каскадна модель і гнучкі моделі життєвого циклу?
4. Які переваги і недоліки має спіральна модель у порівнянні з іншими моделями?
5. Чим структурно-орієнтовані методи відрізняються від об’єктно-орієнтованих підходів?
6. Яку роль відіграє реінжиніринг бізнес-процесів у процесно-орієнтованих методах?
7. Які чинники впливають на вибір конкретного методу розробки інформаційної системи?
8. Як правильно інтегрувати елементи різних методів для оптимізації процесу розробки?
9. Які основні переваги забезпечує використання гнучких моделей у сучасній розробці інформаційних систем?
10. Як впливає вибір моделі та методу розробки на тривалість, вартість та якість створюваної інформаційної системи?

Тема 5.
АПАРАТНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Мета: є ознайомлення здобувачів з основними компонентами апаратного та програмного забезпечення, що забезпечують функціонування інформаційних систем. Лекція спрямована на формування розуміння ключових принципів роботи апаратної платформи, операційних систем, систем управління базами даних, мережевого обладнання та прикладного програмного забезпечення. Це дозволить здобувачам усвідомити, як різні програмно-апаратні компоненти взаємодіють для реалізації бізнес-процесів, а також підготувати їх до вибору і конфігурування необхідного обладнання та програм для створення і підтримки ефективних інформаційних систем.

План лекції

1. Апаратне забезпечення інформаційної системи.
2. Програмне забезпечення.
3. Тенденції розвитку апаратного і програмного забезпечення.

1. Апаратне забезпечення інформаційної системи

Апаратне та програмне забезпечення є ключовими складовими сучасних інформаційних систем. Саме вони забезпечують обробку, зберігання та передачу даних, виконання бізнес-логіки, підтримку операційних процесів, а також загальну стабільність і ефективність роботи IT-інфраструктури. Апаратна база створює фізичний фундамент для роботи систем, тоді як програмне забезпечення надає необхідні інструменти, сервіси та функціональні можливості.

Апаратне забезпечення:

1. **Комп'ютери, сервери, робочі станції:** Основні функції цих компонентів включають обчислювальні операції,

виконання користувацьких завдань, запуск серверних сервісів, обробку великого обсягу даних та забезпечення високої продуктивності. Від характеристик апаратного забезпечення залежить швидкість роботи системи, її масштабованість та здатність обробляти складні завдання.

2. **Системи зберігання даних:**

Жорсткі диски, твердотільні накопичувачі, мережеві системи зберігання даних (SAN, NAS) дозволяють надійно зберігати інформацію, забезпечують швидкий доступ до великих обсягів даних, підтримують резервне копіювання та архівацію, а також сприяють зменшенню часу простою під час відновлення інформації.

3. **Мережеве обладнання:**

До мережевої інфраструктури належать маршрутизатори, комутатори, точки доступу та міжмережеві екрани. Вони відповідають за стабільну та безпечну передачу даних, забезпечення безперебійної роботи локальних і глобальних мереж, а також за захист від зовнішніх загроз.

4. **Переферійні пристрої:**

Принтери, сканери, термінали та інші периферійні пристрої виконують допоміжні функції, забезпечуючи зручність роботи з документами, автоматизацію окремих операцій та інтеграцію з основною системою. Вони сприяють ефективній організації робочого процесу і зниженню часу на виконання рутинних завдань.

Вибір та використання апаратного забезпечення: Вибір апаратної платформи залежить від специфіки інформаційної системи, обсягу даних, що обробляються, і вимог до продуктивності. Наприклад, для обчислювальних центрів або великих баз даних потрібні високопродуктивні сервери з потужними багатоядерними процесорами, великим об’ємом оперативної пам’яті та високошвидкісними SSD-дисками. Для менших компаній достатньо середнього класу серверів і робочих станцій із базовими характеристиками.

Крім серверів і сховищ, у сучасних інформаційних системах значну роль відіграють мережеві пристрої. Маршрутизатори та комутатори забезпечують стабільну передачу даних між серверами, робочими станціями і зовнішніми мережами. Точки доступу Wi-Fi дозволяють розширити мережу на мобільні пристрої, а міжмережеві екрани гарантують безпеку корпоративного трафіку.

Також важливо враховувати енергоефективність і надійність обладнання. Сервери та сховища даних мають забезпечувати стабільну роботу 24/7, а периферійні пристрої, такі як принтери, сканери чи термінали, повинні відповідати конкретним потребам організації.

Перспективи розвитку апаратного забезпечення: Тенденції у сфері апаратного забезпечення включають збільшення обчислювальної потужності процесорів, зниження енергоспоживання, перехід на твердотільні накопичувачі та використання систем охолодження нового покоління. Зростає популярність віртуалізації серверів, що дозволяє використовувати одне фізичне обладнання для створення кількох віртуальних серверів, і знижує витрати на обладнання [21].

Інший важливий напрямок — інтеграція хмарних сервісів із локальними апаратними ресурсами. Це дозволяє організаціям створювати гібридні інформаційні системи, де частина даних обробляється локально, а частина — в хмарі. Такий підхід забезпечує додаткову гнучкість, масштабованість і надійність системи.

Зрештою, вибір апаратного забезпечення залежить від потреб компанії та її стратегічних планів. Інвестиції в якісне обладнання не лише підвищують ефективність роботи інформаційних систем, але й забезпечують їхню стабільність, безпеку та довгострокову продуктивність.

В сучасних інформаційних системах обов’язково використовують комп’ютерну техніку. Набір її компонентів залежить від завдань, які повинна виконувати інформаційна система.

Технічне забезпечення *– це комплекс технічних засобів, що використовуються для роботи інформаційної системи, та відповідної документації на ці засоби.*

Інформаційною системою *називають сукупність взаємозв'язаних а чне забезпечення – це комплекс технічних засобів, що використовуються для роботи інформаційної системи, та відповідної документації на ці засоби.*

Інформаційною системою *називають сукупність взаємозв'язаних апаратних і програмних засобів, які здійснюють зберігання та обробку інформації.*

Апаратні компоненти інформаційної системи називаються також hardware (тверді вироби). Основною апаратною компонентою інформаційних систем є комп'ютери.

Ще одним ключовим напрямком розвитку є розширення можливостей IoT-пристроїв та інтеграція їх із традиційними IT-інфраструктурами. Розумні датчики, контролери та інші IoT-компоненти дозволяють автоматизувати моніторинг, управління виробничими процесами та аналіз даних у реальному часі. У поєднанні із зростаючими обчислювальними потужностями серверів це сприяє створенню більш розумних і гнучких систем.

Крім того, інновації в мережевому обладнанні, такі як перехід на технології 5G і Wi-Fi 6, забезпечують значно вищі швидкості передачі даних і нижчі затримки. Це дозволяє організаціям працювати з великими обсягами даних, покращувати доступ до хмарних сервісів і підтримувати нові покоління додатків, наприклад, додатки на основі доповненої та віртуальної реальності.

Не менш важливими є технології безпеки на апаратному рівні. Сучасне обладнання дедалі частіше оснащується вбудованими засобами захисту, які забезпечують апаратне шифрування даних, захист від атак на рівні мікропрограм та зниження ризиків несанкціонованого доступу. Інвестиції у таке обладнання стають ключовим аспектом для організацій, які прагнуть дотримуватися високих стандартів інформаційної безпеки.

Таким чином, розвиток апаратного забезпечення спрямований на підвищення продуктивності, енергоефективності, безпеки та гнучкості. Ці тенденції відкривають нові можливості для бізнесу, дозволяючи організаціям швидше адаптуватися до змін і використовувати сучасні технології для підтримки конкурентних переваг.

Наприклад, у сфері високопродуктивних обчислень (HPC) нові процесори, такі як лінійки AMD EPYC або Intel Xeon Scalable, пропонують більшу кількість ядер і вищу тактову частоту, що дозволяє прискорити складні наукові моделювання або обробку великих даних. Сервери на базі цих процесорів дедалі частіше використовуються в дата-центрах, які забезпечують хмарні обчислення для корпоративних клієнтів.

Що стосується зберігання даних, поширення NVMe SSD прискорює доступ до інформації, що є критично важливим для таких застосувань, як обробка транзакцій у фінансовій сфері або аналітика в реальному часі в e-commerce. Організації, які впроваджують такі накопичувачі у свої сховища, здатні значно зменшити затримки доступу до даних, тим самим покращуючи швидкість обробки замовлень і підтримку клієнтів.

У галузі телекомунікацій впровадження мереж 5G дає можливість обробляти величезні потоки даних у режимі реального часу. Це актуально, наприклад, для смарт-міст, де тисячі IoT-датчиків моніторять стан інфраструктури, збирають екологічні показники, а потім передають дані в хмару для аналізу. Завдяки високошвидкісним мережам і сучасному серверному обладнанню ці системи можуть працювати з мінімальними затримками, оперативно виявляти проблеми і швидко реагувати на них.

Ще один приклад — медична галузь. У лікарнях зростає використання апаратних платформ, які підтримують обробку великих обсягів медичних даних і машинне навчання. Високопродуктивні сервери дозволяють запускати алгоритми для аналізу зображень, виявлення патологій на ранніх стадіях і персоналізації лікування на основі генетичної інформації.

У сфері інформаційної безпеки сучасні маршрутизатори й міжмережеві екрани обладнані апаратними модулями для шифрування трафіку в реальному часі. Наприклад, компанії, що надають фінансові послуги, впроваджують пристрої з підтримкою апаратного VPN та глибокого аналізу пакетів (DPI), щоб забезпечити захист від кібератак і гарантувати конфіденційність даних клієнтів.

Крім згаданих сфер, приклади можна навести й з виробничого середовища. У сучасній промисловості дедалі частіше використовуються промислові сервери та контролери з вбудованими механізмами штучного інтелекту для автоматизації виробничих ліній. Так, на заводах автомобільної індустрії такі сервери можуть аналізувати потоки даних від сотень датчиків у реальному часі, виявляти потенційні дефекти продукції на ранніх етапах і оперативно коригувати налаштування обладнання. Це значно скорочує час простоїв і зменшує кількість браку.

Ще один напрям — сфера розваг та медіа. З розвитком потокового відео сучасні сервери з високопродуктивними GPU дозволяють забезпечувати якісну трансляцію контенту у форматах 4K і 8K. Компанії, що працюють у галузі онлайн-стрімінгу, використовують сервери з графічними прискорювачами для стиснення відео в режимі реального часу, що дозволяє знизити обсяг даних, які передаються користувачам, і водночас зберігати високу якість картинки. Завдяки цьому глядачі отримують безперебійний доступ до контенту, навіть при високому навантаженні на мережу.

Щодо навчання й досліджень, сучасні апаратні комплекси стають основою для симуляцій і аналізу в наукових інститутах. Наприклад, суперкомп’ютери, оснащені тисячами ядер і прискорювачами, виконують моделювання кліматичних змін, аналізують розповсюдження епідемій, або обчислюють складні фізичні процеси. Це дозволяє вченим отримувати результати за дні чи години, що раніше займало б місяці обчислень.

У фінансовій галузі апаратне забезпечення також забезпечує ключові переваги. Банки та біржі використовують високошвидкісні сервери з низькими затримками для алгоритмічного трейдингу, що дає змогу виконувати мільйони

операцій за секунду. Це дозволяє реагувати на зміни ринку в реальному часі, проводити складні фінансові операції з мінімальними затримками та забезпечувати стабільність роботи фінансових систем навіть при високому навантаженні.

Крім того, в ритейлі використовуються спеціалізовані апаратні комплекси для управління точками продажу (POS-системи), інтерактивними дисплеями, аналітикою поведінки покупців і відстеженням товарів у реальному часі. Ці пристрої забезпечують не тільки швидке обслуговування клієнтів, але й можливість персоналізації пропозицій, що підвищує задоволеність покупців і стимулює їх повертатися до магазину.

Таким чином, новітнє апаратне забезпечення забезпечує конкретні практичні переваги для бізнесу: від пришвидшення аналітичних процесів і підвищення швидкості обробки даних до покращення безпеки й оптимізації інфраструктури в хмарних середовищах. Широкий спектр прикладів демонструє, як апаратне забезпечення знаходить застосування в різних галузях і сприяє оптимізації процесів, скороченню витрат і підвищенню продуктивності.

2. Програмне забезпечення

Комп’ютер сам по собі не має знань про ті галузі, в яких його застосовують; усі знання містяться в програмах, які виконуються на ньому. Змінюючи ці програми, можна перетворити комп’ютер на інструмент для роботи конструктора, астронома чи юриста. Програмне забезпечення також відоме під назвою software (в перекладі — «м’яке забезпечення»).

Програма — це чітко структурована послідовність команд, яку може виконувати обчислювальна машина. Завдяки програмам комп’ютер автоматично виконує обробку інформації. Програма має бути завантажена в пам’ять комп’ютера, щоб він міг слідувати визначеній послідовності дій для розв’язання конкретної задачі.

По суті, програма — це алгоритм, представлений у вигляді команд мовою програмування. Алгоритм же, у свою чергу, є кінцевим набором правил чи впорядкованих дій, які дозволяють механічно виконувати операції для розв'язання певного класу однотипних задач.

Програмне забезпечення є основним інструментом, який забезпечує взаємодію між апаратним забезпеченням і користувачем, дозволяючи ефективно виконувати задачі й обробляти дані. Основні складові програмного забезпечення можна поділити на кілька категорій:

1. **Операційні системи (ОС)**

Операційна система є базовим рівнем програмного забезпечення, яке відповідає за управління апаратними ресурсами, запуск прикладних програм і забезпечення взаємодії між різними пристроями. Вона також підтримує користувацький інтерфейс, що дозволяє людям взаємодіяти із системою. **Приклади:** Windows, Linux, macOS, Android, iOS.

2. **Системне програмне забезпечення.**

Сюди належать утиліти, драйвери пристроїв, програми для резервного копіювання, діагностики системи, антивірусні програми тощо. Вони допомагають забезпечити стабільну і безпечну роботу ОС і апаратного забезпечення.

Приклади: драйвери для відеокарт (NVIDIA, AMD), утиліти моніторингу системи (AIDA64, HWMonitor), антивірусні рішення (Kaspersky, McAfee).

3. **Системи управління базами даних (СУБД)**

СУБД дозволяють створювати, змінювати, організовувати і зберігати великі обсяги даних. Вони підтримують виконання запитів, забезпечують транзакційну цілісність, шифрування даних і реплікацію для забезпечення відмовостійкості.

Приклади: MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server.

4. **Прикладне програмне забезпечення**

Ця категорія охоплює всі програми, які безпосередньо використовуються кінцевими користувачами для виконання певних завдань. Це можуть бути

офісні пакети, графічні редактори, програми для обробки аудіо- та відеофайлів, веб-браузери, інструменти для управління проектами та фінансові додатки. **Приклади:** Microsoft Office, Adobe Photoshop, AutoCAD, Google Chrome, QuickBooks.

5. **Програмне забезпечення для розробки (інструменти розробника)**

Ця категорія включає середовища розробки (IDE), компілятори, дебагери, системи контролю версій та інші інструменти, які дозволяють створювати, тестувати і підтримувати нові програми.

Приклади: Visual Studio, Eclipse, IntelliJ IDEA, Git, Jenkins.

6. **Мережеве програмне забезпечення**

Програми, що забезпечують роботу мереж: сервіси для передачі файлів, поштові сервери, VPN-клієнти, платформи для відеоконференцій і віддаленого управління.

Приклади: OpenVPN, Microsoft Exchange Server, Zoom, TeamViewer.

Програмне забезпечення можна умовно поділити на три основні категорії:

1. **Системне програмне забезпечення.**

Це набір програм, що забезпечує базову функціональність комп'ютера. Системні програми контролюють роботу апаратного забезпечення, керують обчислювальними процесами та надають платформу для запуску інших програм. Основні компоненти цього типу ПЗ включають операційні системи (Windows, Linux, macOS), утиліти для управління файлами, драйвери пристроїв, програми резервного копіювання та антивірусне програмне забезпечення. **Приклад:** Операційна система Windows забезпечує доступ до файлів, керування пам'яттю, підключення до мережі, підтримку мультимедійних пристроїв і запуск прикладних програм.

2. **Прикладне програмне забезпечення.**

Це програми, які створюються для вирішення конкретних завдань кінцевого користувача. Вони дозволяють виконувати роботу з текстами, числами, графікою, базами даних, спілкуванням у мережі та багатьма іншими задачами. **Приклади:**

- **Офісні програми:** Microsoft Word, Excel, PowerPoint.
- **Мультимедійні програми:** Adobe Photoshop для редагування зображень або Audacity для обробки аудіо.
- **Ігри:** Графічно інтенсивні відеоігри, які використовують можливості системного ПЗ для забезпечення продуктивності й графіки.
- **Бізнес-додатки:** CRM-системи, ERP-рішення, бухгалтерські програми типу QuickBooks.

3. **Інструментарій для розробки програмного забезпечення.**

Ця категорія включає програми та середовища, які допомагають розробникам створювати, налагоджувати й підтримувати нові програми. До таких інструментів належать компілятори, інтерпретатори, інтегровані середовища розробки (IDE), системи контролю версій, бібліотеки та фреймворки. **Приклади:**

- **Мови програмування:** Python, Java, C++, JavaScript.
- **Інтегровані середовища розробки:** Visual Studio, IntelliJ IDEA, Eclipse.
- **Інструменти для тестування:** Selenium, JUnit, Postman для автоматизації тестів і перевірки функціональності API.
- **Системи контролю версій:** Git, GitLab, Bitbucket для збереження змін у коді, спільної роботи розробників і впровадження нових функцій.

Системне ПЗ, наприклад, драйвер принтера, дозволяє комп'ютеру взаємодіяти з принтером, забезпечуючи коректний друк документів. Без системного ПЗ апаратні компоненти не зможуть взаємодіяти між собою.

Прикладне ПЗ, наприклад, програма для відеоконференцій Zoom, дозволяє людям віддалено спілкуватися за допомогою аудіо- та відеозв'язку. Це особливо актуально для компаній, які мають віддалених співробітників, а також для освітніх закладів, які проводять онлайн-навчання.

Інструментарій для розробки, наприклад, платформа GitHub, дозволяє командам розробників одночасно працювати над великими проектами, впроваджувати нові функції, виправляти помилки й відстежувати історію змін у коді.

Таким чином, ці три категорії програмного забезпечення забезпечують повний цикл — від запуску та управління апаратним забезпеченням до вирішення прикладних завдань і створення нового програмного продукту.

Класифікація програмного забезпечення може здійснюватися за кількома основними критеріями, зокрема за функціональним призначенням, типами користувачів та методами розробки (Рис. 5.1). Найбільш поширеним є поділ на такі категорії:

1. Системне програмне забезпечення.

Системні програми забезпечують базову роботу комп'ютера і створюють середовище для запуску прикладних програм.

- **Операційні системи (ОС):** Windows, Linux, macOS, iOS, Android.
- **Утиліти:** програми для діагностики системи, резервного копіювання, управління дисками, антивіруси.
- **Драйвери:** програмне забезпечення для забезпечення взаємодії ОС з периферійними пристроями (принтерами, відеокартами, сканерами).
- **Системи віртуалізації:** VMware, VirtualBox, які дозволяють створювати віртуальні машини для запуску декількох ОС на одному фізичному пристрої.

2. Системи програмування – це набір інструментів, програмних компонентів та середовищ, що забезпечують розробку, відлагодження, тестування і підтримку програмного забезпечення. Вони полегшують процес створення програм, даючи розробникам зручні засоби для написання коду, його перевірки, компіляції і налагодження.

Основні складові систем програмування:

2.1. Мови програмування:

Основний засіб, яким користується розробник для створення програм. Мова програмування задає синтаксис та семантику для опису алгоритмів і структур даних.

- *Компільовані мови*: наприклад, C, C++, Go. Вони вимагають попередньої компіляції коду у виконуваний файл.

➢ *Інтерпретовані мови*: наприклад, Python, Ruby, JavaScript. Код виконується безпосередньо за допомогою інтерпретатора.

➢ *Гібридні мови*: наприклад, Java, C#. Вони компілюються у байт-код, який потім виконується віртуальною машиною.

2.2. Інтегровані середовища розробки (IDE):

Це програми, які надають розробникам єдиний інтерфейс для написання, редагування, компіляції, запуску і налагодження коду. IDE часто включають текстовий редактор, компілятор, інтегрований дебагер, підтримку систем контролю версій і додаткові плагіни для розширення функціональності.

Приклади IDE:

- Visual Studio
- IntelliJ IDEA
- Eclipse
- PyCharm
- NetBeans

2.3. Системи контролю версій (VCS):

Забезпечують збереження історії змін у коді, спільну роботу над проектами, відкат до попередніх версій та вирішення конфліктів між змінами різних розробників.

Приклади: Git, Mercurial, Subversion.

2.4.Компілятори і інтерпретатори:

Інструментальні мови та системи програмування використовуються для створення нових програм. Комп’ютер здатний виконувати інструкції, написані в машинному коді, де кожна команда є послідовністю нулів і одиниць. Програмування безпосередньо машинною мовою є дуже незручним і схильним до помилок. Тому для написання програм використовуються алгоритмічні або інструментальні мови, зрозумілі людині. Згодом спеціальні програми, звані трансляторами, перетворюють (транслюють) ці тексти на машинний код.

Види трансляторів:

- **Інтерпретатори**: обробляють програму по одному оператору, відразу виконуючи його, а потім переходять до наступного.
- **Компілятори**: спочатку зчитують, аналізують і перекладають весь програмний код у машинну мову, а вже після завершення трансляції виконують отриману програму.

Типи інструментальних мов:

- **Мови низького рівня**: близькі до машинної мови, наприклад, асемблери.
- **Мови високого рівня**: ближчі до природної мови людини, такі як Pascal, Basic, C/C++, мови для роботи з базами даних тощо.

Система програмування, окрім транслятора, включає текстовий редактор, компонувальник, стандартні бібліотеки, засоби налагодження та інструменти для автоматизації процесу розробки. Прикладами таких систем є Delphi, Visual Basic, Visual C++, Visual FoxPro та інші.

Компілятори перетворюють програмний код на мові програмування у виконуваний код, що розуміє комп'ютер. Інтерпретатори виконують програму рядок за рядком, не створюючи окремого виконуваного файлу.

Приклади компіляторів: GCC (GNU Compiler Collection), Clang, Microsoft C++ Compiler.

Приклади інтерпретаторів: CPython для Python, Node.js для JavaScript.

Системи програмування значно спрощують роботу програмістів, автоматизуючи багато рутинних задач. Вони допомагають уникнути помилок, забезпечують більшу продуктивність, полегшують спільну роботу над кодом і дозволяють розробникам зосередитися на створенні якісних рішень, а не на низькорівневих технічних деталях.

3. Прикладне програмне забезпечення.

Ця категорія включає програми, які користувачі застосовують для виконання конкретних задач.

- **Офісні додатки:** текстові редактори (Microsoft Word, Google Docs), електронні таблиці (Excel, Google Sheets), програми для створення презентацій (PowerPoint).

- **Мультимедійні програми:** Adobe Photoshop, CorelDRAW для роботи із зображеннями, Premiere Pro для відеомонтажу, VLC для відтворення відео.
- **Ігри та розважальні додатки:** ігрові платформи, програми для потокового відтворення музики та відео.
- **Бізнес-додатки:** CRM-системи, ERP-рішення, бухгалтерські програми.
- **Навчальні програми:** програми для онлайн-курсів, тренажери, системи тестування знань.

Рис. 5.1. Класифікація програмного забезпечення

Програмне забезпечення для розробників. Сюди належать інструменти, які використовуються для створення нових програм.

- **Мови програмування та компілятори:** Python, C++, Java, компілятори GCC, Clang.
- **Інтегровані середовища розробки (IDE):** Visual Studio, IntelliJ IDEA, Eclipse, PyCharm.
- **Системи контролю версій:** Git, GitLab, Bitbucket, які дозволяють відстежувати зміни в коді та спільно працювати над проектами.
- **Інструменти для автоматизації тестування:** Selenium, JUnit, Postman.
- **Фреймворки:** Django, Spring, React, Angular, які полегшують розробку веб-додатків і сервісів.

2. **Вбудоване програмне забезпечення.**

Це програмне забезпечення, яке вбудоване в пристрої для управління їхньою роботою.

- **Мікропрограми (firmware):** забезпечують базову функціональність електронних пристроїв (смартфонів, роутерів, фотоапаратів).
- **Програмне забезпечення для промислових контролерів:** програми, які управляють виробничим обладнанням, системами автоматизації та роботами.

Інші підходи до класифікації:

- **За типами ліцензій:**
  - **Комерційне ПЗ:** програмне забезпечення, яке продається або надається за передплатою.
  - **Вільне ПЗ (open-source):** програми, код яких є відкритим, і вони доступні для модифікації (наприклад, Linux, LibreOffice).
  - **Безкоштовне (freeware) та умовно безкоштовне (shareware) ПЗ.**
- **За платформою:**
  - **Десктопне ПЗ:** програми для персональних комп'ютерів (Windows, macOS).
  - **Мобільне ПЗ:** додатки для смартфонів і планшетів (iOS, Android).
  - **Хмарне ПЗ:** сервіси, що працюють у браузері або через Інтернет (Google Workspace, Microsoft 365).
  - **Корпоративне ПЗ:** системи, які використовуються в великих організаціях для підтримки бізнес-процесів.

Таким чином, класифікація програмного забезпечення охоплює широкий спектр категорій і підходів, що дозволяє краще розуміти його різновиди, сфери застосування та особливості використання.

Перспективи розвитку програмного забезпечення: Сучасні тенденції в розробці програмного забезпечення включають:

- перехід до хмарних додатків і платформ як послуги (PaaS);

- активне впровадження машинного навчання і штучного інтелекту для автоматизації процесів;
- створення адаптивного ПЗ, яке може працювати на різних пристроях та операційних системах;
- підвищення рівня безпеки програм шляхом вбудованих механізмів захисту;
- використання технологій контейнеризації та оркестрації (наприклад, Docker і Kubernetes) для більшої гнучкості в розгортанні додатків.

Загалом, програмне забезпечення продовжує залишатися основним драйвером цифрової трансформації, допомагаючи організаціям і користувачам вирішувати все більш складні завдання у різних сферах діяльності.

3. Тенденції розвитку апаратного і програмного забезпечення

Комп'ютерна техніка та периферійне обладнання належать до галузей, які демонструють найшвидші темпи розвитку. Протягом останніх десяти років продуктивність комерційних процесорів значно зросла: їхня тактова частота збільшилася з 33 МГц до 3800 МГц. Обсяг оперативної пам'яті зріс із 16 Мб до 1 Гб. Окрім цього, технічні характеристики периферійних пристроїв значно покращилися, а їхня вартість суттєво зменшилася.

Однак фахівці зауважують, що сучасні напівпровідникові технології наближаються до своїх фізичних обмежень. Подальше зменшення розмірів транзисторів та провідників обмежується хімічними та фізичними властивостями використовуваних матеріалів. Тому в багатьох лабораторіях ведуться активні дослідження альтернативних підходів, таких як використання оптичних елементів або органічних молекул для збереження та зчитування інформації.

Одним із перспективних напрямів збільшення обчислювальної потужності є перехід на 64-розрядні процесори. На ринку вже з'явилися моделі, такі як Intel Itanium і AMD Opteron, але їхнє застосування наразі обмежене через

потребу в 64-розрядних операційних системах та відповідному прикладному програмному забезпеченні. Для повноцінного впровадження нових платформ необхідно створити нові компілятори, адаптувати стандарти розробки та поступово перенести на нову архітектуру комерційні програмні продукти.

Подальший розвиток комп'ютерної техніки залежить не лише від фізичних параметрів чипів, але й від концептуальних змін в архітектурі обчислювальних систем. У галузі активно обговорюється впровадження квантових обчислень, які, попри свої складнощі, обіцяють багатократне зростання обчислювальної потужності. У той же час розробники приділяють увагу системам нейроморфних процесорів, які імітують роботу мозку і дозволяють ефективніше обробляти великі обсяги даних, наприклад, у задачах машинного навчання [25].

Варто також зазначити, що паралельно зі збільшенням швидкості та ємності окремих компонентів, активно розвиваються підходи до оптимізації енергоспоживання. Більшість сучасних процесорів і периферійних пристроїв мають режими енергозбереження, а новітні технології виробництва дозволяють створювати чипи з меншою витратою енергії на одиницю обчислень. Це стає особливо актуальним для мобільних і портативних пристроїв, які потребують тривалого часу роботи від батареї.

Тож, майбутнє комп'ютерної техніки і пов'язаної з нею периферії може визначатися не лише фізичними досягненнями в галузі мікроелектроніки, а й впровадженням нових підходів до архітектури систем, використанням передових матеріалів і розробкою інноваційного програмного забезпечення, здатного повною мірою використовувати можливості нових апаратних платформ.

Ще однією важливою тенденцією є збільшення рівня інтеграції та зменшення розмірів багатьох компонентів персональних комп'ютерів. Ведуться розробки нових мікросхем, які поєднують у собі функції центрального процесора, оперативної пам'яті, контролерів введення-виведення та інших

компонентів. Це відкриває можливості для створення нових поколінь мобільних комп'ютерів, що забезпечують більш широкі функції комунікації.

На ринку вже з'явилися пристрої, які поєднують функціональність персонального комп'ютера, мобільного телефону та GPS-приймача. Такі пристрої здатні визначати свої координати, передавати їх через Інтернет до найближчого сервісного центру, отримувати карти місцевості та відображати їх на екрані разом із прокладеним маршрутом. Для підтримки таких систем розробляються спеціалізовані бази даних з інформацією про міста та туристичні об'єкти, які можуть бути збережені на flash-картах або компактних жорстких дисках ємністю 4-8 Гб.

Тенденції розвитку апаратного забезпечення:

1. **Збільшення обчислювальної потужності:**

Процесори стають багатоядерними, із більшими тактовими частотами та вдосконаленими архітектурами, що дозволяє значно підвищити продуктивність. **Приклад:** Процесори Intel Core 12-го покоління чи AMD Ryzen 7000 серій мають більше ядер і потоків, що дає змогу ефективно працювати з багатозадачними навантаженнями, такими як обробка відео 4K чи 3D-рендеринг.

2. **Мініатюризація та інтеграція:**

Зменшуються фізичні розміри чипів, і багато функцій, які раніше виконували окремі компоненти, інтегруються в один процесор (наприклад, графічні ядра в CPU, вбудовані контролери). **Приклад:** Apple M1 та M2 – приклади процесорів із високим рівнем інтеграції: центральне процесорне ядро, графічний процесор, оперативна пам'ять і контролери введення-виведення інтегровані на одному чипі. Це забезпечує вищу продуктивність при зменшенні енергоспоживання.

3. **Підвищення енергоефективності:**

Нові процесори та компоненти споживають менше енергії при збільшенні продуктивності, що дозволяє створювати більш екологічні рішення та продовжувати автономну роботу мобільних пристроїв. **Приклад:** ARM-процесори, які використовуються в більшості смартфонів і планшетів,

споживають мінімум енергії, дозволяючи пристроям працювати годинами без підзарядки.

4. **Розвиток нових матеріалів та технологій:**

Дослідження в галузі оптичних чипів, квантових обчислень і нейроморфних процесорів спрямовані на подолання фізичних обмежень традиційних напівпровідникових технологій. **Приклад:** IBM розробляє 2-нм транзистори, які, за очікуваннями, забезпечать значний приріст продуктивності та зниження енергоспоживання в порівнянні з сучасними технологіями.

5. **Мережеві та комунікаційні технології:**

Розширення використання 5G, Wi-Fi 6, і Ethernet 400G створює нові можливості для швидкої передачі даних та інтеграції з хмарними сервісами. **Приклад:** Впровадження Wi-Fi 6E, який пропонує розширений частотний діапазон, дозволяє значно зменшити затримки та збільшити пропускну здатність, що особливо корисно для відеоконференцій, онлайн-ігор та потокового відео високої якості.

Тенденції розвитку програмного забезпечення:

1. **Перехід до хмарних технологій:**

Все більше програм переміщуються у хмару, де користувачам доступні високопродуктивні обчислення, масштабовані сховища даних та сервіси в режимі "pay-as-you-go". **Приклад:** Microsoft 365 і Google Workspace надають хмарні інструменти, які дозволяють співпрацювати в реальному часі над документами, таблицями та презентаціями, незалежно від фізичного місцезнаходження користувачів.

2. Зростання ролі штучного інтелекту та машинного навчання: Інструменти та бібліотеки для AI/ML інтегруються в більшість програмних платформ, надаючи можливість створювати інтелектуальні системи, аналізувати великі обсяги даних і приймати рішення в реальному часі. **Приклад:** Хмарні сервіси, такі як AWS SageMaker, Google AI Platform або Azure AI, надають розробникам інструменти для швидкої побудови моделей

машинного навчання. Ці моделі використовуються у всіх галузях – від прогнозування продажів у ритейлі до аналізу медичних зображень.

3. **Впровадження DevOps і CI/CD практик:**

Інструменти для автоматизації тестування, побудови і розгортання програм дозволяють розробникам швидше доставляти нові функції користувачам, забезпечуючи безперервність процесів розробки. **Приклад:** Jenkins та GitHub Actions дозволяють автоматизувати процес розгортання нових версій програмного забезпечення, скорочуючи час між написанням коду і його виходом у продакшн.

4. **Підвищення рівня безпеки:**

Програмне забезпечення стає більш захищеним завдяки інтегрованим засобам шифрування, вбудованим системам виявлення загроз і використанню кращих практик розробки безпечного коду. **Приклад:** Засоби, такі як HashiCorp Vault, забезпечують безпечне управління секретами та доступами, а також допомагають інтегрувати шифрування на рівні програмного забезпечення**.**

5. **Контейнеризація та оркестрація:**

Використання контейнерів (Docker) і платформ оркестрації (Kubernetes) дозволяє програмам працювати незалежно від апаратного середовища, полегшує масштабування і забезпечує гнучкість у виборі платформ. **Приклад:** Docker та Kubernetes стали стандартом у розгортанні мікросервісів. Вони дозволяють легко масштабувати програми в хмарних середовищах, таких як Amazon ECS або Google Kubernetes Engine.

6. **Мультиплатформність і кросплатформені рішення:**

Програмне забезпечення розробляється з урахуванням роботи на різних пристроях і платформах. Це підвищує зручність використання для кінцевих користувачів.

7. **Інтеграція інтерфейсів з природною мовою та голосовим керуванням:**

Програми все частіше використовують голосові помічники, чат-боти та інтерфейси, які спрощують взаємодію з користувачами. **Приклад:** Amazon

Alexa Skills або Google Dialogflow дозволяють створювати голосові помічники та чат-боти, які автоматизують клієнтську підтримку або допомагають у повсякденних завданнях.

Загальний напрямок: Розвиток апаратного і програмного забезпечення рухається в бік підвищення продуктивності, зручності, енергоефективності та безпеки. Інтеграція нових технологій, таких як штучний інтелект, квантові обчислення, інтернет речей (IoT) і хмарні сервіси, визначає перспективи розвитку галузі на найближчі роки.

Зростання обчислювальної потужності, мініатюризація компонентів, інтеграція штучного інтелекту в програмні рішення, а також розвиток хмарних сервісів і нових комунікаційних технологій є основними напрямками сучасних тенденцій у розвитку апаратного і програмного забезпечення. Це дозволяє створювати більш продуктивні, енергоефективні й універсальні рішення, які відкривають нові можливості в бізнесі, науці, медицині, освіті й інших галузях.

Контрольні питання

1. Що таке апаратне забезпечення інформаційної системи і які його основні компоненти?
2. Чим відрізняються процесори різних архітектур і як це впливає на продуктивність системи?
3. Які функції виконує оперативна пам’ять у складі апаратного забезпечення?
4. Що належить до периферійних пристроїв і яка їх роль у роботі інформаційної системи?
5. Що таке програмне забезпечення і як його поділяють за типами?
6. Чим відрізняється системне програмне забезпечення від прикладного?
7. Які мови програмування належать до інструментального програмного забезпечення?
8. Як технології хмарних обчислень змінюють підхід до розробки програмного забезпечення?

Тема 6.
БАЗИ ДАНИХ ТА УПРАВЛІННЯ ДАНИМИ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Мета: ознайомлення здобувачів із базовими поняттями, принципами та технологіями, пов'язаними з базами даних та управлінням даними в сучасних інформаційних системах. Лекція має на меті пояснити, як організовуються, зберігаються дані, чому бази даних є ключовим компонентом інформаційної інфраструктури, а також навчити основам моделювання даних, проектування баз даних і використання систем управління базами даних (СУБД). У результаті студенти отримають розуміння важливості даних, їхньої структури, способів доступу та методів забезпечення цілісності, безпеки й ефективності роботи з ними.

План лекції

1. Архітектура баз даних та моделі даних.
2. Програмні і мовні засоби баз даних.
3. Архітектура інформаційної системи.

1. Архітектура баз даних та моделі даних

Для забезпечення ефективної роботи з базами даних необхідно відокремити прикладні програми від самих даних. Це важливо, оскільки зміни в системі, а також оптимізація обслуговування користувачів часто потребують модифікації методів зберігання даних у базі, шляхів доступу до них, структури та форматів даних, а також зв'язків між елементами. Якщо не використовувати спеціальні підходи та при розробці програмного забезпечення прописувати методи доступу до даних і їх формати безпосередньо у коді, то будь-яка зміна у базі даних вимагатиме значного коригування прикладного коду, що може призводити до суттєвих затрат часу та ресурсів.

Для запобігання цим труднощам використовуються засоби систем управління базами даних (СУБД), які забезпечують незалежність прикладного програмного забезпечення від даних. Такий підхід дозволяє користувачам працювати з базою даних, не знаючи її внутрішнього представлення та механізмів зберігання.

На рисунку 6.1 представлено трирівневу архітектуру СУБД, яку розробив Комітет з планування стандартів SPARC (Standards Planning and Requirements Committee) Американського національного інституту стандартів ANSI (American National Standards Institute).

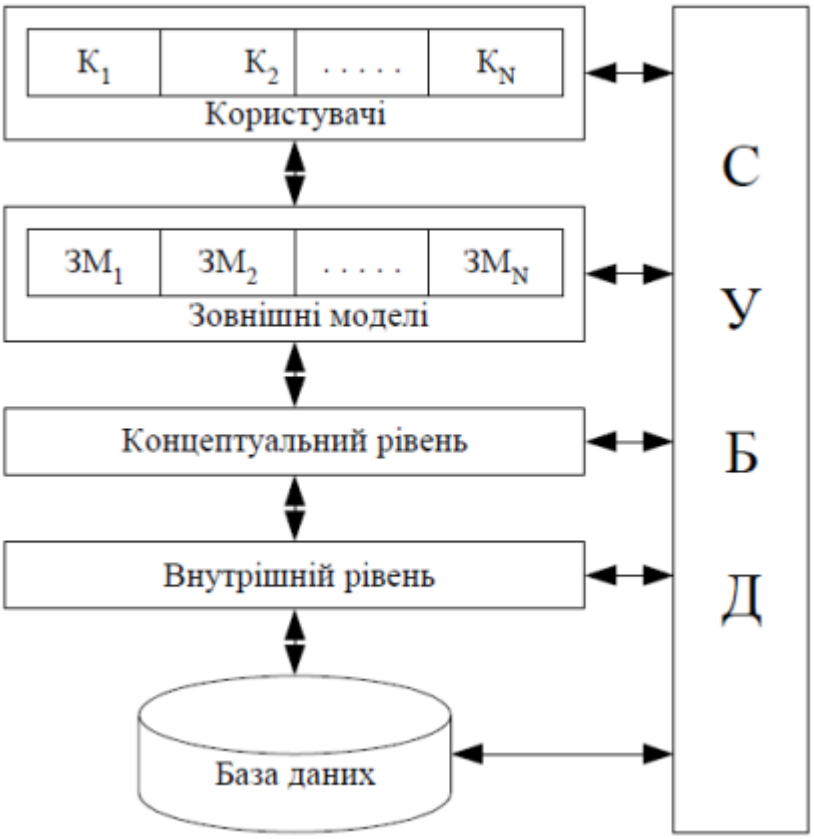

Рис. 6.1. Трирівнева архітектура СУБД

Опис структури даних на будь-якому рівні називається схемою. Згідно з рівнями абстракції в архітектурі СУБД, виділяють три типи схем бази даних. На найвищому рівні знаходяться кілька зовнішніх схем, які представляють різні точки зору на дані. Цей рівень визначає, як конкретне застосування бачить та

обробляє базу даних, надаючи доступ лише до тих даних, які потрібні для його роботи.

На концептуальному рівні опис бази даних називається концептуальною схемою. Вона представляє базу даних у загальному вигляді, охоплюючи всі дані, які використовують різні застосування. По суті, концептуальний рівень відображає модель предметної області, для якої створено базу даних.

На внутрішньому рівні база даних описується внутрішньою схемою. У цьому випадку база даних представлена як набір файлів, які відображають її фізичну організацію та структуру.

Трирівнева архітектура СУБД забезпечує незалежність від даних. Це означає, що зміни на нижніх рівнях не впливають на верхні рівні. У цій моделі виділяють два типи незалежності: логічну і фізичну.

Логічна незалежність від даних означає, що зовнішні схеми залишаються незмінними навіть тоді, коли в концептуальну схему вносять корективи. Іншими словами, якщо змінюється концептуальна структура бази даних, це не вимагає змін у програмах, які використовують зовнішні схеми.

Приклад логічної незалежності:

Уявімо базу даних університету. Спочатку концептуальна схема включає лише інформацію про студентів і курси. Через деякий час університет вирішує додати інформацію про стипендії. Нові таблиці або поля додаються до концептуальної схеми, але вони ніяк не впливають на вже існуючі зовнішні схеми, які використовуються для реєстрації студентів на курси. Усі програми, що керуються зовнішніми схемами, продовжують функціонувати, не знаючи про зміни в концептуальному рівні.

Фізична незалежність від даних означає, що зміни у внутрішній схемі не впливають на концептуальну та зовнішні схеми. Наприклад, якщо змінюються файлові системи, пристрої зберігання даних або структура пошукових індексів, це не потребує внесення змін у вищі рівні представлення даних чи у прикладні програми.

Приклад фізичної незалежності:

Наприклад, компанія, яка використовує систему управління базою даних для зберігання інформації про клієнтів і замовлення, спочатку працює з традиційними жорсткими дисками (HDD). Після модернізації вони вирішують перейти на твердотільні накопичувачі (SSD) або навіть на розподілене сховище в хмарі. Водночас змінюється структура файлів на внутрішньому рівні: можливо, додаються нові індекси, інші методи компресії або дублювання даних. Незважаючи на ці зміни, прикладні програми, які використовують дані через зовнішні схеми, залишаються незмінними. Вони не помічають, що фізична структура зберігання даних була змінена.

Модель даних — це спосіб формалізації даних, який фокусує увагу на ключових елементах певної предметної області, нехтуючи деталями, які не впливають на основну функціональність. На рисунку 6.2. відображена класифікація моделей баз даних. Вона слугує концептуальним інструментом для відображення структур, зв'язків, правил і обмежень, характерних для інформаційної системи.

Основні складові моделі даних:

1. **Структурна складова:**

Ця частина моделі визначає, як організовуються дані, які типи даних допускаються і як вони пов'язані між собою. Наприклад, у реляційній моделі структурною складовою є таблиці (відношення), стовпці (атрибути) і рядки (кортежі). У документаційній моделі даних, як у MongoDB, це може бути набір документів у форматі JSON із вкладеними структурами.
Приклад: У базі даних для університету таблиця "Студенти" може мати атрибути: ID студента, прізвище, ім'я, спеціальність. Це відображає структуру даних, яка забезпечує єдиний формат збереження інформації про всіх студентів.

2. **Керуюча складова:**

Визначає операції, які можна виконувати з цими даними. Наприклад, у реляційній моделі доступні операції вставки, видалення, оновлення та пошуку даних за допомогою SQL-запитів. У графовій моделі можна здійснювати такі дії, як додавання нових вузлів, створення нових ребер (зв'язків) і виконання

складних графових запитів для виявлення шляхів між вузлами.
Приклад: В університетській базі даних керуюча складова може визначати, що адміністратор має право додавати нових студентів до таблиці "Студенти", викладачі можуть оновлювати інформацію про спеціальності, а оператори технічної підтримки можуть переглядати дані, але не змінювати їх.

3. **Обмеження цілісності:**

Це набір правил, які гарантують, що дані в базі залишатимуться коректними та узгодженими. У реляційній моделі обмеження цілісності включають первинні ключі, зовнішні ключі, унікальні значення атрибутів, обов'язковість полів тощо. В інших моделях можуть бути свої механізми, наприклад, схемові валідатори в документаційних базах даних.

Приклад: У таблиці "Студенти" кожен запис повинен мати унікальний ID (первинний ключ), спеціальність має відповідати одному з дозволених значень (довідкова таблиця спеціальностей), а ім'я та прізвище не можуть бути порожніми.

Додаткові приклади:

- У медичній інформаційній системі модель даних може передбачати, що пацієнт має лише одне унікальне медичне карткове число (структурна складова), що лікарі можуть додавати записи про лікування або оновлювати їх (керуюча складова), а також що діагнози та процедури не можуть бути прив'язані до неіснуючого пацієнта (обмеження цілісності).
- В онлайн-магазині структурна складова моделі даних може відображати таблиці "Товари", "Клієнти", "Замовлення". Керуюча складова дозволяє адміністраторам додавати нові товари, а клієнтам оформлювати замовлення. Обмеження цілісності гарантують, що замовлення можуть бути створені тільки на основі існуючих товарів і за умови наявності актуальної інформації про клієнта.

Таким чином, модель даних не лише формує основу для збереження інформації, але й визначає, як ці дані можна використовувати, обмежуючи або розширюючи можливості користувачів і програмних додатків.

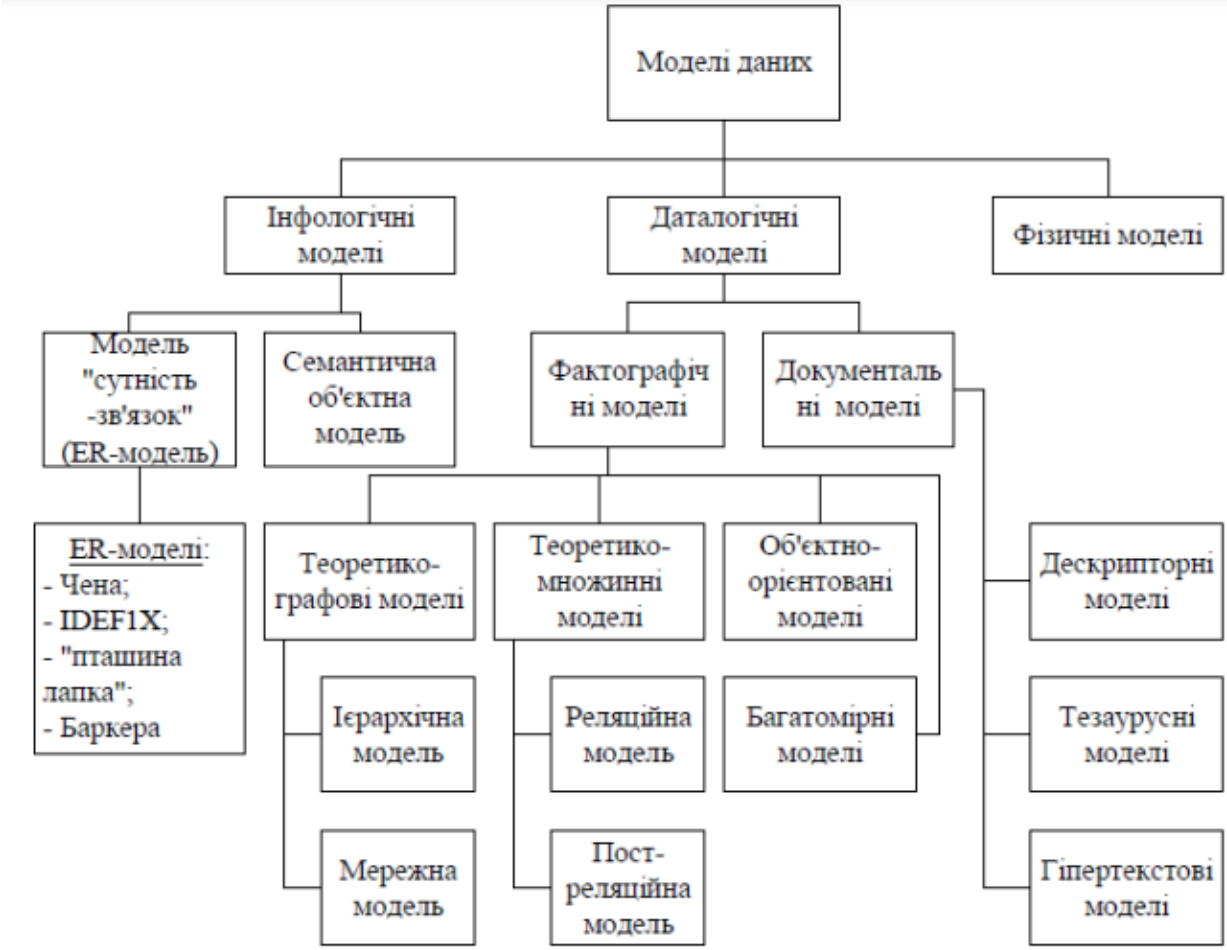

Рис. 6. 2. Класифікація моделей баз даних

На кожному рівні представлення інформації використовується своя модель. *Інфологічна модель* надає уявлення про предметну область у вигляді, незалежному від конкретної СУБД. Вона представляє інформаційно-логічний рівень абстракції, який описує об’єкти предметної області, їх властивості та взаємозв’язки.

Зазвичай інфологічні моделі ототожнюються з концептуальними моделями предметної області. У цьому контексті їх називають *концептуальними інфологічними моделями*, які можуть бути поділені на внутрішні та зовнішні.

Приклад 1:

В університетській інформаційній системі інфологічна модель може відображати такі об’єкти, як студенти, викладачі, курси та групи. Модель визначає, що студент має ім’я, прізвище, номер залікової книжки, спеціальність

і належить до певної групи. Викладач має ім’я, прізвище, посаду і прив’язаний до певних курсів. Група має унікальний номер, спеціальність і список студентів, які до неї входять. Ці об’єкти та їхні зв’язки описані на логічному рівні незалежно від того, яку СУБД буде використано.

Приклад 2:

Для медичної інформаційної системи інфологічна модель може включати об’єкти «Пацієнт», «Лікар», «Прийом» та «Діагноз». У цій моделі вказується, що пацієнт має унікальний номер, прізвище, ім’я, дату народження та адресу. Лікар має ім’я, прізвище, спеціальність та список пацієнтів, яких він лікує. Прийом пов’язаний із конкретним пацієнтом, лікарем та датою, а також має список поставлених діагнозів. Ця структура абстрактно описує предметну область, незалежно від того, яка система управління базами даних буде застосована для її реалізації.

Даталогічна модель є логічним представленням даних, яке фокусується на зв’язках між елементами інформації, незалежно від їхнього конкретного змісту чи фізичного способу зберігання. Іншими словами, даталогічна модель описує структуру даних і логіку їхніх взаємозв’язків, залишаючи поза увагою деталі реалізації та середовище, в якому ці дані зберігаються. Часто ці моделі ототожнюють із логічними моделями даних.

Приклад 1:

Реляційна модель є одним із найвідоміших прикладів даталогічної моделі. Вона представляє дані у вигляді таблиць (відношень), у яких кожен рядок відповідає запису (кортежу), а кожен стовпець – атрибуту. Наприклад, у системі для інтернет-магазину реляційна модель може включати таблиці «Користувачі» (UserID, Name, Email), «Товари» (ProductID, ProductName, Price) і «Замовлення» (OrderID, UserID, ProductID, Quantity). Логічні зв’язки між цими таблицями – наприклад, зв’язок між користувачами та їхніми замовленнями через UserID – визначаються незалежно від того, як ці дані фізично зберігаються.

Приклад 2:

У графовій моделі, яка також відображає логічні зв’язки, дані представлені у вигляді вузлів і ребер. Наприклад, у системі соціальної мережі вузли можуть відповідати користувачам, а ребра – зв’язкам дружби або підпискам. Вузол «Користувач А» може бути пов’язаний з вузлом «Користувач В» ребром типу «Дружба». Логічні зв’язки (хто з ким пов’язаний і яким чином) не залежать від того, чи ці дані зберігаються в реляційній СУБД, графовій базі даних або іншій системі.

Приклад 3:

У документно-орієнтованій моделі, яка також є логічним представленням даних, дані зберігаються у формі документів (наприклад, у JSON чи XML). Логічна структура документа включає ключі та вкладені об’єкти, які дозволяють встановити логічні зв’язки між даними. Наприклад, у базі даних для електронної бібліотеки документ може містити назву книги, автора, рік публікації і розділи з коротким змістом. Зв’язок між автором і його книгами або між розділами і книгою логічно визначається в структурі документа, незалежно від того, як цей документ буде фізично збережений.

Отже, даталогічна модель забезпечує структуроване уявлення про зв’язки між елементами даних і їхні взаємовідносини, що дозволяє гнучко працювати з даними незалежно від середовища збереження.

Фізична модель — це опис того, як дані фактично розміщені та організовані у фізичному середовищі комп’ютера. Вона відображає конкретні способи зберігання даних, включаючи структуру записів, порядок їх розташування на носіях інформації, а також доступні методи для їх пошуку і зчитування.

Деталі фізичної моделі:

1. **Структура записів:**

Описує, у якому вигляді дані зберігаються — наприклад, фіксована чи змінна довжина записів, використання блоків або сторінок пам’яті.

2. **Організація файлів:**

Вказує, чи дані зберігаються у послідовному порядку, у вигляді розсіяного файлу або з використанням індексів.

3. **Шляхи доступу:**

Включає інформацію про індекси, хеш-таблиці, B-дерева чи інші структури даних, які прискорюють пошук і вибірку інформації.

Приклад 1:

Розглянемо базу даних для обліку клієнтів. На фізичному рівні дані можуть бути збережені у файлі, розбитому на сторінки, кожна з яких має фіксований розмір, наприклад 8 Кб. Записи про клієнтів впорядковані за унікальним ідентифікатором (CustomerID), а індекс у вигляді B-дерева забезпечує швидкий доступ до потрібного запису. Ця фізична модель дозволяє швидко знаходити клієнта за його ID без необхідності переглядати весь файл.

Приклад 2:

У випадку інтернет-магазину дані про товари можуть зберігатися у таблиці з великим обсягом записів. На фізичному рівні використовується розподіл записів на кілька файлів для оптимізації роботи з великими обсягами інформації. Індекси на основі хеш-таблиць дозволяють швидко знайти товар за його унікальним кодом, а індекси за категоріями допомагають ефективно фільтрувати товари.

Таким чином, фізична модель зосереджена на конкретних технічних аспектах зберігання і доступу до даних, що дозволяє оптимізувати продуктивність і забезпечити надійність збереження інформації.

Модель "сутність-зв'язок" (Entity-Relationship, або ER-модель) — це концептуальне представлення предметної області, яке включає сутності, зв'язки між ними та їхні атрибути. Сутності в ER-моделі — це ключові об'єкти предметної області, які мають певні властивості (атрибути) і унікальний ідентифікатор. Зв'язки визначають, як сутності взаємодіють одна з одною, а також встановлюють правила цілісності даних. Одним із таких правил є залежність по існуванню, що означає, що одна сутність не може існувати без іншої.

ER-моделі дозволяють візуалізувати структуру даних у вигляді графічних діаграм, що полегшує розуміння предметної області та планування майбутньої бази даних. Вони часто є невід'ємною частиною інструментів автоматизованого проектування баз даних (CASE-продуктів).

Приклад 1:

Університетська інформаційна система

Сутності:

- Студент: атрибути — ID студента, ім'я, прізвище, спеціальність.
- Курс: атрибути — код курсу, назва, кількість кредитів.
- Викладач: атрибути — ID викладача, ім'я, кафедра.

Зв'язки:

- «Студент записаний на курс» (зв'язок між сутностями «Студент» і «Курс»).
- «Курс викладається викладачем» (зв'язок між сутностями «Курс» і «Викладач»).

Цілісність:

- Залежність по існуванню: студент не може бути записаним на курс, якщо курс не існує.

Приклад 2: Інтернет-магазин

Сутності:

- Товар: атрибути — код товару, назва, ціна, кількість на складі.
- Клієнт: атрибути — ID клієнта, ім'я, адреса, телефон.
- Замовлення: атрибути — номер замовлення, дата, статус.

Зв'язки:

- «Клієнт оформлює замовлення» (зв'язок між «Клієнт» і «Замовлення»).
- «Замовлення містить товар» (зв'язок між «Замовлення» і «Товар»).

Цілісність:

- Залежність по існуванню: замовлення не може містити товар, якщо цього товару немає в базі даних.

Таким чином, ER-модель є зручною і зрозумілою основою для проектування баз даних, оскільки дозволяє розділити предметну область на сутності, зрозуміти їхні взаємозв’язки, а також забезпечити цілісність і узгодженість даних у системі.

Семантична об’єктна модель (СОМ) — це модель даних, яка представляє предметну область у вигляді семантичних об’єктів і їхніх властивостей. У межах цієї моделі об’єкти мають сукупність атрибутів, які описують їх характеристики. Атрибути групуються у класи, що дозволяє логічно організувати дані та їхню структуру. На відміну від теоретико-множинних чи теоретико-графових моделей, семантична об’єктна модель надає більш розвинені механізми для представлення семантики даних. Вона краще відображає контекст предметної області, полегшує розуміння даних і створення складних зв’язків між ними.

Основні компоненти моделі:

1. **Семантичні об’єкти:**Це основні елементи моделі, що відповідають реальним сутностям предметної області. Наприклад, у системі управління персоналом такими об’єктами можуть бути "Співробітник", "Посада" чи "Відділ".

2. **Атрибути:** Кожен об’єкт має набір атрибутів, які описують його характеристики. Наприклад, атрибути об’єкта "Співробітник" можуть включати "Ім’я", "Прізвище", "Дата народження", "Посада".

3. **Класи:** Атрибути групуються в класи, які забезпечують структурування і логічну організацію даних. Наприклад, всі співробітники компанії можуть бути представлені у класі "Співробітники", а кожен окремий співробітник — як об’єкт цього класу.

Приклади застосування:

- **Система управління проектами:**

Семантичні об’єкти — "Проект", "Учасник", "Завдання". Атрибути "Проекту" можуть включати "Назва проекту", "Термін виконання", "Керівник проекту". "Завдання" можуть мати атрибути "Опис",

"Відповідальний", "Стан виконання". Об’єкти групуються в класи: "Проекти", "Завдання", "Учасники".

- **Електронна медична картка:**

Семантичні об’єкти — "Пацієнт", "Лікар", "Прийом". Атрибути "Пацієнта" — "Ім’я", "Прізвище", "Номер картки", "Дата народження". Атрибути "Прийому" — "Дата", "Діагноз", "Призначення". Усі пацієнти об’єднуються в клас "Пацієнти", всі прийоми — в клас "Прийоми".

Переваги семантичної об’єктної моделі:

- Більш інтуїтивне відображення реальних об’єктів та їхніх характеристик.
- Покращене розуміння зв’язків між даними завдяки класифікації і групуванню.
- Більша гнучкість при внесенні змін до структури даних.
- Можливість опису складних взаємозв’язків та ієрархій.

COM добре підходить для систем, де важливо чітко визначити контекст даних, забезпечити їхню семантичну цілісність і надати розробникам та аналітикам більш зрозуміле представлення предметної області.

Теоретико-графова модель даних — це модель, у якій дані організовані у вигляді графа, що може бути як загальним графом, так і його окремим випадком, наприклад деревом. Ключова особливість цієї моделі полягає у використанні графових структур для представлення відношень між даними, де вузли графа відповідають об’єктам (сутностям), а ребра — зв’язкам між ними.

Операції на основі теоретико-графової моделі:

Маніпулювання даними в цій моделі здійснюється за допомогою навігаційних операцій. Навігаційні операції передбачають пошук або переміщення між вузлами графа, наприклад, перехід від вузла до вузла через ребро, обходи графа, або виконання певних дій над знайденими вузлами. Такі операції мають позаописовий характер, тобто не описують результат як логічну умову (як у реляційних запитах), а вказують, як саме треба досягти бажаного результату.

Приклади використання:

1. **Мережні бази даних:**

У мережних моделях даних (заснованих на теоретико-графовій моделі) інформація зберігається у вигляді вузлів і зв’язків. Наприклад, у базі даних компанії може бути вузол «Проект», пов’язаний із вузлами «Співробітник» (хто працює над проектом), «Клієнт» (для якого виконується проект) і «Договір». Навігаційна операція дозволяє, наприклад, від вузла «Проект» перейти до пов’язаних з ним співробітників і отримати інформацію про їхні ролі у проекті.

2. **Системи управління ієрархічними структурами:**

Теоретико-графова модель може застосовуватися для управління ієрархіями. Наприклад, дерево каталогу файлів у файловій системі. Вузли дерева відповідають папкам і файлам, а ребра визначають вкладеність папок. Навігаційні операції дозволяють переміщатися по ієрархії, переходячи до дочірніх або батьківських вузлів, і виконувати дії, наприклад, перерахувати всі файли в папці або знайти шлях до кореневого каталогу.

3. **Графові бази даних:**

У сучасних графових базах даних, таких як Neo4j або ArangoDB, теоретико-графова модель лежить в основі їхньої архітектури. Наприклад, соціальна мережа може бути представлена як граф, де вузли — це користувачі, а ребра — зв’язки між ними (дружба, підписка тощо). Навігаційні операції дозволяють від одного користувача пройти по зв’язках, щоб знайти друзів, друзів друзів або визначити найкоротший шлях між двома користувачами.

Переваги теоретико-графової моделі:

- Гнучкість у моделюванні складних зв’язків між даними.
- Природність представлення ієрархій і мережевих структур.
- Підтримка складних навігаційних запитів, які важко реалізувати в інших моделях.

Обмеження:

- Потреба в явному вказанні шляхів доступу до даних, що може ускладнити розробку.

- Відсутність описових запитів (як у реляційній моделі), що може ускладнювати оптимізацію і автоматизацію.

Таким чином, теоретико-графова модель ідеально підходить для тих предметних областей, де важливі складні зв’язки між об’єктами і потрібна швидка навігація між ними.

Ієрархічна модель даних заснована на деревоподібній структурі, де інформація організована у вигляді ієрархії «батько-нащадок». Основні одиниці цієї моделі — записи, що містять набір атрибутів різних типів, подібних до полів у таблиці. Кожен запис у структурі займає певне місце: батьківський запис може мати довільну кількість підлеглих записів, але кожен підлеглий запис пов’язаний лише з одним батьківським. Це створює чітку ієрархію, яка дозволяє ефективно відстежувати зв’язки між елементами.

Деталі моделі:

1. **Записи (вузли):**Кожен запис відповідає окремій сутності предметної області.

2. **Зв’язки (ребра):** Зв’язки між записами визначають відносини «батько-нащадок».

3. **Дерево:**Уся структура моделі нагадує дерево, у якому від одного кореневого вузла відходять гілки до підлеглих вузлів.

Приклад 1: Ієрархія компанії

Розглянемо базу даних, яка відображає організаційну структуру компанії.

- Кореневий запис: «Генеральний директор».
- Підлеглі записи: «Відділи» (наприклад, «Відділ маркетингу», «Відділ продажів»).
- Підлеглі до відділів записи: «Менеджери», «Фахівці».

У цій ієрархії кожен відділ може містити кілька менеджерів і фахівців, але кожен менеджер і фахівець належить лише до одного відділу.

Приклад 2: Каталог файлів

У файловій системі ієрархічна модель представлена у вигляді каталогу, де:

- Кореневий вузол — це головна папка.

- Кожна папка містить підпапки і файли.
- Файли належать тільки одній папці, але папка може мати багато файлів.

Це ілюструє типовий випадок ієрархічної організації, яку легко відобразити у вигляді дерева: корінь -> папка -> підпапка -> файли.

Переваги ієрархічної моделі:

- Чітка структура даних, зрозуміла навіть непрофесіоналам.
- Ефективний доступ до даних у випадках, коли запити відповідають ієрархічному представленню.
- Швидкість вибірки даних у добре оптимізованій ієрархії.

Недоліки:

- Складність у випадку зміни структури: додавання нових рівнів або зміна зв’язків потребує значної переробки моделі.
- Незручність роботи з даними, які не вписуються в ієрархію.
- Відсутність гнучкості у випадку складних багатовимірних відносин.

Ієрархічна модель надає ефективний спосіб організації даних, коли вони мають природну ієрархічну структуру.

Мережна модель даних — це модель, яка представляє структури даних у вигляді графа загального вигляду. У цій моделі вузлами графа можуть бути як прості (атомарні) елементи даних, так і записи зі складною структурою. Важливою особливістю мережної моделі є те, що кожен вузол (запис) може мати довільну кількість батьківських вузлів, що відрізняє її від ієрархічної моделі, де запис може мати лише одного батька.

Особливості мережної моделі:

1. **Гнучкість зв’язків:** Дані в мережній моделі пов’язані не лише вертикально (як у дереві), але й горизонтально. Це дозволяє створювати складні зв’язки між різними елементами.

2. **Підтримка складних структур:** Мережна модель може ефективно відображати складні відносини між даними, наприклад, багато-до-багатьох.

3. **Навігаційний доступ:** Для отримання даних у мережній моделі використовуються навігаційні запити, які вказують шляхи до потрібних вузлів через зв’язки графа.

Приклад 1: Система управління проектами

Уявімо базу даних для управління проектами, яка включає записи про співробітників, проекти, і відділи.

- Вузли графа:
  - Співробітники (містять дані про ім’я, прізвище, посаду).
  - Проекти (назва проекту, терміни виконання).
  - Відділи (назва відділу, керівник).
- Зв’язки:
  - Один співробітник може бути призначений на кілька проектів.
  - Один проект може належати кільком відділам.
  - Один відділ може мати кілька співробітників.

У цій моделі кожен співробітник і кожен проект можуть мати кількох «батьків» (наприклад, кілька відділів, які відповідають за проект) і одночасно кілька «нащадків» (наприклад, кілька співробітників, призначених на проект).

Приклад 2: База даних медичної клініки

- Вузли графа:
  - Лікарі (ім’я, спеціальність, номер ліцензії).
  - Пацієнти (ім’я, прізвище, номер медичної картки).
  - Прийоми (дата прийому, діагноз, призначення).
- Зв’язки:
  - Один лікар може приймати багато пацієнтів.
  - Один пацієнт може звертатися до кількох лікарів.
  - Один прийом може бути пов’язаний із конкретним лікарем та пацієнтом.

У мережній моделі кожен вузол (наприклад, пацієнт) може мати зв’язки з кількома батьками (різними лікарями) і нащадками (прийомами).

Переваги мережної моделі:

- Гнучкість у представленні складних відносин.
- Ефективний доступ до даних у випадках, коли запити передбачають складну навігацію між пов’язаними елементами.
- Можливість реалізації багато-до-багатьох зв’язків без дублювання даних.

Недоліки:

- Складність реалізації та підтримки, оскільки потрібно точно знати структуру зв’язків.
- Навігаційний доступ до даних може бути менш зручним, ніж декларативний підхід у реляційній моделі.

Мережна модель надає потужний інструмент для роботи зі складними зв’язками між даними, роблячи її ідеальною для предметних областей, де багато об’єктів взаємопов’язані.

Реляційна модель даних ґрунтується на математичному понятті відношення, яке в термінах баз даних реалізується у вигляді таблиць. У цій моделі дані організовані у вигляді відношень (таблиць), де кожен рядок представляє окремий запис (кортеж), а кожен стовпець відповідає атрибуту (полю) цього запису.

Основні компоненти реляційної моделі:

- **Відношення (таблиця):** Множина кортежів, організованих у вигляді рядків і стовпців. Наприклад, таблиця «Студенти» може містити такі стовпці: «ID студента», «Прізвище», «Ім’я», «Спеціальність».
- **Кортеж (рядок):** Окремий запис у таблиці. Кожен рядок містить конкретні значення атрибутів. Наприклад, один рядок у таблиці «Студенти» може містити: «001», «Іваненко», «Олексій», «Комп’ютерні науки».
- **Атрибут (стовпець):** Характеристика об’єкта, яку описує таблиця. Наприклад, атрибутом «Спеціальність» може бути значення «Комп’ютерні науки», «Економіка» тощо.
- **Ключі:**

- **Первинний ключ:** Унікальний ідентифікатор рядка в таблиці. Наприклад, «ID студента» може бути первинним ключем у таблиці «Студенти».
- **Зовнішній ключ:** Поле, яке вказує на первинний ключ іншої таблиці, щоб створити зв’язок між таблицями.

Приклад 1: База даних університету

- Таблиця «Студенти» має стовпці: ID студента, Ім’я, Прізвище, Спеціальність.
- Таблиця «Курси» має стовпці: ID курсу, Назва курсу, Кількість кредитів.
- Таблиця «Реєстрація» має стовпці: ID студента, ID курсу, Оцінка. Відношення між таблицями забезпечуються за допомогою ключів:
- ID студента в таблиці «Реєстрація» — зовнішній ключ, що посилається на таблицю «Студенти».
- ID курсу в таблиці «Реєстрація» — зовнішній ключ, що посилається на таблицю «Курси».

Переваги реляційної моделі:

- **Простота:** Таблична структура є природною і зрозумілою для більшості користувачів.
- **Гнучкість:** Відносно легко додавати нові таблиці, атрибути або записи.
- **Стандартизація:** Використання мови SQL для роботи з даними забезпечує сумісність між різними СУБД.
- **Цілісність даних:** Підтримка первинних і зовнішніх ключів допомагає зберігати зв’язки між даними коректними.
- **Можливість виконання складних запитів:** Реляційна модель дозволяє використовувати запити для з’єднання даних з різних таблиць, здійснення сортування, фільтрації та групування.

Недоліки:

- **Продуктивність:** При дуже великих обсягах даних і складних запитах реляційна модель може поступатися іншим моделям, наприклад, документно-орієнтованим або графовим.

- **Потреба в нормалізації:** Для ефективної роботи реляційної моделі потрібен ретельний процес проектування, що включає нормалізацію таблиць.

Таким чином, реляційна модель є основою для більшості традиційних систем управління базами даних, забезпечуючи зручність організації, доступу та маніпулювання даними.

Об'єктно-орієнтована модель даних ґрунтується на концепції об’єктів, які поєднують дані та операції над ними в єдину сутність. Об’єкт у цій моделі представляє конкретну сутність предметної області, що має свій стан і поведінку. Стан об’єкта описується його атрибутами (полями), які зберігають дані, тоді як поведінка визначається методами (операціями), які можна виконувати над цими даними.

Ключовою особливістю об’єктно-орієнтованої моделі є підтримка зв’язків між об’єктами, а також використання таких концепцій, як наслідування, інкапсуляція й поліморфізм. Це дозволяє організувати дані та їхню логіку більш природним і гнучким способом, особливо у складних системах.

Основні компоненти моделі:

- **Об’єкти:** Конкретні екземпляри типів (класів), які містять атрибути (дані) та методи (поведінку).
- **Класи:** Описують типи об’єктів. У класі визначаються всі атрибути й методи, які будуть у його екземплярах.
- **Зв’язки між об’єктами:** Об’єкти можуть бути пов’язані між собою через відношення, наприклад, «один-до-одного», «один-до-багатьох», «багато-до-багатьох».
- **Наслідування:** Класи можуть успадковувати атрибути й методи від інших класів, що дозволяє повторно використовувати код та організувати об’єкти у чіткі ієрархії.
- **Інкапсуляція:** Обмеження доступу до внутрішніх даних об’єкта та управління ним через визначені методи.

- **Поліморфізм:** Здатність викликати однакові методи для об’єктів різних типів, що спрощує взаємодію між ними.

Приклад : Онлайн-магазин

- **Клас «Товар»:**Атрибути: код товару, назва, ціна, кількість на складі. Методи: змінити кількість, змінити ціну.
- **Клас «Замовлення»:** Атрибути: номер замовлення, дата створення, список товарів.

Методи: додати товар у замовлення, видалити товар, отримати загальну вартість.

Зв’язки:

- Один об’єкт «Замовлення» може містити кілька об’єктів «Товар».
- Кожен «Товар» може бути присутнім у кількох об’єктах «Замовлення».

Переваги об'єктно-орієнтованої моделі:

- Природне відображення предметної області через об’єкти, що відповідають реальним сутностям.
- Зручна підтримка складних зв’язків між об’єктами.
- Можливість легко змінювати та розширювати модель за рахунок наслідування.
- Легкість в організації великих проектів завдяки інкапсуляції і розподілу відповідальності між об’єктами.

Недоліки:

- Складніша концепція для розуміння порівняно з реляційною моделлю.
- Більші витрати ресурсів на виконання деяких операцій, пов’язаних із пошуком і взаємодією об’єктів.
- Не завжди виправдане застосування у випадках, коли структура даних проста і не потребує багатого набору поведінкових операцій.

Таким чином, об’єктно-орієнтована модель забезпечує гнучкість, зручність і природність представлення предметної області, що особливо корисно для систем із великою кількістю зв’язків і складними об’єктами.

Архітектура баз даних визначає спосіб організації даних та управління ними, забезпечуючи узгодженість, цілісність і доступність інформації. Завдяки трирівневій архітектурі (зовнішній, концептуальний та внутрішній рівні) досягається незалежність даних, що дозволяє розробникам змінювати внутрішню структуру зберігання або логічну модель без впливу на кінцевих користувачів і прикладні програми.

Різноманітність моделей баз даних — реляційна, ієрархічна, мережна, об’єктно-орієнтована, документно-орієнтована, графова — забезпечує гнучкість у виборі підходу до зберігання та обробки даних. Кожна з них має свої переваги і недоліки, що визначають їх придатність для конкретних завдань. Наприклад, реляційна модель ідеальна для структурованих даних, тоді як графова модель оптимальна для систем із складними взаємозв’язками між об’єктами.

Вибір архітектури та моделі бази даних залежить від специфіки предметної області, обсягу даних, вимог до продуктивності, масштабованості та зручності адміністрування. У результаті правильно обрана архітектура і модель бази даних забезпечують ефективне управління інформацією, підтримку її цілісності, безпеки та доступності, що є ключовими аспектами сучасних інформаційних систем.

Програмні і мовні засоби баз даних відіграють ключову роль у забезпеченні ефективного зберігання, обробки та доступу до даних. Вони включають системи управління базами даних (СУБД), які надають основні механізми управління даними, утиліти для адміністрування, резервного копіювання та відновлення, а також транслятори, які забезпечують виконання запитів на відповідних мовах. Мови даних, такі як SQL, дозволяють формулювати запити до бази, визначати структури даних і забезпечувати доступ до них на рівні, зручному для розробників і кінцевих користувачів.

Еволюція мов даних і програмних інструментів зробила можливим створення все більш складних інформаційних систем, здатних інтегрувати дані з різних джерел, забезпечувати високу продуктивність і масштабованість, а

також адаптуватися до швидко змінюваних бізнес-вимог. Зростаюча популярність стандартизованих мов, таких як SQL, спрощує взаємодію з різними СУБД і дозволяє створювати портативні, легко підтримувані та розширювані застосування.

Отже, програмні та мовні засоби баз даних є фундаментом сучасних інформаційних систем. Вони не лише забезпечують ефективну роботу з даними, але й створюють основу для інтеграції, аналітики та автоматизації процесів у різних сферах діяльності, від бізнесу до наукових досліджень і державного управління.

2. Програмні і мовні засоби баз даних

Основу програмного забезпечення бази даних становить система управління базами даних (СУБД). У структурі СУБД можна виокремити ядро, яке відповідає за базові механізми роботи з базою даних, такі як зберігання, пошук і модифікація даних. Окрім цього, СУБД включає різноманітні додаткові компоненти: інструменти для тестування і налагодження, утиліти для відновлення бази даних, механізми збору та аналізу статистичних даних, а також засоби автоматизації адміністративних завдань.

Одним із ключових елементів СУБД є транслятори та компілятори, які забезпечують підтримку мов програмування і запитів, що використовуються для роботи з базою даних. Це дозволяє розробникам створювати ефективні застосування, які інтегруються з базою даних, виконують запити, аналізують результати і забезпечують зручний доступ до інформації для кінцевих користувачів.

Приклад:

- Ядро СУБД може забезпечувати такі базові функції, як обробка SQL-запитів, управління транзакціями, забезпечення цілісності даних.

- Додаткові компоненти можуть включати утиліти для створення резервних копій, відновлення після збоїв, аналізу продуктивності запитів або оптимізації індексів.
- Транслятори SQL перетворюють високорівневі запити в набори операцій, які ядро СУБД виконує на фізичному рівні.

Таким чином, СУБД є комплексним програмним середовищем, яке надає всі необхідні інструменти для ефективної роботи з базами даних, а також слугує основою для створення додатків, що спрощують доступ до даних та їхнє використання в різноманітних інформаційних системах.

Застосування — це програмне забезпечення, розроблене для вирішення певного набору завдань у конкретній предметній області або для надання універсальних інструментів, які можна використовувати у різних сферах. Основна мета застосування — спрощення та автоматизація процесів, пов’язаних із обробкою, аналізом і управлінням даними.

Таке програмне забезпечення може працювати з різними джерелами даних:

- **Фактографічними:** бази даних, таблиці, структуровані записи.
- **Документальними:** текстові файли, електронні документи, зображення.
- **Веб-ресурсами:** API, веб-сторінки, відкриті дані.

Крім того, застосування може мати різноманітну архітектуру залежно від потреб користувачів і технічних вимог:

- **Дволанкова архітектура (клієнт-сервер):** клієнтське програмне забезпечення безпосередньо звертається до сервера бази даних.
- **Триланкова архітектура:** додатковий середній шар (наприклад, сервер додатків) обробляє бізнес-логіку, полегшуючи роботу клієнтів і серверів баз даних.
- **Розподілена архітектура:** дані та обробка розподіляються між кількома вузлами (наприклад, хмарні обчислення або мікросервісні архітектури).

Приклад 1:

Застосування для управління складськими запасами може автоматизувати контроль за рівнем залишків, прийом товарів, облік партій та відвантаження.

Воно використовує базу даних товарів і постачальників (фактографічні джерела) та може мати дволанкову архітектуру — користувачі через інтерфейс взаємодіють із сервером баз даних.

Приклад 2:

Універсальне текстове редакторське програмне забезпечення дозволяє створювати, редагувати та форматувати документи. Воно працює з документальними джерелами даних, а також може інтегруватися з хмарними сховищами (веб-джерела) для спільної роботи над файлами. Архітектура може бути триланковою: клієнтський інтерфейс взаємодіє із сервером додатків, а той — із хмарними сховищами.

Переваги застосувань:

- Автоматизація рутинних процесів.
- Підвищення точності та швидкості обробки даних.
- Універсальність: можливість застосування в різних галузях.
- Гнучкість архітектури: можливість масштабування та інтеграції з іншими системами.

Таким чином, застосування — це потужний інструмент, який використовує різноманітні джерела даних та архітектурні підходи, щоб ефективно вирішувати завдання у конкретних предметних областях.

Застосування бази даних — це програмне забезпечення, яке взаємодіє з ресурсами певної системи управління базами даних (СУБД). Для отримання доступу до даних у базі використовується інтерфейс прикладного програмування (API) СУБД, що забезпечує виконання запитів, отримання результатів та виконання інших операцій.

Такі застосування можуть бути створені за допомогою стандартних мов програмування, наприклад, Pascal, C, Basic, або сучасних мов, таких як Python, Java, C#. Як правило, вони інтегрують SQL-оператори безпосередньо в код програми, що дозволяє розробникам ефективно виконувати запити, отримувати потрібні дані, модифікувати їх та зберігати назад у базу.

Приклад:

У системі обліку замовлень онлайн-магазину застосування бази даних може виконувати наступні функції:

- Створювати нові записи про замовлення в таблиці «Замовлення».
- Здійснювати пошук товарів за кодом або назвою у таблиці «Товари».
- Оновлювати статус замовлення в таблиці «Замовлення» (наприклад, з «очікується» на «відправлено»).
- Генерувати звіти про продажі за останній місяць, виконуючи складні SQL-запити з об'єднанням даних із таблиць «Замовлення» і «Товари».

Ці операції виконуються через API СУБД, яке дозволяє програмі встановлювати з'єднання з базою, відправляти запити, обробляти результати та забезпечувати цілісність і безпеку даних. Застосування бази даних може бути як консольним (працює через командний рядок), так і зручним графічним інтерфейсом, що значно спрощує взаємодію з користувачем.

Мова даних — це спеціалізована мова, яка використовується для опису, управління й обробки даних у межах тієї моделі даних, що підтримується певною системою управління базами даних (СУБД). Іншими словами, мова даних дозволяє визначати структури даних (наприклад, створювати таблиці або графи), виконувати операції над даними (вставка, видалення, оновлення, пошук), а також вирішувати інші задачі, наприклад, забезпечувати цілісність і безпеку даних.

Основні компоненти мови даних:

- **Мова визначення даних (Data Definition Language, DDL):**

Використовується для опису структури даних. З її допомогою створюються, змінюються та видаляються об'єкти бази даних (наприклад, таблиці, індекси, схеми).

- **Мова маніпулювання даними (Data Manipulation Language, DML):**

  Дозволяє виконувати основні операції над даними: додавання, видалення, оновлення та читання записів.

- **Мова управління даними (Data Control Language, DCL):**

Відповідає за керування доступом до даних та забезпечення безпеки бази даних. З її допомогою адміністратори бази встановлюють права на виконання різних операцій.

- **Мова запитів (Query Language):**

Дозволяє користувачам отримувати необхідну інформацію шляхом виконання запитів, які можуть включати сортування, групування, об’єднання таблиць та виконання агрегатних операцій.

Приклад:

Найпоширенішою мовою даних у реляційній моделі є SQL (Structured Query Language), яка включає всі зазначені компоненти. Наприклад:

- **DDL:**

```sql
CREATE TABLE Students (
  StudentID INT PRIMARY KEY,
  Name VARCHAR(100),
  BirthDate DATE
);
```

- **DML:**

```sql
INSERT INTO Students (StudentID, Name, BirthDate)
VALUES (1, 'John Doe', '2000-01-15');
```

- **DCL:**

```sql
GRANT SELECT, INSERT ON Students TO User1;
```

- **Запит:**

```sql
SELECT Name, BirthDate
FROM Students
WHERE BirthDate > '1999-12-31';
```

Мова даних є ключовим компонентом будь-якої СУБД, оскільки саме через неї відбувається взаємодія користувачів, розробників та адміністраторів із базою даних. Вона дозволяє описати структуру даних, виконувати операції над ними та забезпечувати цілісність, безпеку й ефективність роботи інформаційної системи.

Мова запитів — це спеціалізована мова, яка надає користувачам засоби для доступу до бази даних у декларативному стилі. Вона дозволяє описувати, які дані необхідно отримати або змінити, та встановлювати зв'язки між сутностями, не вказуючи при цьому точний алгоритм виконання цих операцій. Іншими словами, користувач формулює свій запит до даних у термінах того, що йому потрібно, а не як це зробити.

Мови запитів переважно працюють у інтерактивному режимі: користувач вводить запит, і система виконує його, повертаючи результати. Крім того, мови запитів можуть бути інтегровані у програмний код. Це дозволяє розробникам створювати застосування, які автоматично виконують запити до бази даних і обробляють результати у своїх функціях.

Особливості мов запитів:

- **Декларативний характер:**

Мова запитів описує, які дані потрібні, а не як їх отримати. Це полегшує написання запитів і робить їх зрозумілими навіть для користувачів, які не є розробниками.

- **Орієнтація на користувача:**

Мови запитів розроблені для того, щоб бути зручними для кінцевих користувачів, з мінімальними вимогами до знання технічних деталей роботи бази даних.

- **Підтримка інтерактивної роботи:**

Користувачі можуть виконувати запити безпосередньо через консоль або графічний інтерфейс, одразу отримуючи результати.

- **Можливість вбудовування:**

Багато мов запитів підтримують інтеграцію з популярними мовами програмування, такими як Java, Python, C#, що дозволяє включати запити безпосередньо в код застосувань.

Приклад:

Найпоширенішою мовою запитів для реляційних баз даних є SQL (Structured Query Language). Наприклад, простий запит на вибір імені та дати народження студентів із бази даних:

```sql
SELECT Name, BirthDate
FROM Students
WHERE BirthDate > '1999-12-31';
```

Цей запит декларативно описує, що потрібно: вибрати стовпці «Name» і «BirthDate» з таблиці «Students» для студентів, які народилися після 1999 року. Алгоритм виконання запиту (тобто, як саме система знайде ці дані) визначається СУБД, а не користувачем.

Мова маніпулювання даними (Data Manipulation Language, DML) — це компонент мов даних, призначений для виконання операцій над інформацією, що зберігається в базі даних. DML забезпечує доступ до даних та їхнє оновлення, надаючи команди для виконання таких дій, як вставка нових записів, видалення існуючих, змінення значень атрибутів і вибірка потрібних даних для подальшої обробки.

Основні операції DML:

Вставка (INSERT): Дозволяє додати новий запис у таблицю.
Приклад:

```sql
INSERT INTO Students (StudentID, Name, BirthDate)
VALUES (1, 'John Doe', '2000-01-15');
```

Оновлення (UPDATE):

Дозволяє змінювати існуючі дані в таблиці.

Приклад:

```sql
UPDATE Students
SET BirthDate = '1999-12-31'
WHERE StudentID = 1;
```

Видалення (DELETE):

Дозволяє видаляти записи з таблиці.

Приклад:

```sql
DELETE FROM Students
WHERE StudentID = 1;
```

Вибірка (SELECT):

Використовується для отримання даних із бази.

Приклад:

```sql
SELECT Name, BirthDate
FROM Students
WHERE BirthDate > '1999-12-31';
```

Особливості DML:

- **Операційні можливості:** DML реалізує ті операції, які підтримує модель даних. Наприклад, у реляційній моделі це вставка рядків у таблиці, у

графовій моделі — додавання вузлів і ребер, у документно-орієнтованій моделі — додавання документів.

- **Контрольовані дії:** Виконання DML-операцій відбувається відповідно до правил і обмежень, які визначені в базі даних, наприклад, цілісності даних або типів полів.
- **Масштабованість:** DML дозволяє виконувати операції як над окремими записами, так і над великими наборами даних.

DML є невід'ємною частиною роботи з базами даних. Вона забезпечує основні дії, які дозволяють отримувати потрібні дані, змінювати їх і керувати їхньою актуальністю, відповідно до операційних можливостей обраної моделі даних.

Мова SQL (Structured Query Language) — це найбільш розповсюджена мова, яка використовується для взаємодії з реляційними базами даних. Вона дозволяє описувати структури даних, виконувати запити, вставляти, оновлювати і видаляти записи, а також управляти правами доступу до даних. Мова SQL розроблена спеціально для доступу до баз даних, і, хоча вона не є повноцінною мовою програмування, її функціональності достатньо для виконання широкого спектру задач роботи з даними.

Міжнародні стандарти SQL:

- **SQL1 (SQL-86, SQL-89):** Перший офіційний стандарт мови SQL, який включав базові можливості для визначення і маніпулювання даними в реляційній моделі.
- **SQL2 (SQL-92):** Розширений стандарт, який включив додаткові типи даних, покращену підтримку підзапитів, нові операції об'єднання (JOIN) і більш виразні засоби маніпулювання даними.
- **SQL3 (SQL:1999 і далі):** Ще більше розширив можливості мови, додавши підтримку об'єктно-орієнтованих конструкцій, тригерів, збережених процедур і рекурсивних запитів.

Використання мови SQL:

- **Самостійне використання:**

SQL-запити можна виконувати безпосередньо в середовищах управління базами даних (таких як Oracle SQL*Plus, Microsoft SQL Server Management Studio, MySQL Workbench), що дозволяє адміністраторам бази і аналітикам працювати з даними без написання додаткового програмного коду.

- **Вбудований SQL:**

Мова SQL часто інтегрується у програми, написані на мовах загального призначення, таких як C, Java, Python. У цьому випадку SQL-команди вбудовуються в програмний код, і програмний застосунок виконує їх через спеціальні драйвери (наприклад, JDBC для Java або psycopg2 для Python). Це дозволяє створювати більш складні програми, які можуть динамічно формувати запити, обробляти результати і забезпечувати інтерактивну роботу з базами даних.

Приклад використання:

1. Самостійний SQL-запит:

```sql
SELECT Name, BirthDate
FROM Students
WHERE BirthDate > '2000-01-01';
```

2. Вбудований SQL у Python:

```python
import psycopg2
conn = psycopg2.connect("dbname=mydb user=myuser password=mypass")
cur = conn.cursor()
cur.execute("SELECT Name, BirthDate FROM Students WHERE BirthDate > %s", ('2000-01-01',))
results = cur.fetchall()
for row in results:
    print(row)
cur.close()
conn.close()
```

Мова SQL є основним інструментом для роботи з реляційними базами даних, забезпечуючи простоту, стандартність і універсальність. Завдяки

міжнародним стандартам SQL1, SQL2 і SQL3 її можливості постійно розширюються, що дозволяє вирішувати все складніші задачі. SQL може використовуватися як самостійно, так і у складі програм, що робить її незамінним інструментом для розробників, адміністраторів баз даних і аналітиків.

3. Архітектура інформаційної системи

Ефективність роботи інформаційної системи значною мірою визначається її архітектурою, тобто способом організації її компонентів. Архітектурні рішення впливають на продуктивність, масштабованість, надійність і зручність підтримки системи. Функціональні компоненти інформаційної системи можуть бути розміщені як на одному комп'ютері, так і на декількох вузлах мережі. Якщо ж інформаційна система працює на одному комп'ютері, то можливі різні сценарії використання програмного забезпечення.

Варіанти розміщення компонентів у рамках одного комп'ютера:

1. **Застосування та СУБД:**

У цьому випадку користувач взаємодіє із системою безпосередньо через інтерфейс СУБД або через прикладну програму, яка працює разом із базою даних. Такий підхід спрощує адміністрування і знижує затримки в обміні даними.

2. **Застосування та ядро СУБД:**

У цій конфігурації прикладна програма звертається не до повнофункціональної СУБД, а до її ядра, яке забезпечує базові операції роботи з даними. Цей підхід дозволяє мінімізувати використання ресурсів і підвищити продуктивність, якщо система не потребує розширених можливостей СУБД.

3. **Незалежне застосування:**

Застосування може використовувати власні засоби обробки даних без інтеграції з ядром СУБД чи повнофункціональною базою даних. У такому випадку дані можуть зберігатися у файлах, а всі операції над ними виконуються засобами

самої програми. Такий підхід часто використовується для невеликих програм, де створення окремої бази даних не є виправданим.

Додаткові аспекти:

- Вибір конкретного сценарію розгортання залежить від характеру задач, які вирішує інформаційна система, обсягів даних, частоти запитів і вимог до продуктивності.
- Використання СУБД забезпечує централізоване зберігання даних, легше масштабування та можливість спільної роботи кількох користувачів.
- Робота без СУБД (наприклад, з використанням файлової системи) може бути доцільною у невеликих проектах, але в міру зростання обсягів даних і складності запитів така архітектура швидко вичерпує свої можливості.

Правильний вибір архітектури дозволяє досягти оптимального балансу між продуктивністю, простотою впровадження та зручністю обслуговування інформаційної системи.

У першому випадку взаємодія користувача і СУБД виконується або напряму через користувацький інтерфейс СУБД, або за допомогою застосування (рис. 6.3.).

Рис. 6. 3. Використання застосування і СУБД

У другому випадку взаємодія користувача і СУБД виконується за допомогою застосування (рис. 6. 4). Такий підхід дозволяє підвищити швидкість роботи застосування, зменшити об'єм необхідної пам'яті.

Рис. 6.4. Використання застосування і ядра СУБД

Розробка незалежних застосувань дозволяє взаємодіяти з базою даних без використання СУБД (рис. 6.5). Це сприяє підвищенню швидкості роботи програми та зниженню вимог до пам'яті. Однак подібний підхід має свої недоліки: він збільшує складність внесення змін у програму і позбавляє її переваг стандартних інструментів СУБД для обслуговування бази даних.

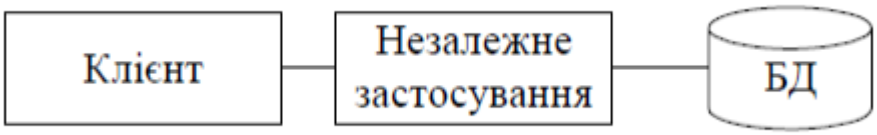

Рис. 6. 5. Використання незалежного застосування

Коли комп'ютери інтегруються в мережу, з'являється можливість розподіляти як програми, що працюють із єдиною базою даних, так і саму базу даних між кількома вузлами мережі. Найпоширенішим підходом є схема, у якій кожен користувач має власну локальну базу даних (КБД) і звертається до центральної серверної бази даних (СБД) для доступу до інформації, що використовується спільно всіма користувачами (рис.6.6).

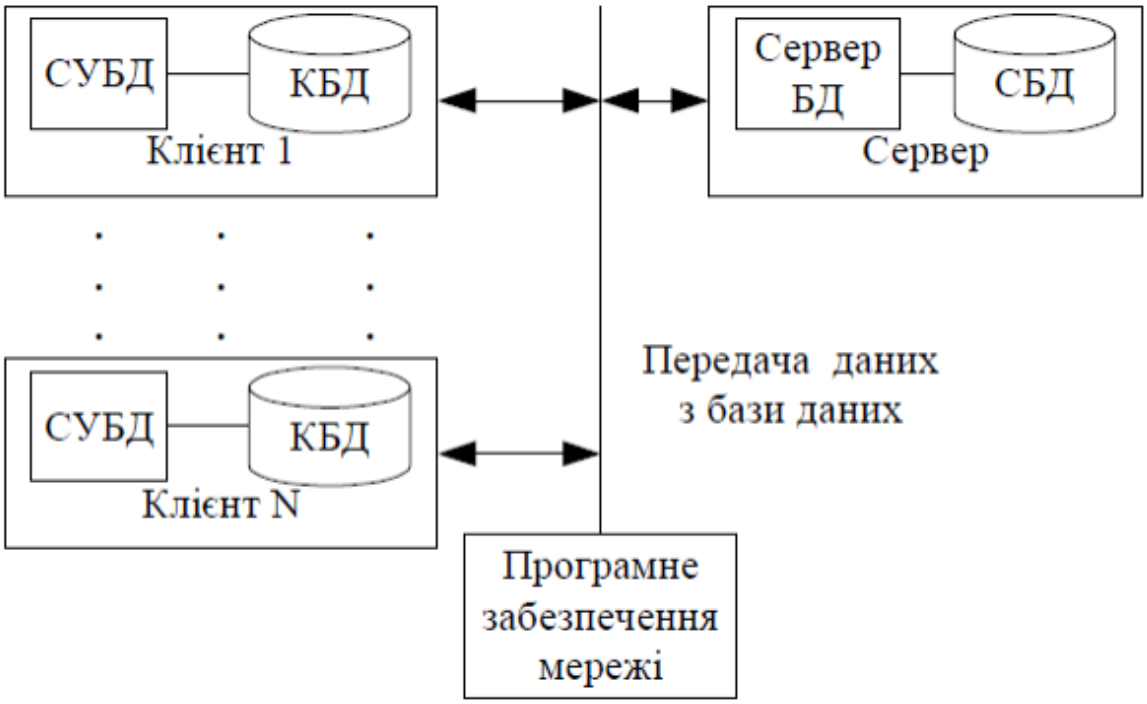

Рис. 6.6. Використання сервера БД

Сервером вважається комп'ютер або програмне забезпечення, які здійснюють управління певними ресурсами. Клієнтом, у свою чергу, називають

комп’ютер чи програму, які ці ресурси використовують. Такий спосіб організації дозволяє об’єднати переваги централізованого зберігання даних із можливістю їхньої індивідуальної обробки кожним користувачем.

Архітектура інформаційної системи визначає загальну структуру її компонентів і спосіб їхньої взаємодії для досягнення поставлених функціональних і технічних цілей. Вона є фундаментом, на якому будується вся система, і включає організацію апаратного забезпечення, програмних компонентів, мережевої інфраструктури, а також методів обробки, зберігання і передачі даних.

Основні рівні архітектури інформаційної системи:

1. **Фізичний рівень (апаратне забезпечення):**
    - Сервери, робочі станції, мережеве обладнання, системи зберігання даних.
    - Забезпечує базову платформу для роботи програмного забезпечення.
2. **Логічний рівень (програмне забезпечення):**
    - Операційні системи, системи управління базами даних (СУБД), сервери додатків, прикладне програмне забезпечення.
    - Відповідає за обробку даних, виконання бізнес-логіки, взаємодію користувача із системою.
3. **Рівень даних:**
    - Структури зберігання даних (таблиці, сховища документів, графи), метадані.
    - Описує, як дані організовані, доступні і керовані в системі.
4. **Рівень інтерфейсу (представлення):**
    - Інтерфейси користувача (графічний, веб-інтерфейс, мобільні додатки).
    - Забезпечує зручний доступ до функцій і даних системи для кінцевих користувачів.
5. **Мережевий рівень:**

- Протоколи, маршрутизація, безпека передачі даних.
- Відповідає за взаємодію між компонентами системи, незалежно від їх фізичного розташування.

Архітектурні підходи:

- **Клієнт-сервер:** Користувачі взаємодіють із сервером, який виконує основні обчислення і обробку даних.
- **Триланкова:** Доданий проміжний рівень (сервер додатків), що виконує бізнес-логіку, забезпечуючи гнучкість і масштабованість.
- **Мікросервіси:** Система складається з невеликих сервісів, які працюють незалежно один від одного і можуть бути масштабовані окремо.
- **Хмарна архітектура:** Використання хмарних сервісів для обчислень, зберігання даних і масштабування ресурсів за потребою.

Архітектура інформаційної системи є критично важливою для її ефективності, продуктивності і надійності. Від вибору архітектури залежить те, наскільки легко систему можна розширювати, модернізувати і підтримувати. Саме тому архітектурні рішення мають враховувати не лише поточні потреби, а й перспективи зростання та змін у майбутньому.

Контрольні питання

1. Представити та пояснити архітектуру бази даних.
2. Які головні рівні абстрагування в базах даних?
3. Назвати основні моделі даних і дати їх характеристику.
4. В чому різниця між реляційною і об'єктно-реляційною моделями?
5. Які переваги і недоліки об'єктно-орієнтованої моделі?
6. Дати класифікацію програмних і мовних засобів по роботі з БД.
7. Вказати основні архітектурні рішення інформаційної системи з базами даних.
8. Що таке схема БД? Перелічити її компоненти.
9. Що розуміється під незалежністю даних?

Тема 7.
ІНТЕРНЕТ І МЕРЕЖЕВІ ТЕХНОЛОГІЇ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Мета: полягає в ознайомленні здобувачів із сучасними мережевими технологіями та їхнім використанням у сфері інформаційних систем. Лекція розкриває основні принципи функціонування Інтернету, його архітектуру, ключові протоколи та служби, а також роль мережевих технологій у забезпеченні ефективного управління даними, доступу до ресурсів і взаємодії між користувачами. Окрема увага приділяється тенденціям розвитку мережевих технологій та їхньому впливу на інформаційні системи майбутнього. Важливим завданням є забезпечення розуміння можливостей інтеграції Інтернету і сучасних мережевих інструментів у бізнес-середовище, науку й освіту.

План лекції

1. Архітектура Інтернету.
2. Сучасні сервіси Інтернету.
3. Розвиток мережевих технологій.
4. Інтеграція Інтернету в різні галузі.

1. Архітектура Інтернету

Інтернет також забезпечує архівування даних, дозволяючи зберігати інформацію за датою, автором, тематикою чи жанром. Крім того, завдяки автоматизованим системам перекладу, інформація стає позамовною, тобто доступною для різних мовних аудиторій, що сприяє глобальній доступності контенту. Користувачі в мережі не лише споживають інформацію, але й активно створюють її та поширюють, стаючи одночасно читачами, авторами та комунікаторами.

Однією з найгостріших проблем є надмірність інформації, яка ускладнює пошук дійсно корисних і потрібних даних. Цей феномен, відомий як

«інформаційний шум» або «інформаційний шок», може призводити до зниження якості сприйняття і використання інформації. У свою чергу, мобільність і висока швидкість поширення даних дозволяють інформації миттєво ставати доступною у всьому світі, що змінює традиційні межі сприйняття часу та простору, створюючи нові форми взаємодії між читачами та авторами.

Мережеві інформаційні потоки також відрізняються позатериторіальністю, оскільки інформація не підпорядковується суто національним регулюванням і законодавству. Це ускладнює контроль за дотриманням прав та правил у глобальному масштабі. Крім того, через динамічний характер інформаційних потоків виникають труднощі з визначенням первісного джерела даних, що може призвести до втрати автентичності та достовірності.

Ще одним викликом є недовговічність інформації. Дані можуть швидко втратити актуальність через переміщення або закриття ресурсів, хакерські атаки, а також інформаційне перенасичення. Наприклад, на однакові запити в різний час можна отримати різні результати через постійне зростання обсягу даних. Це зростання в геометричній прогресії ставить питання про обмеження зберігання інформації та ефективного її структурування.

Зрештою, інформаційні потоки в мережі мають гетерархічну природу. Це означає, що поширення даних у мережі відбувається здебільшого горизонтально, без чіткої ієрархії чи централізованого управління. Такий підхід сприяє більш відкритому й децентралізованому обміну інформацією, однак водночас створює нові виклики, пов’язані з управлінням, перевіркою та зберіганням даних.

Мережева комунікація зберігає традиційні складові будь-якого комунікативного процесу:

- **Адресант (автор)** – особа або організація, що створює повідомлення.
- **Адресат (споживач)** – кінцевий отримувач або аудиторія, для яких призначене повідомлення.

- **Повідомлення** – зміст, інформація, яку передає адресант адресатові.
- **Канал комунікації** – шлях або засіб, через який передається інформація (наприклад, електронна пошта, соціальні мережі, форуми).
- **Комунікативний шум** – фактори, які заважають або спотворюють передачу повідомлення (помилки у передачі даних, інформаційне перевантаження, технічні несправності).
- **Фільтри** – внутрішні чи зовнішні обмеження (цензура, корпоративна політика, алгоритми модерації), що регулюють доступність інформації.
- **Зворотний зв'язок** – реакція аудиторії на отримане повідомлення (коментарі, лайки, відповіді).

Переваги мережевої комунікації

Завдяки сучасним інформаційним технологіям мережеве спілкування стало глобальним явищем. Основні переваги включають:

- **Креативність і гнучкість**: користувачі можуть створювати нові форми контенту, адаптувати способи подачі інформації та оперативно реагувати на зміни.
- **Мультиплікативний ефект**: повідомлення в мережі мають потенціал для швидкого поширення і впливу на великі аудиторії.
- **Транснаціональний характер**: інформація більше не обмежується географічними або культурними бар'єрами, що дозволяє мережам охоплювати всі основні сфери людської діяльності – від політики до фінансів.
- **Різнорівневе функціонування**: мережеві комунікації працюють на індивідуальному, організаційному, національному та міжнародному рівнях.

Недоліки мережевої комунікації

Поряд із перевагами, мережеве спілкування має і значні недоліки:

- **Інтернет-залежність**: все частіше відзначається соціально-психологічний вплив, при якому люди надмірно залежать від онлайн-комунікацій, що

може негативно впливати на психічне здоров'я, соціальну активність і реальні міжособистісні стосунки.

- **Девіантна поведінка**: у цифровому середовищі виникають нові форми порушення соціальних норм, такі як хакерство, кібербулінг, поширення дезінформації.
- **Зростання інформаційного шуму**: величезна кількість контенту, що створюється в мережі, ускладнює пошук достовірної та якісної інформації.

Отже, мережева комунікація відкриває нові можливості для взаємодії, але водночас вимагає відповідального підходу до використання цифрових платформ, адекватного управління інформаційними потоками і вирішення нових соціальних та етичних викликів.

Архітектура Інтернету — це багаторівнева структура, яка визначає принципи взаємодії між його основними компонентами. Вона забезпечує з'єднання мільярдів пристроїв, реалізуючи глобальну мережу передачі даних, що працює за загальноприйнятими протоколами та стандартами. Архітектура Інтернету складається з кількох ключових рівнів, кожен з яких виконує конкретну роль у забезпеченні комунікації.

Основні рівні архітектури Інтернету:

1. **Фізичний рівень:**
    - Включає канали зв'язку (оптоволокно, кабелі, бездротові технології), маршрутизатори, комутатори та інші апаратні засоби.
    - Забезпечує передачу бітів даних між вузлами.
2. **Мережевий рівень:**
    - Реалізується за допомогою IP-протоколу (IPv4 або IPv6).
    - Відповідає за адресацію та маршрутизацію даних, забезпечуючи їхнє транспортування від джерела до призначення.
3. **Транспортний рівень:**
    - Найпоширенішими протоколами цього рівня є TCP (забезпечує надійність передачі) і UDP (більш швидкий, але менш надійний).

- Відповідає за контроль цілісності даних, встановлення з'єднання між пристроями, розподіл великих обсягів даних на пакети.

4. **Прикладний рівень:**
   - Охоплює протоколи, що безпосередньо взаємодіють із користувачем або програмами, такі як HTTP/HTTPS (використовуються для перегляду веб-сторінок), FTP (передача файлів), SMTP (відправлення електронної пошти), DNS (перетворення доменних імен на IP-адреси).
   - Забезпечує сервіси та інтерфейси, через які користувачі взаємодіють із мережею.

Особливості архітектури Інтернету:

- **Децентралізованість:** Інтернет не має одного центрального вузла або органу управління, що робить його стійким до збоїв і гнучким у масштабуванні.
- **Модульність:** Кожен рівень архітектури виконує конкретні функції, що спрощує впровадження нових технологій і протоколів.
- **Гнучкість:** Завдяки використанню стандартів і протоколів, пристрої від різних виробників можуть взаємодіяти між собою.
- **Масштабованість:** Архітектура дозволяє легко додавати нові пристрої, мережі й сервіси, не порушуючи роботу існуючої інфраструктури.

Архітектура Інтернету також включає концепції безпеки, розширення та управління. Зростаюча кількість користувачів і пристроїв вимагає вдосконалення протоколів і впровадження нових стандартів, які забезпечують захист даних і стабільність мережі. Крім того, такі технології, як віртуалізація, хмарні обчислення і мережеві функції, що визначаються програмно (SDN), інтегруються в традиційну архітектуру Інтернету, роблячи її більш гнучкою і ефективною.

Одним із ключових аспектів є перехід на протокол IPv6. Ця зміна дозволяє значно збільшити кількість доступних адрес, що критично важливо для Інтернету речей (IoT) і інших сучасних мережевих технологій. IPv6 також

поліпшує підтримку мобільних пристроїв, маршрутизацію та безпеку, що робить Інтернет готовим до масштабного росту й нових викликів.

Ще одним важливим напрямком є розвиток контент-мереж (CDN), які забезпечують швидшу доставку даних користувачам за рахунок розподілу ресурсів ближче до кінцевих точок. Це покращує продуктивність сервісів потокового відео, хмарних додатків і веб-сайтів із високим рівнем трафіку. Інтеграція CDN у загальну архітектуру Інтернету забезпечує зниження затримок і підвищення якості роботи користувачів у будь-якому куточку світу.

Таким чином, архітектура Інтернету продовжує еволюціонувати, зберігаючи свою основну функцію — забезпечення глобальної комунікації. Вона адаптується до нових технологічних трендів, підтримує інновації та залишається фундаментом цифрової інфраструктури, яка об’єднує людей, пристрої й дані в єдину мережеву екосистему. Її багаторівневий характер дозволяє легко адаптуватися до нових технологічних вимог і підтримувати стабільну, ефективну роботу Інтернету навіть при постійному зростанні обсягу трафіку та кількості підключених пристроїв.

2. Сучасні сервіси Інтернету

Сучасні інтернет-сервіси — це широкий спектр платформ і технологій, які істотно змінюють наше повсякденне життя. Вони створюють нові можливості для комунікації, освіти, бізнесу, творчості, розваг і доступу до інформації. Завдяки інтернету ми можемо миттєво спілкуватися з людьми на іншому кінці світу, проводити відеоконференції, навчатися через онлайн-курси, замовляти товари та послуги, слухати музику, дивитися фільми або займатися творчістю в хмарних середовищах.

Приклади:

- **Комунікаційні сервіси:**

Платформи на зразок Zoom, Microsoft Teams або Google Meet дозволяють проводити відеозустрічі, навчальні лекції та корпоративні наради. Соціальні мережі, такі як Facebook, Instagram, Twitter і TikTok, дають змогу обмінюватися інформацією, публікувати контент і залишатися на зв'язку з друзями та аудиторією.

- **Навчальні платформи:**

Курси на Coursera, Udemy або Khan Academy відкривають доступ до знань з будь-якої точки світу. Університети пропонують онлайн-програми, що дають можливість отримати диплом без відвідування традиційних лекцій.

- **Електронна комерція:**

Сервіси на зразок Amazon, eBay, AliExpress або місцевих інтернет-магазинів дозволяють купувати товари, не виходячи з дому. Інтернет-банкінг і платіжні системи, такі як PayPal, Apple Pay чи Google Pay, спрощують фінансові транзакції.

- **Розваги:**

Стримінгові платформи, такі як Netflix, Spotify, YouTube і Twitch, пропонують величезний вибір фільмів, серіалів, музики та прямих трансляцій. Геймерські сервіси, на зразок Steam або Xbox Live, дають доступ до багатокористувацьких ігор, створюючи спільноти гравців з усього світу.

- **Творчі інструменти:**

Хмарні сервіси, такі як Adobe Creative Cloud або Canva, дозволяють створювати та редагувати графіку, відео і документи в реальному часі. Музичні продакшн-платформи, такі як Soundtrap або BandLab, спрощують створення і розповсюдження музики.

- **Хмарні технології:**

Завдяки Google Drive, Dropbox, OneDrive і іншим подібним сервісам можна зберігати дані в хмарі, ділитися файлами, а також спільно редагувати документи в режимі реального часу.

Сучасні інтернет-сервіси стали невід'ємною частиною нашого життя, значно спрощуючи взаємодію з інформацією та іншими людьми. Вони не тільки

підвищують ефективність у роботі й навчанні, а й створюють безліч нових форматів дозвілля, що робить наше повсякденне існування комфортнішим та насиченішим.

Комунікаційні сервіси — це інструменти, створені для обміну інформацією між користувачами, підтримки зв'язку та ефективної взаємодії. Вони включають широкий спектр платформ, які охоплюють як приватне, так і професійне спілкування.

До основних видів комунікаційних сервісів належать:

- **Месенджери:**

Skype, Viber, WhatsApp та подібні програми забезпечують текстову, голосову та відеокомунікацію в режимі реального часу. Вони ідеально підходять для особистого спілкування, ділових конференцій або обговорення проектів у командах.

- **Соціальні мережі:**

Facebook, Instagram, LinkedIn надають платформу для створення та обміну контентом, налагодження професійних контактів, ведення бізнес-сторінок і побудови бренду. Вони також дозволяють обговорювати актуальні новини, ділитися досвідом та взаємодіяти з різноманітними аудиторіями.

- **Відеоконференцзв'язок:**

Сервіси на зразок Zoom, Microsoft Teams, Google Meet забезпечують проведення віртуальних зустрічей, вебінарів і лекцій. Це незамінні інструменти для віддаленої роботи, навчання та співпраці між командами, що перебувають у різних куточках світу.

- **Корпоративні комунікаційні платформи:**

Slack, Microsoft Teams та інші інтегровані рішення пропонують зручні канали для спілкування всередині організацій, обміну файлами, постановки задач і обговорення проектів. Вони допомагають створити єдиний інформаційний простір для всіх співробітників.

- **Платформи для аудіо- та відеострімінгу:**

Clubhouse, Discord, Twitch дозволяють обмінюватися інформацією у форматі аудіо- або відеотрансляцій, спільно обговорювати теми, проводити тематичні заходи чи навіть організовувати цілі онлайн-конференції.

Комунікаційні сервіси стають усе більш інтерактивними, інтуїтивно зрозумілими та зручними для користувачів. Багато з них інтегруються з іншими інструментами — календарями, системами управління задачами, хмарними сховищами, що робить їх універсальними платформами для ефективної роботи та взаємодії.

Інформаційно-пошукові сервіси — це онлайн-інструменти, які дозволяють користувачам швидко та ефективно знаходити потрібну інформацію в інтернеті. Вони виконують функцію орієнтирів у великому інформаційному просторі, допомагаючи отримувати доступ до даних, веб-сторінок, документів і мультимедійного контенту.

Основні види інформаційно-пошукових сервісів:

- **Пошукові системи:**

Google, Bing, Yahoo! та інші — це універсальні платформи, що дозволяють знаходити веб-сторінки, зображення, відео, новини та інші ресурси за ключовими словами. Вони використовують алгоритми ранжування, які враховують релевантність і авторитетність джерел, щоб запропонувати користувачам найкращі відповіді на їхні запити.

- **Онлайн-бібліотеки та енциклопедії:**

Wikipedia, Google Books, Britannica Online — це ресурси, які надають доступ до широкого спектра знань. Вони охоплюють як базову інформацію з різних галузей, так і повнотекстові версії книг, статей, наукових праць і історичних документів.

- **Спеціалізовані інформаційно-пошукові платформи:**

PubMed, arXiv, IEEE Xplore — сервіси, орієнтовані на пошук наукових статей, технічних звітів, патентів і дослідницьких матеріалів. Вони допомагають професіоналам, науковцям і студентам отримувати доступ до спеціалізованої інформації в певних галузях.

- **Новинні агрегатори та медіа-платформи:**

Google News, Flipboard, Feedly — сервіси, що збирають і структурують новини з різних джерел. Вони дозволяють користувачам бути в курсі останніх подій, стежити за тематичними трендами та створювати персоналізовані стрічки новин.

- **Відео- та мультимедійні пошукові сервіси:**

YouTube, Vimeo, Dailymotion — сервіси для пошуку відеоконтенту, що охоплює освітні лекції, розважальні шоу, музичні кліпи та інструкційні відео. Вони також пропонують рекомендації на основі уподобань користувача.

Сучасні інформаційно-пошукові сервіси постійно вдосконалюються, інтегруючи штучний інтелект і машинне навчання. Це дозволяє їм забезпечувати точніші результати пошуку, прогнозувати інтереси користувачів і навіть автоматично підбирати контент. Такі сервіси стають невід'ємним інструментом як для повсякденного використання, так і для професійної діяльності, навчання та наукових досліджень.

Електронна комерція — це широкий спектр онлайн-сервісів і платформ, які дозволяють користувачам здійснювати купівлю, продаж, обмін товарів і послуг через Інтернет. Вони відкривають доступ до глобальних ринків, забезпечуючи зручний та швидкий спосіб здійснення торгових операцій, незалежно від фізичного місця знаходження покупців і продавців.

Основні види платформ електронної комерції:

- **Інтернет-магазини:**

Rozetka, Amazon, Aliexpress, Wildberries — це онлайн-платформи, які надають користувачам можливість вибирати, замовляти та оплачувати товари. Вони зазвичай мають зручний інтерфейс, каталог продукції, відгуки покупців та інструменти для відстеження замовлень.

- **Маркетплейси:**

eBay, OLX, Etsy, Prom.ua — платформи, які об'єднують різних продавців і дозволяють їм пропонувати свої товари. Вони надають спільну інфраструктуру,

через яку покупці можуть знайти великий вибір продукції, а продавці — вийти на ширшу аудиторію.

- **Спеціалізовані торгові платформи:**

На кшталт Airbnb для оренди житла, Uber Eats для доставки їжі або AutoScout24 для продажу автомобілів. Такі сервіси орієнтовані на конкретні ринки, забезпечуючи зручний доступ до спеціалізованих товарів або послуг.

- **Соціальна комерція:**

Facebook Marketplace, Instagram Shopping — це торгові сервіси, інтегровані в соціальні мережі. Вони дозволяють продавцям взаємодіяти з покупцями через соцмережі, розміщувати товарні оголошення, приймати замовлення та отримувати миттєвий зворотний зв'язок.

Електронна комерція активно розвивається завдяки інтеграції нових технологій, таких як мобільні платежі, криптовалюта, автоматизована обробка замовлень і системи рекомендацій на основі штучного інтелекту. Вона також підтримує багатоканальність, коли покупці можуть починати процес покупки на одному пристрої і завершувати на іншому. У результаті, електронна комерція перетворилася на ключовий елемент сучасної економіки, що забезпечує доступність товарів і послуг для мільйонів людей по всьому світу.

Сервіси для зберігання та обміну файлами — це платформи, що дозволяють зберігати дані в хмарі, а також легко ділитися ними з іншими користувачами. Вони надають зручний доступ до документів, мультимедіа та інших файлів незалежно від місця розташування, часу чи пристрою, який використовується.

Основні категорії сервісів:

- **Хмарні сховища:**

Платформи на зразок Google Drive, Dropbox, OneDrive, Mega пропонують зручний спосіб зберігання файлів онлайн. Вони забезпечують резервне копіювання, синхронізацію між різними пристроями, можливість створення спільного доступу та обмеження прав на редагування чи перегляд.

- **Сервіси для обміну великими файлами:**

Fex.net, WeTransfer, DropMeFiles — це сервіси, які спеціалізуються на пересиланні великих обсягів даних, таких як відео, графічні матеріали або набори даних. Вони забезпечують швидке завантаження файлів на сервер, створення посилання для скачування і обмеження доступу за часом або кількістю завантажень.

- **Відеохостинги:**

YouTube, Vimeo, TikTok — це платформи для зберігання, публікації та перегляду відеоконтенту. Вони дозволяють користувачам завантажувати відео, ділитися ними через посилання, створювати плейлисти та отримувати відгуки від аудиторії. Багато з них також пропонують можливості монетизації контенту для авторів.

Завдяки постійному вдосконаленню хмарних технологій і системи глобальних дата-центрів ці сервіси стали невід'ємною частиною цифрового життя. Вони інтегруються з офісними додатками, засобами комунікації (такими як Slack, Microsoft Teams) і навіть з мобільними операційними системами, що значно підвищує продуктивність роботи і полегшує обмін інформацією. Більшість сучасних хмарних сховищ також пропонують інструменти для спільної роботи, дозволяючи кільком користувачам одночасно редагувати документи, створювати презентації або вносити правки у проекти в реальному часі.

Таким чином, сервіси для зберігання та обміну файлами забезпечують не лише зручне розташування даних, а й гнучкість, безпеку та простоту у співпраці, що робить їх незамінними для роботи, навчання та особистого використання.

Хмарні обчислювальні технології — це сервіси, які надають обчислювальні ресурси та програмні можливості через Інтернет, використовуючи розподілені дата-центри. Вони дають змогу користувачам виконувати складні обчислення, запускати додатки, обробляти великі обсяги даних та навіть грати в сучасні відеоігри без необхідності мати потужне локальне обладнання.

Основні напрямки застосування:

- **Хмарний геймінг:**

Сервіси на кшталт GeForce Now, Xbox Cloud Gaming, Google Stadia дозволяють грати у високоякісні ігри на будь-якому пристрої, підключеному до Інтернету. Гравець надсилає команди, які обробляються на потужних серверах, а відеопотік із грою передається назад. Такий підхід знімає потребу у високопродуктивному геймерському ПК чи консолі, забезпечуючи доступ до ігор на смартфонах, планшетах або недорогих ноутбуках.

- **Хмарні обчислення:**

Платформи, як-от Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, пропонують інфраструктуру, яка дозволяє розробникам створювати, тестувати та розгортати програми у хмарному середовищі. Ці платформи забезпечують доступ до віртуальних серверів, баз даних, контейнерів, інструментів машинного навчання та інших сервісів. Вони дають змогу масштабувати ресурси відповідно до навантаження, оптимізувати витрати і швидко реагувати на змінні потреби бізнесу.

Переваги хмарних технологій:

- **Гнучкість:** користувачі можуть легко збільшувати або зменшувати обчислювальні потужності залежно від потреб.
- **Масштабованість:** ресурси доступні практично необмежено, що дозволяє підтримувати проекти будь-якого масштабу.
- **Доступність:** сервіси можуть використовуватися на будь-якому пристрої з доступом до Інтернету.
- **Зниження витрат:** замість інвестицій у власну інфраструктуру компанії платять лише за ті ресурси, які вони фактично використовують.

Сьогодні хмарні обчислення охоплюють не лише розваги та розробку додатків, а й такі сфери, як аналітика даних, штучний інтелект, навчання, медіа-продукція. Наприклад, відеоредактори можуть виконувати рендеринг на потужних хмарних серверах, а науковці — обробляти великі обсяги даних для

досліджень. Усе це робить хмарні технології критично важливим елементом сучасної IT-екосистеми.

Платіжні онлайн-сервіси — це платформи, які спрощують здійснення фінансових операцій через Інтернет. Вони дозволяють користувачам швидко та безпечно оплачувати товари, послуги, переказувати кошти, контролювати банківські рахунки та виконувати інші фінансові завдання.

Основні категорії платіжних онлайн-сервісів:

- **Фінансові сервіси:**

PayPal, Google Pay (GPay), Apple Pay — це глобальні платформи, які дозволяють оплачувати покупки в інтернет-магазинах, здійснювати перекази коштів між користувачами та інтегрувати способи оплати в мобільні додатки. Завдяки високому рівню безпеки та зручному інтерфейсу вони широко використовуються для швидких фінансових операцій.

- **Банківські онлайн-сервіси:**

Додатки і веб-сайти банків, такі як Privat24, Monobank, Oschadbank Online, дають змогу клієнтам виконувати весь спектр банківських операцій: перевіряти стан рахунків, переказувати кошти, поповнювати мобільні телефони, відкривати депозити, оформлювати кредити, оплачувати послуги та отримувати електронні виписки.

- **Комунальні та рахункові сервіси:**

MyGas, KOEK, City24 — спеціалізовані платформи для оплати комунальних послуг, електроенергії, водопостачання, газу та Інтернету. Вони дозволяють користувачам контролювати витрати, отримувати нагадування про неоплачені рахунки та зберігати історію оплат.

Багато платіжних сервісів інтегруються з іншими онлайн-платформами: інтернет-магазинами, мобільними додатками, сервісами доставки, маркетплейсами. Це дозволяє користувачам оплачувати послуги буквально в кілька кліків. Крім того, сучасні платіжні системи пропонують високий рівень захисту: використання двофакторної аутентифікації, шифрування даних, підтримка токенізації карт, що підвищує безпеку фінансових операцій.

Таким чином, платіжні онлайн-сервіси не тільки роблять фінансові транзакції швидшими і зручнішими, але й сприяють розвитку електронної комерції, покращуючи доступ до товарів і послуг як у межах однієї країни, так і на міжнародному рівні.

Освітні сервіси — це платформи й інструменти, які дозволяють організовувати та проводити навчання, освітні заходи та професійний розвиток в онлайновому середовищі. Вони забезпечують доступ до знань, інтерактивні заняття, спільну роботу та дистанційне спілкування між викладачами і студентами, незалежно від їхнього фізичного розташування.

Основні типи освітніх сервісів:

- **Інструменти для вебінарів та онлайн-конференцій:**
  - Zoom, Microsoft Teams, Google Meet — це платформи, які дозволяють викладачам проводити заняття, лекції та семінари в режимі реального часу. Вони підтримують відеозв'язок, демонстрацію екрана, спільне використання документів, інтерактивні дошки, чати та опитування.
  - Такі інструменти широко використовуються не лише в освіті, але й у корпоративному навчанні, професійних тренінгах і воркшопах.
  - Приклад: викладач проводить у Zoom лекцію для 30 студентів, демонструючи слайди, відповідаючи на запитання в чаті та організовуючи групову роботу в окремих кімнатах.
- **Освітні платформи та портали:**
  - «На Урок», «ВсеОсвіта», Google Classroom — це портали, які пропонують доступ до навчальних матеріалів, тестів, інтерактивних уроків і тренажерів. Вони надають викладачам зручний спосіб організовувати навчальний процес, а учням — отримувати матеріали та здавати завдання онлайн.
  - Такі сервіси часто мають базу готових навчальних матеріалів, інтеграцію з хмарними сховищами, функціонал для створення й оцінювання тестів, а також аналітичні інструменти для моніторингу успішності.

- Приклад: учитель створює на платформі «На Урок» курс з фізики, завантажує туди навчальні відео та інтерактивні завдання, а учні проходять їх у зручний для себе час, отримуючи миттєвий зворотний зв’язок.

Освітні сервіси також інтегруються з іншими технологіями, такими як хмарні сховища (Google Drive, OneDrive), системи управління курсами (Moodle, Blackboard), інструменти для створення інтерактивних уроків (Kahoot, Quizlet) та платформи онлайн-курсів (Coursera, edX). Це дозволяє забезпечувати гнучкість навчання, доступ до різноманітного контенту та підтримувати різні формати навчання — від самостійного вивчення матеріалів до інтерактивних занять у реальному часі.

Таким чином, сучасні освітні сервіси є важливим інструментом як у школах і університетах, так і в корпоративному секторі, дозволяючи викладачам і учням ефективно взаємодіяти та отримувати доступ до якісних знань незалежно від відстані й технічних обмежень.

Сервіси штучного інтелекту (AI) — це інструменти, що використовують алгоритми машинного навчання для автоматизації завдань, генерування текстового чи візуального контенту, аналізу даних і оптимізації процесів. Вони спрощують вирішення рутинних завдань, підвищують ефективність у творчих, маркетингових і бізнес-операціях.

Приклади сервісів штучного інтелекту:

- **ChatGPT:**

Чат-бот, розроблений OpenAI, що дозволяє генерувати текстові відповіді на питання користувачів, створювати статті, вести інтерактивні діалоги, а також допомагає у вирішенні складних питань.

Приклад використання: журналіст може попросити ChatGPT підготувати чернетку новинної статті, а бізнес-аналітик — згенерувати звіт на основі введених даних.

- **MidJourney:**

Сервіс генерації зображень, який створює візуальний контент на основі текстових описів. З його допомогою можна швидко отримати унікальні

ілюстрації для маркетингових кампаній, веб-дизайну чи персональних проектів. *Приклад використання:* дизайнер створює концепт нового логотипу або обкладинки альбому, вводячи опис бажаного стилю та елементів.

- **Jasper AI:**

Інструмент, орієнтований на маркетологів і контент-менеджерів, що дозволяє автоматично створювати SEO-оптимізовані тексти, рекламні копії, пости в соціальних мережах та електронні листи.

Приклад використання: менеджер з контенту використовує Jasper AI для створення блогу з ключовими словами, оптимізованими для пошукових систем, або для написання текстів рекламних оголошень, що привертають увагу потенційних клієнтів.

Додаткові можливості:

- Інструменти для автоматизації рутинних завдань, таких як переклади текстів, підготовка чернеток документів чи створення коротких анотацій.
- Аналітичні сервіси, що використовують AI для обробки великих даних, виявлення трендів і формування рекомендацій для бізнесу.
- Сервіси генерації аудіоконтенту, зокрема, синтезаторів голосу та музики на основі текстових описів чи нот.

Сервіси штучного інтелекту, такі як ChatGPT, MidJourney і Jasper AI, є потужними інструментами, які вже сьогодні змінюють підходи до створення контенту, оптимізації бізнес-процесів і реалізації творчих ідей. Завдяки їм компанії та окремі користувачі можуть зменшити час і ресурси, необхідні для виконання завдань, одночасно підвищуючи якість і інноваційність своїх проектів.

Сервіси Інтернету стали ключовим компонентом сучасного життя, змінивши спосіб, у який ми працюємо, спілкуємося, отримуємо освіту, розважаємося та ведемо бізнес. Вони забезпечують швидкий доступ до знань, відкривають нові можливості для співпраці, розширюють ринки, сприяють творчій самореалізації та автоматизують складні процеси. Сьогоднішній

Інтернет пропонує користувачам безпрецедентний рівень гнучкості та доступності інформації, ресурсів і послуг.

Різноманітність доступних платформ і сервісів — від хмарних сховищ і освітніх порталів до штучного інтелекту та маркетплейсів — дозволяє задовольняти потреби найрізноманітніших аудиторій. Ці інструменти створюють середовище, де кожен може знайти щось корисне: бізнес — підвищити ефективність і залучити клієнтів, творчі люди — знайти нові засоби самовираження, студенти — отримати якісну освіту, а звичайні користувачі — зробити своє життя комфортнішим.

Попри наявність викликів, таких як інформаційне перенасичення чи кібербезпека, розвиток інтернет-сервісів демонструє неабиякий потенціал. Їхній подальший прогрес сприятиме ще глибшій інтеграції технологій у повсякденне життя, відкриваючи перед людством нові перспективи та можливості.

3. Розвиток мережевих технологій

Внаслідок прогресу мережевих технологій значно змінилися підходи до обробки й розповсюдження інформації. Спочатку обчислювальні центри, які працювали на базі мейнфреймів, були орієнтовані на централізовану обробку даних. Проте зі збільшенням кількості користувачів і складністю задач виникла потреба у віддаленій обробці даних. Це призвело до появи термінальних мереж, де кожен термінал надавав користувачам доступ до ресурсів центру. Така концепція започаткувала розвиток ранніх форм комп’ютерних мереж.

Наступним великим кроком стало поширення персональних комп’ютерів, які вимагали нових рішень для ефективного обміну інформацією між окремими пристроями. У відповідь з’явилися локальні мережі (LAN), що об’єднували ПК у межах офісів, навчальних закладів або виробничих приміщень. Однак із часом стало очевидно, що цього недостатньо. Бізнес, наука і, зрештою, пересічні користувачі почали потребувати доступу до ресурсів поза межами локальних мереж, що стимулювало розвиток глобальних мереж.

З появою Інтернету інформаційні потоки стали глобальними. Користувачі отримали можливість працювати з електронною поштою, шукати необхідні дані через пошукові системи, обмінюватися файлами та використовувати численні онлайн-сервіси. Глобальна мережа перетворилася на платформу, яка з'єднує мільярди пристроїв по всьому світу, дозволяючи отримувати актуальну інформацію в реальному часі, працювати спільно над проектами, проводити відеоконференції й навіть виконувати складні обчислення.

Нині мережеві технології продовжують розвиватися, забезпечуючи ще більшу швидкість передачі даних, покращену безпеку та надійність. Інновації, такі як хмарні обчислення, Інтернет речей (IoT) і штучний інтелект, інтегруються у мережеву інфраструктуру, створюючи нові можливості для бізнесу, освіти, науки й розваг. Таким чином, мережеві технології стали однією з найважливіших складових інформаційного суспільства, впливаючи на всі аспекти нашого життя.

Мережеві технології створюють фундамент для взаємодії між комп'ютерами, пристроями та користувачами, забезпечуючи передачу даних у цифровій формі. Вони відіграють ключову роль у функціонуванні Інтернету, локальних і корпоративних мереж, що дозволяє не лише обмінюватися інформацією, а й оперативно отримувати доступ до ресурсів, спільно працювати над проектами й підтримувати безперебійний зв'язок.

Сучасні мережеві технології включають широкий спектр рішень: від локальних мереж (LAN) у межах офісу або будинку до глобальних мереж (WAN), які охоплюють континенти й об'єднують мільярди пристроїв. Завдяки їм, користувачі мають можливість не тільки переглядати веб-сторінки чи надсилати електронні листи, але й брати участь у відеоконференціях, обмінюватися великими файлами, використовувати хмарні сервіси для зберігання даних і працювати з віддаленими ресурсами в режимі реального часу.

Крім того, мережеві технології забезпечують основи для сучасних трендів у цифровому світі. Зростання популярності Інтернету речей (IoT) і розгортання

5G-мереж сприяє ще більшій інтеграції пристроїв та підвищенню швидкості передачі даних. Завдяки цьому з’являються нові можливості для автоматизації процесів, впровадження розумних пристроїв у побуті, оптимізації логістики та вдосконалення виробничих ланцюгів.

У майбутньому мережеві технології продовжать розвиватися, підвищуючи ефективність комунікацій, забезпечуючи більшу безпеку передачі даних і розширюючи можливості для співпраці між людьми та машинами. Вони залишаються критично важливим елементом у формуванні глобального цифрового середовища, яке стає невід’ємною частиною життя кожної людини, бізнесу та організації.

Основні компоненти комп'ютерних мереж забезпечують взаємодію між пристроями, дозволяють передавати дані, забезпечують доступ до ресурсів і підтримують зв’язок у межах локальних або глобальних мереж.

1. Клієнт і сервер:

- **Клієнт** — це пристрій або програма, яка надсилає запит на отримання певних ресурсів, наприклад веб-сторінок, файлів чи баз даних.
- **Сервер** — пристрій або програма, що обробляє ці запити і надає відповідні ресурси.

Приклад:

Користувач відкриває веб-браузер на своєму комп’ютері (клієнті) і запитує доступ до веб-сторінки. Сервер веб-хостингу отримує цей запит, знаходить потрібний файл і надсилає його назад клієнту, який відображає сторінку у браузері.

2. Мережеві пристрої:

- **Маршрутизатор (router):**

Зв’язує різні мережі, забезпечуючи передачу даних між ними. Наприклад, домашній маршрутизатор направляє трафік між домашньою Wi-Fi мережею та Інтернетом.

- **Комутатор (switch):**

Розподіляє дані між пристроями в одній локальній мережі. Уявімо, офісна мережа, де кілька комп'ютерів підключені до одного комутатора, який забезпечує швидкий обмін інформацією між ними.

- **Модем:**

Підключає локальну мережу до Інтернет-провайдера. Наприклад, DSL або кабельний модем у квартирі забезпечує доступ до Інтернету, підключаючи домашню мережу до глобальної мережі.

3. Мережеві протоколи:

Протоколи — це правила, які визначають, як передаються дані.

- **TCP/IP:** основний набір протоколів, який забезпечує доставку пакетів даних у потрібну точку.
- **HTTP:** протокол, за яким браузер отримує веб-сторінки із сервера.
- **FTP:** протокол для передачі файлів між клієнтом і сервером.

Приклад:

Коли користувач завантажує файл з веб-сайту, браузер використовує HTTP, а сервер може використовувати FTP для передачі великих обсягів даних.

4. Кабелі та бездротові з'єднання:

- **Кабельні з'єднання:**

Використовуються для стабільного і швидкого з'єднання. Наприклад, оптоволоконний кабель забезпечує високу швидкість передачі даних на великі відстані.

- **Бездротові з'єднання:**

Wi-Fi, 4G, 5G забезпечують зручність і мобільність. Уявімо, що смартфон підключається до Інтернету через Wi-Fi, а ноутбук — через мобільний 4G-роутер.

Сучасні комп'ютерні мережі включають хмарні технології, де сервери часто знаходяться у віддалених дата-центрах. Користувачі отримують доступ до програм і файлів через Інтернет без прив'язки до фізичних пристроїв. Це відкриває нові можливості для спільної роботи, зберігання даних і гнучкості у використанні ресурсів.

Завдяки поєднанню клієнт-серверної моделі, мережевих пристроїв, протоколів і різних типів з’єднань, комп’ютерні мережі продовжують розвиватися. З’являються нові стандарти бездротової передачі, такі як Wi-Fi 6, які забезпечують ще більшу швидкість і стабільність. Інтеграція 5G дозволяє підключати значно більше пристроїв до мережі одночасно, сприяючи розвитку Інтернету речей (IoT) і розумних будинків. У майбутньому ми побачимо ще більше інновацій, які зроблять мережі ще швидшими, безпечнішими та доступнішими для всіх користувачів.

Комп’ютерні мережі класифікуються за кількома критеріями, серед яких масштаби охоплення, тип передачі даних, конфігурація і призначення. Найпоширенішими є такі типи мереж (Таблиця 1):

Таблиця 1.

Типи комп'ютерних мереж

Тип мережі	Опис	Приклад використання
LAN (Local Area Network)	Локальна мережа в межах будівлі або офісу.	Мережа в школі або офісі.
WAN (Wide Area Network)	Глобальна мережа, що охоплює великі відстані.	Інтернет.
WLAN (Wireless LAN)	Бездротова локальна мережа.	Wi-Fi у кав’ярні.
VPN (Virtual Private Network)	Віртуальна приватна мережа для захисту з'єднань.	Захищений віддалений доступ.

1. Локальні мережі (LAN, Local Area Network)

- Призначені для з’єднання пристроїв у межах невеликої географічної території, наприклад офісу, будівлі чи домашньої мережі.
- Забезпечують високу швидкість передачі даних і стабільність.
- Приклад: мережа в офісі, яка об’єднує комп’ютери, принтери й сервери для спільного доступу до ресурсів.

2. Регіональні мережі (MAN, Metropolitan Area Network)

- З’єднують кілька локальних мереж у межах одного міста або агломерації.
- Часто використовуються великими організаціями або місцевими провайдерами для обміну даними на міському рівні.

- Приклад: мережа університетського кампусу, що охоплює всі його корпуси.

3. Глобальні мережі (WAN, Wide Area Network)

- Охоплюють величезні географічні території, включно з різними країнами та континентами.
- Об’єднують локальні та регіональні мережі через високошвидкісні канали зв’язку, супутникові з’єднання чи оптоволоконні магістралі.
- Інтернет є найбільшим прикладом глобальної мережі.

4. Персональні мережі (PAN, Personal Area Network)

- Зазвичай охоплюють невелику територію навколо однієї людини (кілька метрів).
- Використовуються для підключення мобільних пристроїв, ноутбуків, планшетів і гаджетів.
- Приклад: Bluetooth-мережа між смартфоном і бездротовою гарнітурою.

5. Корпоративні або інфраструктурні мережі

- Зазвичай розробляються для великих підприємств чи установ.
- Поєднують різні типи мереж (LAN, WAN) з фокусом на безпеку, централізоване управління та масштабованість.
- Приклад: мережа міжнародної корпорації, яка з’єднує офіси в кількох країнах із доступом до єдиного корпоративного порталу.

6. Спеціалізовані мережі (зокрема, IoT і VPN)

- **Мережі Інтернету речей (IoT):** поєднують пристрої, такі як датчики, камери, побутова техніка, які збирають і передають дані через мережу.
- **VPN (Virtual Private Network):** створюють захищене з’єднання через Інтернет для доступу до внутрішніх ресурсів компанії або для безпечного перегляду веб-сайтів.

Типи комп’ютерних мереж відрізняються за масштабами, метою і технологіями підключення. Вибір типу залежить від завдань, які потрібно виконати: від локального обміну файлами в межах одного приміщення до безпечного доступу до корпоративних ресурсів у різних куточках світу.

Кібербезпека в мережах охоплює широкий набір заходів, спрямованих на забезпечення захисту інформації, пристроїв і мережевої інфраструктури від несанкціонованого доступу, зловмисних дій, крадіжок даних та інших загроз. Вона є невід'ємною складовою сучасних інформаційних технологій, адже мережі стають дедалі складнішими, а зловмисники — більш винахідливими.

Основні загрози та виклики кібербезпеці в мережах:

1. **Хакерські атаки:**
    - **DDoS (розподілені атаки на відмову в обслуговуванні):** величезна кількість запитів може паралізувати роботу веб-сайтів і сервісів.
    - **MITM (людина посередині):** зловмисники перехоплюють трафік між клієнтом і сервером, щоб викрасти конфіденційну інформацію.
    - **Фішинг:** шахраї маскуються під надійні джерела, щоб отримати доступ до паролів, фінансових даних або інших чутливих відомостей.
2. **Зловмисне програмне забезпечення (malware):**
    - Програми-вимагачі (ransomware), які блокують доступ до даних і вимагають викуп.
    - Трояни, що забезпечують зловмисникам прихований доступ до мережі.
    - Шпигунські програми, які стежать за користувачами і передають їхні дані.
3. **Вразливості в програмному забезпеченні та обладнанні:**
    - Невчасно встановлені оновлення або неправильно налаштовані пристрої можуть створювати лазівки для атак.
4. **Недбалість користувачів:**
    - Використання слабких паролів, відкриття підозрілих вкладень або посилань, недотримання політик безпеки.

Методи захисту мереж:

1. **Мережеві екрани (firewalls):**

- Обмежують і контролюють трафік, пропускаючи лише авторизовані запити.

Приклад: компанія використовує апаратний мережевий екран для фільтрації трафіку, забезпечуючи безпечний доступ лише до перевірених серверів.

2. **Системи виявлення і запобігання вторгнень (IDS/IPS):**
   - Аналізують трафік і повідомляють про підозрілі дії або автоматично блокують їх.

Приклад: використання IDS дозволяє адміністраторам оперативно дізнаватися про можливі атаки.

3. **Шифрування даних:**
   - Забезпечує захист даних під час передачі (SSL/TLS, VPN, IPsec) і на сховищах.

Приклад: при доступі до корпоративної пошти через Інтернет дані шифруються протоколом TLS, що ускладнює їх перехоплення.

4. **Багатофакторна аутентифікація (MFA):**
   - Для доступу до мережевих ресурсів потрібні не лише паролі, а й додаткові методи перевірки, такі як SMS-коди, апаратні токени або біометричні дані.

Приклад: для входу в корпоративну VPN-мережу користувач повинен ввести пароль і підтвердити доступ через мобільний додаток.

5. **Навчання персоналу:**
   - Підвищення рівня обізнаності працівників щодо сучасних загроз, методів фішингу, необхідності оновлення паролів.

Приклад: організація проводить щоквартальні тренінги для співробітників з кібербезпеки.

У сучасному світі кібербезпека в мережах не обмежується лише традиційними методами. З розвитком штучного інтелекту та машинного навчання з’являються нові інструменти для виявлення аномальної активності та прогнозування атак. Наприклад, системи, які аналізують поведінку

користувачів і виявляють підозрілі дії, допомагають швидше реагувати на загрози.

Із впровадженням 5G і збільшенням кількості підключених пристроїв (IoT) виникають нові виклики для кібербезпеки. Забезпечення захисту мільярдів пристроїв, від домашніх гаджетів до промислових датчиків, стає пріоритетним завданням.

Кібербезпека в мережах — це багаторівневий процес, що вимагає сучасних технологій, комплексного підходу та навчання користувачів. Постійне вдосконалення методів захисту, активний моніторинг та своєчасна реакція на загрози дозволяють мінімізувати ризики, захищати інформацію та забезпечувати безперебійну роботу мережевих інфраструктур.

Розвиток мережевих технологій став одним із ключових факторів, що визначають сучасний інформаційний світ. Від перших локальних обчислювальних мереж до масштабних глобальних інфраструктур, таких як Інтернет, мережеві технології продемонстрували величезний потенціал для оптимізації роботи, прискорення обміну інформацією та створення нових можливостей для співпраці.

Ці технології вплинули на всі аспекти нашого життя — від бізнесу й науки до освіти, охорони здоров’я та розваг. Вони дозволяють легко ділитися ресурсами, ефективно працювати в команді, миттєво отримувати доступ до знань і розвивати нові форми комунікації.

Подальший розвиток мережевих технологій відкриє ще більше перспектив для інновацій, безпеки та інтеграції, перетворюючи їх на невід’ємну частину не лише сучасного, але й майбутнього глобального суспільства.

4. Інтеграція Інтернету в різні галузі

Інтеграція Інтернету в різні галузі сприяє модернізації традиційних підходів і створює нові можливості для бізнесу, освіти, охорони здоров’я, виробництва, культури та інших сфер. Завдяки доступу до глобальних

мережевих ресурсів, компанії та організації отримують інструменти для автоматизації процесів, зменшення витрат, підвищення ефективності й покращення взаємодії з клієнтами.

Розширення впливу Інтернету у різних сферах:

- **Бізнес і торгівля:**

Інтернет дозволив малим і середнім підприємствам вийти на глобальний ринок, використовуючи платформи електронної комерції. У цьому контексті особливо важливою є роль аналітики даних і інструментів штучного інтелекту, які допомагають прогнозувати попит, оптимізувати запаси й налаштовувати персоналізовані пропозиції для клієнтів.

Приклад: використання платформи Amazon Web Services (AWS) для розгортання масштабованих онлайн-магазинів, здатних обробляти тисячі замовлень щодня.

- **Освіта:**

Освітні інституції активно використовують Інтернет для розширення доступу до знань, створення інтерактивних курсів та віртуальних лабораторій. Зокрема, технології віртуальної реальності (VR) та доповненої реальності (AR) інтегруються у навчальні програми, дозволяючи студентам вивчати складні предмети, такі як медицина чи інженерія, в безпечному і контрольованому середовищі.

Приклад: використання VR-симуляторів для навчання студентів-хірургів.

- **Охорона здоров’я:**

Інтернет не тільки розширив можливості телемедицини, а й сприяв створенню глобальних баз медичних даних. Це дозволяє лікарям у віддалених регіонах отримувати доступ до найновіших досліджень, брати участь у міжнародних конференціях, співпрацювати з колегами по всьому світу і використовувати передові діагностичні алгоритми.

Приклад: застосування глобальних платформ на основі штучного інтелекту, які допомагають у ранньому виявленні онкологічних захворювань.

- **Виробництво та логістика:**

Завдяки Інтернету речей (IoT) і хмарним технологіям підприємства можуть відстежувати виробничі процеси в реальному часі, проводити прогнозне технічне обслуговування обладнання і знижувати витрати. Крім того, інтеграція Інтернету в логістичні системи дозволяє забезпечити точний контроль за переміщенням вантажів, оптимізацію маршрутів і швидку адаптацію до змін ринкових умов.

Приклад: використання IoT-сенсорів для моніторингу температури і вологості в логістичних ланцюгах доставки харчових продуктів.

- **Культура та розваги:**

Інтернет дав змогу культурним установам транслювати події в реальному часі, створювати інтерактивні виставки і забезпечувати доступ до творчих творів широкій аудиторії. Також завдяки стрімінговим сервісам музиканти й кінематографісти отримали нові канали для дистрибуції своїх робіт, а глядачі й слухачі — зручний доступ до контенту на вимогу.

Приклад: трансляція всесвітньо відомих оперних вистав через платформи, такі як Met Opera on Demand.

Сьогодні Інтернет не лише є основою для багатьох бізнес-операцій, але й відіграє важливу роль у формуванні суспільних тенденцій і культурних змін. Мережеві технології сприяють демократизації доступу до знань, розширюючи можливості для освіти в найвіддаленіших регіонах планети. Онлайн-курси, інтерактивні тренажери та хмарні платформи відкривають нові шляхи до саморозвитку для мільйонів людей, незалежно від їхнього місця проживання чи фінансового становища.

Крім того, інтеграція Інтернету у виробничі процеси сприяє підвищенню ефективності та зниженню витрат. Смарт-заводи, де пристрої обмінюються даними в реальному часі, оптимізують ресурси, зменшують ризики простоїв і прискорюють впровадження нових продуктів на ринок. Це не лише робить виробництво гнучкішим, але й створює нові можливості для персоналізації продукції, адаптуючи її під конкретні потреби споживачів.

В охороні здоров’я Інтернет сприяє інтеграції штучного інтелекту та машинного навчання у діагностичні й терапевтичні процеси. Завдяки цьому лікарі отримують інструменти для швидкого аналізу великих обсягів медичних даних, більш точної постановки діагнозу і персоналізації лікування. Підключення пацієнтів до телемедичних платформ дозволяє зменшити навантаження на медичні установи, одночасно забезпечуючи доступність медичних послуг навіть у віддалених районах.

Щодо суспільного життя, Інтернет сприяє активізації громадянської активності. Соціальні мережі, блоги та онлайн-платформи дають змогу швидко поширювати інформацію про суспільні проблеми, організовувати волонтерські акції, залучати кошти на благодійність чи протестні заходи. Це значно впливає на політичні процеси, змінюючи спосіб комунікації між урядами і громадянами, а також підвищуючи прозорість і відповідальність владних структур.

Врешті-решт, інтеграція Інтернету у різні галузі стимулює інновації, знижує бар’єри для входу на ринки та дає змогу як великим, так і малим компаніям створювати продукти й послуги нового покоління. Майбутнє Інтернету тісно пов’язане з подальшим розвитком хмарних технологій, штучного інтелекту, блокчейну та інших інновацій, які ще більше розширять горизонти можливостей для різних сфер життя.

Інтернет також значно вплинув на державний сектор, сприяючи розвитку електронного уряду. Громадяни отримують можливість сплачувати податки, подавати документи, реєструвати бізнес чи отримувати консультації в онлайн-режимі, що економить час і ресурси як для державних органів, так і для населення.

У сфері науки Інтернет став незамінним інструментом для міжнародної співпраці. Раніше дослідники могли витрачати місяці на обмін даними чи узгодження спільних проектів. Сьогодні ж завдяки мережевим технологіям це відбувається миттєво. Віддалений доступ до суперкомп’ютерів, участь у спільних експериментах у реальному часі, автоматизоване збирання й аналіз даних — усе це стало звичайною практикою. Крім того, відкриті наукові бази

даних і платформи для обміну дослідницькими результатами сприяють пришвидшенню інноваційного процесу, розширюючи горизонти відкриттів і впровадження нових технологій.

У культурній сфері інтеграція Інтернету дозволила зберігати та популяризувати культурну спадщину. Онлайн-музеї, цифрові архіви, віртуальні екскурсії до архітектурних пам'яток і бібліотек роблять культурні цінності доступними для широкої аудиторії. Люди з будь-якого куточка світу тепер можуть вивчати мистецтво, історію, літературу, музику та інші аспекти світової культури, не покидаючи свого дому.

Отже, інтеграція Інтернету у різні галузі не лише змінює підходи до роботи, навчання й розваг, а й формує нові способи мислення, спілкування й взаємодії з навколишнім світом.

У майбутньому інтеграція Інтернету буде ще більш глибокою, сприяючи появі нових галузей, таких як дистанційне обслуговування робототехнічних систем, автоматизовані роздрібні мережі чи персоналізовані системи навчання. Інтернет продовжуватиме об'єднувати людей, компанії та ідеї, створюючи нові можливості для інновацій, співпраці та глобального розвитку.

Контрольні питання

1. Що таке Інтернет і як він впливає на функціонування інформаційних систем?
2. Які основні компоненти складають архітектуру Інтернету?
3. Які існують моделі взаємодії клієнт-сервер у мережевих технологіях?
4. Які основні типи комп'ютерних мереж і як вони використовуються в інформаційних системах?
5. Як мережеві технології сприяють обміну даними та підвищенню ефективності бізнес-процесів?
6. Які ризики та виклики виникають у сфері кібербезпеки при роботі з мережевими технологіями?

Тема 8.
БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ

Метою лекції є ознайомлення слухачів із сучасними підходами, принципами та методами забезпечення безпеки інформаційних систем, а також із основними загрозами та викликами, що постають перед організаціями у процесі захисту конфіденційності, цілісності та доступності інформаційних ресурсів. Лекція покликана сформувати розуміння важливості інформаційної безпеки, розглянути ключові концепції управління ризиками, ознайомити з основними технологіями, інструментами та стандартами у сфері захисту даних.

План лекції

1. Характеристика загроз інформаційної безпеки.
2. Поняття інформаційної війни.
3. Технології та інструменти захисту.

1. Характеристика загроз інформаційної безпеки

У сучасному світі інформація та інформаційні технології стали одним із найцінніших стратегічних ресурсів, що визначають економічну та оборонну спроможність держави. Інформація вже не просто засіб підтримки діяльності — вона перетворилася на основу конкурентоспроможності й національної безпеки. Проте поширення та впровадження інформаційних технологій породило безліч викликів та ризиків, які вимагають особливої уваги.

По-перше, повсюдне використання комп’ютерних технологій у різних сферах — від державного управління до приватного сектору — призвело до появи нових внутрішніх і зовнішніх загроз. Традиційні методи захисту інформації вже не завжди ефективні. Сьогодні ми спостерігаємо активізацію нетрадиційних каналів витоку даних, а також постійно зростаючий ризик несанкціонованого доступу до критичних інформаційних ресурсів.

По-друге, інтеграція державних і корпоративних інформаційних систем до глобального інформаційного простору створює потенційні можливості для масового несанкціонованого контролю за інформаційними потоками. Це включає навмисне втручання в роботу систем, модифікацію даних, а також інші форми кіберзагроз. Усе це може мати серйозні наслідки для національної безпеки та економічної стабільності.

По-третє, в сучасних умовах дедалі частіше застосовуються засоби інформаційної зброї, спрямованої на порушення інформаційної інфраструктури, маніпулювання громадською думкою, дезінформацію та проведення інформаційних війн. Ці методи стають звичайною реальністю, що вимагає від держав і організацій постійної готовності до захисту своїх інформаційних активів.

Нарешті, відставання в розвитку вітчизняних інформаційних технологій змушує закуповувати іноземне обладнання та залучати іноземних фахівців для створення інформаційних систем. Це не лише підвищує ризик несанкціонованого доступу до оброблюваних даних, але й формує залежність від закордонних виробників телекомунікаційного обладнання, комп'ютерної техніки та програмного забезпечення. Як наслідок, зростає вразливість критичної інформаційної інфраструктури перед зовнішніми загрозами.

Таким чином, у сучасних умовах особлива увага приділяється забезпеченню інформаційної безпеки. Необхідно не тільки створювати надійні механізми захисту даних, але й розвивати власний технологічний потенціал, щоб зменшити залежність від іноземних рішень та посилити позиції держави у сфері інформаційних технологій.

Інформатизація – це комплексний процес, який охоплює організаційні, соціально-економічні та науково-технічні заходи, спрямовані на створення умов для задоволення зростаючих інформаційних потреб. Основна мета інформатизації – забезпечити доступ громадян, органів державної влади та управління до актуальних, достовірних і зручних для використання інформаційних ресурсів.

Цей процес передбачає розвиток інформаційних систем, комп'ютерних мереж, баз даних та інформаційних технологій, які працюють на основі сучасної обчислювальної та комунікаційної техніки. Інформатизація також сприяє підвищенню ефективності прийняття рішень, зменшенню адміністративних бар'єрів, оптимізації управлінських процесів і вдосконаленню якості надання державних послуг.

Окрім технічного аспекту, інформатизація має соціальну складову: вона дозволяє громадянам реалізувати своє право на доступ до інформації, сприяє прозорості та відкритості діяльності органів влади, стимулює розвиток освіти, науки та культури. Таким чином, інформатизація не лише змінює технологічний ландшафт, але й трансформує суспільні відносини, надаючи нові можливості для економічного зростання, соціального розвитку та зміцнення демократії.

Інформаційна безпека є однією з ключових складових національної безпеки, яка охоплює управління ризиками та загрозами в інформаційній сфері. Вона передбачає комплекс заходів, спрямованих на забезпечення інформаційного суверенітету країни, підтримку достовірного та позитивного іміджу в міжнародному інформаційному просторі, а також захист внутрішнього інформаційного середовища.

Інформаційна безпека включає:

- Удосконалення державного регулювання у сфері інформаційної політики, що сприяє впровадженню новітніх технологій, які підвищують захищеність критичних інформаційних ресурсів.
- Створення умов для поширення правдивої та позитивної інформації про Україну як всередині країни, так і за її межами. Це включає проактивну роботу із засобами масової інформації та налагодження ефективної комунікації на міжнародному рівні.
- Залучення медіа до протидії корупції, зловживанню владою та іншим факторам, які становлять загрозу національній безпеці.

- Забезпечення конституційного права громадян на свободу слова та доступ до інформації, що є необхідною умовою для формування демократичного суспільства та довіри до державних інститутів.

Таким чином, інформаційна безпека — це не лише питання технічного захисту даних, але й стратегічний підхід до формування інформаційного середовища, яке підтримує стійкість держави, її міжнародний авторитет та безпеку громадян.

Загрози інформаційній безпеці охоплюють широкий спектр ризиків і небезпек, які можуть порушити конфіденційність, цілісність або доступність інформаційних ресурсів. Ці загрози походять з різних джерел, мають різні способи реалізації та можуть призводити до суттєвих наслідків як для окремих організацій, так і для національної безпеки в цілому. Їхня класифікація і глибоке розуміння дають змогу ефективніше планувати заходи захисту та реагувати на інциденти.

1. За джерелами походження:

- **Внутрішні загрози:**

Загрози, що походять від працівників або інших осіб, які мають легальний доступ до інформаційних систем. Це можуть бути навмисні дії (зловживання привілеями, викрадення даних) або випадкові помилки (випадкове видалення файлів, неправильна конфігурація систем).

Приклад: співробітник компанії копіює конфіденційну документацію та передає її конкурентам.

- **Зовнішні загрози:**

Атаки від сторонніх осіб або організацій, які не мають авторизованого доступу до системи. До них належать хакери, кіберзлочинні угруповання, державні чи недержавні актори, які прагнуть викрасти дані, вплинути на роботу систем або зруйнувати їх.

Приклад: кіберзлочинці запускають вірус-вимагач, що шифрує дані на комп’ютерах організації, і вимагають викуп.

2. За характером дії:

- **Пасивні загрози:**

Ці загрози спрямовані на непомітне отримання або перехоплення інформації без її змін. Вони можуть бути складнішими для виявлення, адже не залишають явних слідів втручання.

Приклад: використання сніффера для перехоплення незашифрованих даних у мережі.

- **Активні загрози:**

Включають дії, які безпосередньо змінюють, пошкоджують або знищують інформацію. Такі атаки часто спричиняють очевидні порушення роботи системи.

Приклад: вірус, який видаляє системні файли, порушуючи роботу операційної системи.

3. За об'єктами атаки:

- **Критично важливі інфраструктури:**

Системи, від яких залежить функціонування енергетики, транспорту, охорони здоров'я, фінансів і національної оборони.

Приклад: атака на енергетичну мережу, що спричиняє масштабні відключення електроенергії.

- **Приватні дані та особиста інформація:**

Інформація про користувачів, клієнтів або співробітників, яка може бути викрадена, продана або використана для шантажу.

Приклад: злом бази даних медичних записів і продаж цієї інформації на чорному ринку.

4. За технічними методами реалізації:

- **Мережеві загрози:**

Включають DDoS-атаки, MITM-атаки, злам Wi-Fi мереж, а також використання експлойтів у протоколах.

Приклад: перевантаження веб-сайту мільйонами запитів, що робить його недоступним для звичайних користувачів.

- **Програмні загрози:**

Віруси, трояни, руткіти, програмне забезпечення-вимагачі (ransomware).

Приклад: троян, який збирає облікові дані користувачів і пересилає їх на сервер зловмисника.

- **Загрози соціальної інженерії:**

Використання обману, маніпуляцій або фішингових кампаній для отримання конфіденційної інформації.

Приклад: фальшивий електронний лист від "банку", який просить користувача оновити дані свого облікового запису.

5. За масштабом:

- **Локальні загрози:**

Порушення безпеки, що впливають на одну організацію або обмежене коло користувачів.

Приклад: втручання у внутрішню базу даних компанії для модифікації звітів.

- **Глобальні загрози:**

Загрози, що мають вплив на кілька організацій, галузей або навіть цілу країну.

Приклад: масштабна кібератака, яка порушує роботу банківської системи у кількох державах.

6. За рівнем технічної складності:

- **Тривіальні загрози:**

Використання відомих уразливостей, які легко усунути за допомогою оновлень та базових політик безпеки.

Приклад: атака на стару версію веб-додатка, яка не була оновлена.

- **Складні, багаторівневі загрози:**

Мета яких – обійти кілька рівнів захисту, використовуючи комбінацію різних методів атаки.

Приклад: зловмисники спочатку отримують доступ до внутрішньої мережі через фішинг, а потім використовують складні експлойти для викрадення даних.

Загрози інформаційної безпеки — це різноманітні та динамічні ризики, які постійно змінюються. Вони можуть бути як простими, так і надзвичайно складними, як локальними, так і глобальними. Знання про характеристики цих загроз дозволяє краще їх ідентифікувати, оцінити ризики та розробити ефективні стратегії захисту. Використання передових технологій, регулярний моніторинг, навчання персоналу і постійне вдосконалення заходів безпеки є ключовими складовими успішного протистояння сучасним кіберзагрозам.

2. Поняття інформаційної війни

Термін «**інформаційна війна**» вперше був вжитий у наприкінці 80-х років XX століття, в період, коли Сполучені Штати активно розвивали свої стратегії використання інформаційних ресурсів у військових конфліктах. Цей термін виник завдяки аналітичним і стратегічним розробкам американських військових теоретиків, які намагалися зрозуміти, як інформація може стати ключовим елементом у сучасних формах протистояння. Особливої популярності цей термін набув після завершення Холодної війни, коли вплив інформаційних технологій на політичну й економічну стабільність СРСР став очевидним.

Справжній імпульс концепція інформаційної війни отримала під час війни в Перській затоці у 1991 році. Тоді вперше відкрито продемонстрували, як за допомогою інформаційних технологій можна не лише покращити управління військовими операціями, але й впливати на громадську думку, деморалізувати супротивника і мінімізувати власні втрати. Під час цієї кампанії американські військові активно використовували супутниковий зв'язок, системи глобального позиціонування (GPS), засоби радіоелектронної боротьби та інші сучасні технології, які стали основою для подальшого розвитку концепції інформаційної війни.

Після подій в Іраку термін «інформаційна війна» почав стрімко поширюватися в експертних колах і став основою для створення цілого ряду

доктрин і стратегій, пов'язаних з інформаційною безпекою та управлінням інформаційними потоками. Він не тільки позначав новий тип військової активності, а й визначав нову епоху, в якій інформація та технології обробки даних стали вирішальними елементами як у військових конфліктах, так і в геополітичному суперництві.

Інформаційна війна — це комплексний процес протистояння між різними суб’єктами, що використовують інформаційні ресурси та технології для досягнення стратегічних цілей. Вона включає застосування інформаційної зброї, тактики маніпулювання даними, дезінформації, кібернападів і навіть соціальної інженерії з метою отримання переваги над опонентом. Таке протистояння може виникати між державами, корпораціями, політичними групами або навіть окремими особами.

Інформаційна зброя — це засоби та методи, які дозволяють здійснювати контрольований вплив на інформаційні потоки. Це може включати модифікацію переданих даних, знищення критичної інформації, створення фальшивих повідомлень або маніпулювання алгоритмами обробки даних. Наприклад, хакерські атаки, спрямовані на зміну виборчих результатів через маніпуляцію базами даних, або кампанії з дезінформації, які впливають на громадську думку.

Інформаційна загроза — це зовнішній або внутрішній вплив, який може активувати небажані зміни в роботі інформаційної системи. Такі загрози можуть виникати як через навмисне впровадження шкідливого коду, так і через несанкціоноване втручання в алгоритми, що забезпечують стабільну роботу системи. Прикладом може бути шкідливе програмне забезпечення, яке вводить у систему невірні дані, призводячи до збою в автоматизованих процесах.

Сугестія — це прихований вплив на інформаційну систему, яка здатна навчатися та адаптуватися. Цей вплив спрямований на зміну її поведінки, орієнтуючи систему на виконання завдань, які можуть суперечити її первинним цілям. Наприклад, це може бути прихована модифікація навчальних даних для

штучного інтелекту, яка змушує його надавати неправильні результати або формувати небажані шаблони.

Сугестивний вплив — це вид маніпуляції, коли інформаційній системі непомітно задаються нові цілі, про які вона сама «не знає». Це може відбуватися через контроль навчальних алгоритмів або зміни умов середовища, у якому працює система. Прикладом є зміна умов роботи чат-бота таким чином, щоб він почав пріоритизувати певні теми або відповіді, які відрізняються від початково заданих принципів.

Приклади:

1. **Інформаційна війна в політиці:** використання соціальних мереж для поширення дезінформації під час виборчих кампаній.
2. **Інформаційна зброя в кіберпросторі:** атаки типу "відмова в обслуговуванні" (DDoS), які паралізують роботу критично важливих ресурсів.
3. **Інформаційна загроза в бізнесі:** впровадження шкідливого коду, який змінює фінансові дані, викликаючи помилки в розрахунках або звітах.
4. **Сугестія в системах штучного інтелекту:** внесення змін у навчальний набір даних, через що система починає робити неправильні прогнози.
5. **Сугестивний вплив у розробці алгоритмів:** маніпуляція метаданими для забезпечення непропорційного впливу на результати роботи алгоритму, що змінює пріоритети його рішень.

Ці концепції демонструють складність і багатогранність сучасної інформаційної війни, де кожен компонент відіграє свою роль у досягненні стратегічних цілей.

Цілі інформаційної війни суттєво відрізняються від традиційних воєнних конфліктів. Її метою є не фізичне знищення супротивника чи ліквідація його збройних сил, а порушення ключових компонентів функціонування держави, таких як фінансова, транспортна та комунікаційна інфраструктура. Ця форма війни прагне дестабілізувати економічні процеси, створити хаос у системах управління та змусити населення країни, що зазнала атаки, приймати рішення

під впливом країни-агресора. Іншими словами, йдеться про руйнування економічного, соціального та інформаційного середовища, що є основою стабільності держави.

Основні напрями дій у межах інформаційної війни включають:

- **Атаки на інфраструктуру життєзабезпечення:**

Напади на телекомунікаційні мережі, транспортні системи, енергетичні об'єкти (наприклад, електростанції) для створення тривалих перебоїв у постачанні ресурсів, транспортному сполученні та інформаційному обміні.

Приклад: кібератака, яка виводить з ладу систему управління електростанцією, залишаючи регіон без електроенергії.

- **Промислове шпигунство та економічна розвідка:**

Викрадення або викривлення комерційно цінної інформації, інтелектуальної власності, патентів чи дослідницьких даних для отримання конкурентної переваги. Це також може включати маніпуляцію даними, яка призводить до фінансових втрат чи стратегічних помилок.

Приклад: злам корпоративної мережі великої фармацевтичної компанії з метою викрадення формули нового препарату.

- **Кібератаки на персональні та захищені дані:**

Злом баз даних, викрадення ідентифікаційної інформації, проникнення до систем з обмеженим доступом для отримання конфіденційної інформації або її використання у пропагандистських цілях.

Приклад: крадіжка персональних даних тисяч користувачів із метою шантажу або поширення дезінформації.

Окрім зазначених методів, інформаційна війна може також включати кампанії дезінформації, спрямовані на зміну громадської думки, підрив довіри до влади чи створення соціальної напруги. Це може бути організоване через соціальні мережі, мас-медіа, блог-платформи або підставні інтернет-ресурси.

Цілі інформаційної війни — не лише у створенні безладу та економічної шкоди. Вони також можуть включати нав'язування альтернативної картини реальності, маніпуляцію свідомістю людей і досягнення політичних цілей без

прямого військового вторгнення. Таким чином, інформаційна війна стала важливою складовою сучасних конфліктів, що вимагає не тільки технічних, а й соціальних, політичних і психологічних заходів для її попередження та нейтралізації.

Основним інструментом інформаційної війни виступає інформаційна зброя, що охоплює різні технологічні й психологічні методи впливу на інформаційні системи, ресурси та свідомість людей.

Інформаційна зброя – це пристрої, технології та методи, які використовуються для завдання значної шкоди супротивнику у процесі інформаційного протистояння, здійснюючи цілеспрямований і потенційно небезпечний вплив на його інформаційні ресурси, системи чи середовище.

Цей термін має широке значення і включає засоби, спрямовані на знищення, модифікацію або викрадення інформації, а також на формування певного уявлення про події або реальність.

До інформаційної зброї можна віднести:

- **Інформаційно-технічні засоби:**

Це інструменти, які дозволяють проникнути в захищені інформаційні системи, обійти механізми контролю доступу, змінити, знищити або викрасти дані. До таких засобів належать шкідливе програмне забезпечення (віруси, трояни, руткіти), кібератаки на сервери, впровадження фальшивих сертифікатів для злому зашифрованих каналів зв’язку тощо.

Приклад: використання програмного забезпечення для шифрування даних з наступним вимаганням викупу, або злом баз даних для викрадення конфіденційної інформації, наприклад, фінансових записів чи персональних даних.

- **Інформаційно-психологічні засоби:**

Ці інструменти спрямовані на формування у громадськості помилкових уявлень або зміщення акцентів у сприйнятті подій. Інформаційно-психологічна зброя використовує методи дезінформації, маніпуляції суспільною думкою,

розповсюдження фейкових новин, створення пропагандистських матеріалів, спрямованих на деморалізацію населення або підрив довіри до влади.

Приклад: масова інформаційна кампанія в соціальних мережах, яка поширює неправдиву інформацію про економічну кризу або політичні змови з метою викликати паніку та дестабілізувати суспільство.

Інформаційна зброя також включає засоби маніпуляції алгоритмами пошукових систем, системи штучного інтелекту, які створюють контент, що вводить в оману, використання бот-мереж для автоматизованого поширення певних наративів, а також підробку документів або підміни відео та аудіо для створення неправдивих свідчень (так звані "діпфейки").

Сукупність інформаційно-технічних та інформаційно-психологічних методів дозволяє впливати не лише на окремих осіб чи організації, а й на цілі держави. За допомогою інформаційної зброї можна послабити економічну стабільність, спровокувати політичну кризу, знизити рівень довіри до державних інституцій та міжнародних партнерів. У цьому сенсі інформаційна зброя стає потужним інструментом впливу на глобальному рівні.

Об’єктами інформаційного впливу можуть виступати як суто технічні, так і соціальні елементи. До них належать інформаційно-технічні системи, які забезпечують обробку, зберігання та передачу даних, а також інформаційно-аналітичні системи, які використовуються для аналізу та прийняття рішень. У багатьох випадках до складу цих систем входить людина, яка стає невід’ємним компонентом процесу обробки та інтерпретації інформації. Крім того, об’єктами впливу можуть бути інформаційні ресурси, тобто самі дані або знання, а також суспільні механізми формування громадської думки. Останнє включає засоби масової інформації, соціальні мережі, канали пропаганди та інші інструменти, які впливають на суспільну свідомість та сприйняття реальності. Нарешті, ще одним важливим об’єктом впливу є психіка людини, оскільки інформаційні атаки часто спрямовані на зміну її емоційного стану, переконань і поведінки.

Інформаційна війна, таким чином, може бути спрямована проти трьох ключових складових:

- **Технічна інфраструктура:** комп'ютери, сервери, мережеві пристрої.
- **Програмне забезпечення:** операційні системи, прикладні програми, алгоритми обробки даних.
- **Людина:** як користувач, так і суб'єкт прийняття рішень, що піддається маніпуляціям і психологічному тиску.

Головною метою таких впливів часто є не лише дестабілізація інфраструктури чи викрадення інформації, а й фундаментальна зміна світогляду людини. Це може включати придушення її морального та творчого потенціалу, нав'язування альтернативної картини світу чи підрив довіри до традиційних джерел інформації. Наприклад, у ході дезінформаційних кампаній створюється спотворена реальність, у якій людина починає приймати неправдиві дані за істину, змінюючи свої рішення, поведінку й ставлення до певних подій або явищ. Усі ці впливи разом формують багатогранну стратегію інформаційної війни, яка значно перевершує традиційні уявлення про конфлікт у цифрову епоху.

3. Технології та інструменти захисту

Захист інформації є проблемою, яка супроводжує людство протягом усього його існування. Різні підходи до її вирішення завжди залежали від рівня технологічного розвитку. У сучасному інформаційному суспільстві технології стали тим чинником, який загострює цю проблему: комп'ютерні злочини сьогодні є однією з характерних рис нашого часу.

Відповідно до Закону України «Про захист інформації в автоматизованих системах», **захист інформації –** це комплекс організаційних, технічних і правових заходів, спрямованих на забезпечення її безпеки. Метою цих заходів є недопущення завдання шкоди інтересам власників інформації, автоматизованих систем, а також користувачів, які мають доступ до цієї інформації.

У рамках захисту інформації використовуються як адміністративні методи (встановлення політик доступу, контроль за дотриманням процедур), так і технічні рішення (шифрування даних, мережеві брандмауери, антивірусні програми). Крім того, правова складова включає регулювання відносин між суб’єктами, що володіють або обробляють інформацію, та накладення відповідальності за порушення встановлених правил.

Захист інформації є багаторівневим процесом, який охоплює:

- **Фізичний захист:** обмеження доступу до обладнання та серверних приміщень.
- **Технічний захист:** використання сучасних засобів шифрування, аутентифікації, систем виявлення вторгнень.
- **Організаційний захист:** впровадження політик безпеки, регулярне навчання персоналу, проведення аудиту безпеки.
- **Правовий захист:** встановлення відповідальності за порушення норм захисту інформації, регламентація умов доступу та використання інформації.

Завдяки інтегрованому підходу до захисту інформації забезпечується цілісність, конфіденційність і доступність даних, що є необхідними умовами для ефективного функціонування сучасних автоматизованих систем.

Комп’ютерними злочинами вважаються як втручання в роботу комп’ютерних систем, так і ті правопорушення, де комп’ютери виступають як ключові технічні засоби.

Основні причини комп'ютерних злочинів та викрадення інформації включають низку чинників, які зумовлюють високий рівень ризику для сучасних інформаційних систем:

1. **Перехід від паперових до електронних технологій зберігання даних:**

Різкий перехід на електронний формат супроводжувався недостатнім розвитком технологій захисту даних. Традиційні способи захисту інформації не встигали

адаптуватися до нових викликів, створюючи сприятливе середовище для викрадення та несанкціонованого використання цифрових даних.

Приклад: компанія переводить свою документацію з паперової в електронну форму, але не вживає належних заходів безпеки. Як наслідок, злочинці отримують доступ до комерційних секретів через незахищені сервери.

2. **Широке впровадження мережевих технологій:**

Зростання кількості локальних і глобальних обчислювальних мереж забезпечило користувачам легший доступ до інформаційних ресурсів, але також створило додаткові вектори атаки. Наявність глобальних мереж, таких як Інтернет, значно розширила потенційне коло зловмисників.

Приклад: несанкціонований доступ до баз даних через вразливості у відкритих API, що надають клієнтам доступ до їхніх облікових записів у банківській системі.

3. **Ускладнення програмних засобів:**

Програми та операційні системи стають дедалі складнішими, що, в свою чергу, призводить до зростання кількості вразливостей. Помилки в коді, неналежне тестування та оновлення програм створюють можливості для експлойтів і атак.

Приклад: хакери використовують нову вразливість у популярній CRM-системі для викрадення клієнтських баз даних компаній, які не встигли встановити оновлення безпеки.

Перелічені фактори створюють сприятливі умови для комп'ютерних злочинів і викрадення даних. Щоб протидіяти таким загрозам, необхідно постійно вдосконалювати технології захисту, підвищувати обізнаність користувачів і впроваджувати ефективні практики управління інформаційною

Основними принципами інформаційної безпеки є забезпечення цілісності даних, конфіденційності та доступності для авторизованих користувачів. Ці принципи створюють фундамент для захисту інформаційних ресурсів, оскільки будь-яке порушення цих умов може мати серйозні наслідки як для організацій, так і для окремих користувачів.

До основних порушень безпеки інформації відносяться:

- **Несанкціонований доступ:**

Це доступ до інформації, який здійснюється з порушенням встановлених правил. Наприклад, хакер отримує доступ до закритої бази даних через злом пароля адміністратора.

Приклад: У великий онлайн-магазин незаконно проникає зловмисник, щоб викрасти дані клієнтів.

- **Витік інформації:**

Результат дій, унаслідок яких конфіденційна інформація стає відомою стороннім особам. Це може бути як навмисний витік, наприклад, з боку невдоволеного співробітника, так і випадковий, через недбалість або помилку в налаштуваннях.

Приклад: Витік персональних даних клієнтів через публікацію на відкритому сервері.

- **Втрата інформації:**

Ситуація, коли дані перестають існувати або стають недоступними для своїх власників. Це може статися через технічні збої, ненавмисне видалення файлів або відмову обладнання.

Приклад: Через збої в системі резервного копіювання організація втрачає ключові фінансові записи.

- **Підробка інформації:**

Навмисні дії, спрямовані на зміну або перекручення даних. Це може включати зміну фінансових звітів, підробку документів або внесення змін до даних в інформаційній системі.

Приклад: Співробітник змінює дані в системі обліку, щоб приховати факти фінансових зловживань.

- **Блокування інформації:**

Дії, які припиняють доступ до даних. Це можуть бути шкідливі програми, що шифрують файли (наприклад, програми-вимагачі), або внутрішні зміни в політиках доступу.

Приклад: Шифрувальник блокує доступ до бази даних підприємства, вимагаючи викуп за ключ дешифрування.

- **Порушення роботи інформаційної системи:**

Будь-які обставини, які спотворюють нормальний процес обробки даних. Це можуть бути як технічні помилки, так і навмисні дії зловмисників.

Приклад: DDoS-атака на сервер, що призводить до його перевантаження і неможливості обробляти запити клієнтів.

Сучасні системи захисту покликані мінімізувати ризики кожного з цих порушень. Наприклад, багаторівнева система авторизації допомагає знизити ризик несанкціонованого доступу, шифрування даних зменшує ймовірність витоку інформації, а регулярне резервне копіювання дозволяє швидко відновити втрачені дані. Разом із належною підготовкою персоналу та постійним моніторингом безпеки ці заходи створюють міцний фундамент для забезпечення інформаційної безпеки в організації.

ПРИКЛАД

Порушення інформаційної безпеки, як правило, пов'язані з неправомірним доступом до ресурсів.

Так, у результаті шестимісячного розслідування ЦРУ виявило суттєві порушення серед своїх співробітників. Четверо працівників були звільнені через створення і використання прихованого чату в мережі підрозділу, внаслідок чого їх визнали ненадійними й непридатними для роботи в аналогічних установах. Серед звільнених був високопосадовець, який обіймав одну з провідних позицій у розвідувальному управлінні. Крім цього, 96 інших осіб отримали різноманітні дисциплінарні покарання.

Таємний чат, що виник у 1980-х роках, використовувався приблизно 160 співробітниками для неформального спілкування, жартів та розмов без дотримання встановлених правил безпеки. ЦРУ у своїй офіційній заяві назвало це «грубим порушенням цілісності мережі». Інцидент ще раз підкреслив наявність проблем із захистом інформації в організації та серйозне ставлення до їх усунення.

Це не єдиний випадок, коли інформаційна безпека ставала причиною гучних скандалів у ЦРУ. Наприкінці 1996 року Джона Дейча, тодішнього директора управління, було звільнено за те, що він зберігав секретні документи на домашньому комп'ютері з підключенням до Інтернету. Такі інциденти вчергове доводять важливість дотримання суворих правил безпеки навіть у найбільш закритих і стратегічно важливих структурах.

Основною загрозою для інформаційної безпеки зазвичай є самі люди. Їхні дії як навмисні, так і випадкові завжди слід враховувати при створенні системи захисту. Експерти в галузі комп'ютерної безпеки класифікують порушників на чотири основні групи:

- ***Сторонні особи, які не знайомі з діяльністю організації.***
- ***Сторонні особи, які мають певні знання про організацію, включаючи колишніх співробітників.***
- ***Непрограмісти серед поточного персоналу.***
- ***Програмісти, які працюють в організації.***

Цей поділ дозволяє більш ефективно прогнозувати потенційні загрози і планувати відповідні заходи безпеки.

Для опису різних типів комп'ютерних злочинців часто використовуються терміни «хакери», «кракери», «пірати» та «шкідники».

Хакери (або хекери) — це узагальнена назва для людей, які займаються зламом комп'ютерних систем. Іноді цей термін також стосується програмістів, одержимих створенням або модифікацією програмних продуктів, що не обов'язково мають практичну цінність. Згідно з однією з версій, слово «hack» почали використовувати в Массачусетському технологічному інституті для позначення проєктів, які виконуються скоріше заради задоволення, ніж для реального застосування. У вузькому значенні «хакери» — це ті, хто отримує несанкціонований доступ до ресурсів, керуючись переважно цікавістю та бажанням самоствердження. Відмінністю хакерів від професійних зловмисників є те, що вони часто не переслідують комерційної вигоди, а діють радше через цікавість чи спробу випробувати свої можливості.

Кракери (або крекери), навпаки, є професійними порушниками безпеки. Вони використовують свої навички для серйозних протиправних дій і не мають жодних моральних обмежень. Основною метою кракерів є злам систем задля отримання доступу до конфіденційних даних, фінансової інформації або ресурсів компаній.

Пірати становлять найбільш криміногенну категорію. Це професіонали високого рівня, які спеціалізуються на крадіжках комерційних програм, технологічних ноу-хау та іншої інтелектуальної власності. Їхня діяльність здебільшого виконується на замовлення або для конкретного покупця. У випадках, коли таких замовлень немає, пірати можуть переключитися на викрадення даних кредитних карток, банківських рахунків чи зловживання телефонними лініями. Головний мотив їхніх дій — матеріальна вигода, а не цікавість чи розваги.

Статистика свідчить, що більшість випадків розкрадання інформації пов’язана з діями працівників організацій, а не із зовнішніми атаками. Згідно з дослідженням корпорації IDG, 88 % таких злочинів здійснюються співробітниками компаній, і лише 12 % — результат зовнішніх проникнень із використанням спеціальних технічних засобів.

Приклад

Прикладом може слугувати атака української кіберармії, до якої долучилися хакери з багатьох країн світу, на російські державні вебресурси та приватні компанії після початку повномасштабного вторгнення росії в Україну. Зокрема, у перші дні війни було зламано і тимчасово виведено з ладу сайти кремля, Міністерства оборони рф, Центрального банку та великих державних корпорацій. Багато з цих атак координувались через соціальні мережі та спеціалізовані канали в месенджерах, а їхніми цілями були не лише збитки, але й психологічний вплив, демонстрація вразливості та підтримка української оборони.

Шкідники, також відомі як вандали, реалізують у кіберпросторі свої руйнівні схильності: вони заражають системи вірусами, частково або повністю

виводять їх з ладу, і часто роблять це без будь-якої матеріальної вигоди. Найчастіше їхнім єдиним стимулом є моральне задоволення, яке вони отримують від нанесення шкоди, або прагнення помсти.

Існує також група, яка займає проміжне положення між професійними хакерами і недосвідченими користувачами. Це так звані «**експериментатори**» або «**піонери**». Як правило, це молоді люди, які, освоюючи інструменти і ресурси комп'ютерних мереж, прагнуть навчатися лише на власних помилках, вивчаючи, «як не треба». Основну частину цієї групи становлять діти та підлітки. Їхня мотивація переважно ігрова. Часто з таких експериментаторів виростають висококваліфіковані фахівці, в тому числі й ті, які поважають закон.

Серед головних причин порушень інформаційної безпеки можна назвати недостатню реалізацію творчого потенціалу та недооцінку можливих наслідків протиправних дій. Цей чинник є універсальним і не залежить від національної приналежності чи професійної сфери. Попри те, що особисті проблеми не можуть виправдати незаконну діяльність, суспільство тільки починає формувати правильне ставлення до комп'ютерних злочинців.

Контрольні питання

1. Які основні види загроз інформаційної безпеки існують і як вони впливають на функціонування інформаційних систем?
2. Що таке інформаційна війна і які основні цілі вона переслідує?
3. Які відмінності існують між хакерами, кракерами, піратами та вандалами?
4. Як шифрування та багаторівнева аутентифікація сприяють захисту інформації?
5. Які інструменти дозволяють виявляти та запобігати кібератакам?
6. Які юридичні та організаційні заходи допомагають забезпечити інформаційну безпеку?
7. Що таке сугестія в контексті інформаційної війни і як вона впливає на інформаційну систему?

Тема 9.
ШТУЧНИЙ ІНТЕЛЕКТ І МАШИННЕ НАВЧАННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Мета лекції – ознайомити слухачів із основними поняттями, принципами роботи та можливостями штучного інтелекту (ШІ) та машинного навчання (МН), зокрема їх застосуванням в інформаційних системах. Лекція спрямована на те, щоб показати, як інструменти ШІ та МН допомагають у прийнятті рішень, автоматизації процесів, аналізі великих обсягів даних, а також виявленні закономірностей, що сприяють підвищенню ефективності функціонування сучасних інформаційних систем.

План лекції

1. Основи штучного інтелекту.
2. Напрями штучного інтелекту.
3. Машинне навчання як складова ШІ.

1. Основи штучного інтелекту

Штучний інтелект (ШІ) є одним із найпотужніших інструментів, що здатен впливати на всі сфери людської діяльності: від промисловості та медицини до розваг і повсякденного життя. Його основи почали формуватися задовго до сучасних проривів у машинному навчанні та обробці даних. Розуміння базових принципів ШІ дозволяє не лише краще уявити, як працюють сучасні розумні системи, але й оцінити їхній потенціал для вирішення складних задач, оптимізації процесів та покращення якості життя.

У 1956 році в Дартмутському коледжі (США) Джон Маккарті, фахівець з інформатики та когнітивних наук, ініціював двомісячний семінар для науковців, які займалися дослідженнями нейронних мереж та процесів, пов’язаних із вивченням інтелекту. Під час цього семінару було досягнуто домовленості про введення нової назви для цієї галузі досліджень — «штучний інтелект»

(Artificial Intelligence), що згодом стало офіційною термінологією для цієї наукової дисципліни.

Джон Маккарті, пояснюючи штучний інтелект в одному зі своїх інтерв'ю, назвав його наукою і розробкою інтелектуальних машин та систем, зокрема комп'ютерних програм, що покликані допомогти зрозуміти людський інтелект. Він підкреслив, що ці методи не обов'язково мають бути біологічно достовірними. Інтелект, за словами Маккарті, – це здатність виконувати обчислення, які дозволяють досягати певних цілей у реальному світі. При цьому інтелект може проявлятися в різних формах і ступенях як у людей, так і у багатьох тварин та деяких машин.

Посилання на комп'ютери, програми та комп'ютерні науки часто зустрічаються в багатьох визначеннях, які дають дослідники та розробники штучного інтелекту. Наприклад, Джордж Люгер у своїй відомій праці «Штучний інтелект: структури та стратегії вирішення складних проблем» стверджує: «Штучний інтелект можна визначити як галузь комп'ютерної науки, що займається автоматизацією розумної поведінки». Зазвичай проводять таку аналогію: якщо мозок є біологічним засобом, що через процеси в нейронах генерує інтелект, то комп'ютер — це технічний засіб, що за допомогою програм створює штучний інтелект.

Нині в області штучного інтелекту спостерігається конкуренція між двома основними підходами: символьним і конекціоністським. Ці підходи характеризуються наступним чином.

Символьний підхід (Top-Down AI):

Також відомий як семіотичний підхід, цей напрям зосереджений на розробці експертних систем, баз знань і логічних механізмів виведення, які імітують високорівневі когнітивні процеси. Він включає моделювання таких аспектів людського інтелекту, як мислення, мовлення, логічні міркування, емоції та творчість.

Конекціоністський підхід (Bottom-Up AI):

Іноді його називають біологічним підходом, він передбачає дослідження

нейронних мереж і еволюційних обчислень, які відтворюють інтелектуальну поведінку на основі принципів біологічних систем. Цей підхід також спрямований на створення обчислювальних систем, таких як нейрокомп'ютери, що намагаються імітувати структуру і функціональність мозку.

Зараз все більше уваги приділяється другому напряму, особливо після того, як стало можливим застосування глибокого навчання. Проте основні досягнення першого напряму базуються на гіпотезі Physical Symbol System Hypothesis (гіпотеза про фізичну символьну систему), яку запропонували Аллен Ньюелл і Герберт Саймон у 1976 році.

Ця гіпотеза стверджує, що фізична символьна система має як необхідні, так і достатні засоби для здійснення базових інтелектуальних дій. Іншими словами, для виконання осмислених дій обов'язково потрібні символьні обчислення, і можливості таких обчислень є цілком достатніми, щоб ці дії вважалися осмисленими.

Тобто, якщо ми вважаємо, що певна тварина, людина або навіть машина виконує осмислені дії, це означає, що в їх основі лежать певні форми символьних обчислень. Наприклад, ваша кішка, в певному сенсі, можна розглядати як обчислювальну машину, адже її поведінка свідчить про здатність до осмислених дій. Навпаки, якщо комп'ютер спроможний виконувати такі символьні обчислення, то це означає, що на його основі можна створити штучний інтелект.

Сильний Штучний Інтелект (СШІ) – це програма, яка не просто імітує розум, а є розумом у повному розумінні цього слова. Іншими словами, СШІ здатний виконувати будь-які інтелектуальні завдання, які може виконувати людина, досягаючи людського рівня інтелекту або навіть перевершуючи його. Основна мета досліджень у галузі штучного інтелекту полягає саме у створенні СШІ. Такий штучний інтелект наділений людськими характеристиками, включаючи свідомість, чутливість, розум, мудрість, інтуїцію та самосвідомість.

Сильний Штучний Інтелект, також відомий як **Artificial General Intelligence (AGI)**, є гіпотетичною концепцією інтелектуальної системи, здатної

виконувати широкий спектр когнітивних завдань на рівні або навіть вище людського інтелекту. На відміну від вузького штучного інтелекту, який спеціалізується на вирішенні конкретних задач (наприклад, розпізнавання образів чи гра в шахи), AGI володіє загальною універсальністю. Це означає, що він міг би самостійно вчитися, адаптуватися до нових ситуацій та вирішувати різноманітні проблеми без попередньої програмної підготовки.

Сильний ШІ поки що існує лише як теоретична концепція, і жодних реальних систем AGI на сьогодні не створено. Проте дослідники активно обговорюють, як виглядатиме така система, які технології можуть стати її основою та які ризики вона може нести. Наприклад, однією з ідей є створення універсального алгоритму навчання, який міг би працювати в будь-якому середовищі: від наукових досліджень до вирішення побутових завдань. Інший підхід – це вдосконалення сучасних нейронних мереж та використання методів глибокого навчання, щоб досягти високого рівня автономності.

Ще один напрямок, який розглядається на рівні досліджень, це **Artificial Super Intelligence (ASI)** – штучний суперінтелект, який перевершує не лише людський інтелект, а й будь-які розумові здібності, доступні людям. ASI міг би вивчати і вдосконалювати сам себе, перевершуючи навіть найвидатніших фахівців у будь-якій сфері знань. Хоча це питання поки що перебуває на межі наукової фантастики, воно викликає значний інтерес серед фахівців з етики, філософії, права та інформаційної безпеки.

Наприклад, якщо AGI навчиться самостійно проєктувати ефективні джерела енергії або швидше знаходити ліки від хвороб, це може революційно змінити світ. Проте якщо ASI виявиться неконтрольованим чи його цілі не будуть узгоджені з людськими цінностями, це може становити екзистенційну загрозу. Таким чином, розвиток AGI та ASI супроводжується як великими надіями, так і серйозними викликами.

Слабкий Штучний Інтелект, відомий також як **Artificial Narrow Intelligence (ANI)**, є системою, призначеною для виконання конкретних завдань або розв’язання окремих прикладних задач. На відміну від загального штучного

інтелекту, який має широкий спектр інтелектуальних здібностей, слабкий ШІ зосереджується на вузьких, чітко визначених функціях, які не вимагають наявності людських якостей, таких як свідомість, самосвідомість або розуміння.

Приклади слабкого ШІ:

1. **Голосові асистенти:** Системи на зразок Amazon Alexa, Google Assistant чи Apple Siri здатні розпізнавати команди, знаходити інформацію, керувати пристроями «розумного будинку» або запускати програми. Проте вони не мають загального розуміння того, що таке свідомість чи самосприйняття.
2. **Системи рекомендацій:** Netflix або Spotify використовують слабкий ШІ для аналізу уподобань користувачів і пропонування фільмів, серіалів або музичних треків, які можуть їм сподобатися. Ці системи ефективні, але працюють виключно в межах своєї вузької спеціалізації.
3. **Автоматизація бізнес-процесів:** ШІ-системи, що обробляють рахунки, формують звіти чи перевіряють правопис у текстах, дозволяють економити час та зусилля співробітників, але не можуть самостійно вийти за рамки запрограмованих функцій.
4. **Розпізнавання образів та мови:** Наприклад, системи безпеки, які аналізують зображення з камер спостереження, виявляють аномалії чи розпізнають обличчя. Подібні рішення є надзвичайно корисними у сфері безпеки, проте не мають загальної інтелектуальної спроможності.

Гіпотеза слабкого ШІ базується на твердженні, що такі системи можуть демонструвати інтелектуальні можливості (наприклад, розпізнавати мову чи аналізувати дані), проте вони не здатні на справжнє мислення чи самоусвідомлення. Іншими словами, слабкий ШІ є потужним інструментом для виконання специфічних завдань, але не претендує на роль повноцінного інтелектуального агента.

Штучний інтелект (ШІ) є міждисциплінарною галуззю знань, що включає комп’ютерні науки, математику, логіку, лінгвістику, психологію, філософію та нейробіологію. Основна ідея ШІ полягає у створенні систем, здатних

виконувати завдання, які зазвичай потребують людського інтелекту. Це, зокрема, розпізнавання образів, навчання, планування, прийняття рішень і обробка природної мови.

Ключові компоненти штучного інтелекту:

1. **Машинне навчання (ML)**: Основний підхід, який дозволяє комп'ютерам самостійно вчитися на основі даних. Замість прямого програмування системи створюються алгоритми, які вдосконалюються через аналіз даних.

2. **Нейронні мережі та глибоке навчання (DL)**: Натхненні біологією людського мозку, штучні нейронні мережі є моделями, що здатні виявляти приховані закономірності у великих обсягах даних. Глибоке навчання, зокрема, використовує багатошарові нейронні мережі для досягнення високої точності в таких завданнях, як розпізнавання зображень або обробка тексту.

3. **Обробка природної мови (NLP)**: Технології, що дозволяють комп'ютерам розуміти, генерувати та взаємодіяти з людською мовою. Приклади: чат-боти, системи машинного перекладу, голосові помічники.

4. **Розпізнавання образів та відео**: ШІ використовується для аналізу візуальних даних, таких як зображення або відео, і допомагає автоматизувати такі завдання, як розпізнавання облич, медична діагностика за знімками, або контроль за виробничими процесами.

5. **Планування та прийняття рішень**: ШІ здатен моделювати сценарії, оцінювати наслідки різних рішень і обирати оптимальний варіант. Це знаходить застосування в автоматизації логістики, фінансовому аналізі, розробці стратегій у бізнесі тощо.

6. **Робототехніка та автономні системи**: ШІ використовується для створення роботів, здатних взаємодіяти з фізичним світом, адаптуватися до змін і виконувати складні завдання без втручання людини.

Важливі концепції в основах ШІ:

- **Алгоритми та моделі**: ШІ базується на наборах математичних формул та алгоритмів, які обробляють дані і приймають рішення.

- **Навчання з учителем і без учителя**: Це підходи, які визначають, як система отримує дані для навчання. У першому випадку система використовує позначені дані (де відомі правильні відповіді), у другому — вона вивчає структуру даних без прямих вказівок.
- **Гіпотеза фізичної символьної системи**: Ідея, що інтелектуальна діяльність може бути описана в термінах маніпуляції символами, є одним з перших теоретичних підходів у ШІ.
- **Етичні та соціальні аспекти**: Основи ШІ включають розуміння того, як інтелектуальні системи можуть впливати на суспільство, економіку, приватність і безпеку, а також як забезпечити їх використання на благо людства.

Приклади застосування ШІ:

- У медицині: діагностика захворювань на основі аналізу зображень, прогнозування ходу хвороб, створення персоналізованих планів лікування.
- У транспорті: автономні автомобілі, оптимізація маршрутів, моніторинг дорожньої ситуації.
- У фінансах: виявлення шахрайства, автоматизована торгівля акціями, прогнозування ризиків.
- У роздрібній торгівлі: персоналізовані рекомендації, управління запасами, прогнозування попиту.
- У розвагах: генерація контенту, створення інтерактивних ігор, покращення якості відео та аудіо.

Таким чином, основи штучного інтелекту лежать у здатності систем навчатися, адаптуватися і виконувати завдання, що традиційно потребують людського втручання. Від простих алгоритмів до складних нейронних мереж, ШІ продовжує розширювати свої можливості, змінюючи світ навколо нас.

Сьогодні штучний інтелект застосовується у різноманітних сферах: від медицини та освіти до фінансів і розваг. Наприклад, у медицині ШІ використовується для діагностики захворювань, прогнозування їх розвитку, а також для створення персоналізованих планів лікування. У фінансовому секторі

ШІ допомагає виявляти шахрайські операції, прогнозувати ринкові тренди та автоматизувати обробку кредитних заявок.

ШІ також змінює способи взаємодії з технологіями в повсякденному житті. Віртуальні асистенти, такі як Siri або Alexa, можуть виконувати голосові команди, нагадувати про заплановані події та навіть керувати «розумним» будинком. Рекомендаційні системи у стрімінгових сервісах, таких як Netflix чи Spotify, пропонують користувачам контент, який їм найбільше підходить, на основі їхніх вподобань та поведінки [8].

У сфері транспорту автономні автомобілі, які керуються системами ШІ, вже стають реальністю, забезпечуючи безпечніше й ефективніше пересування. Водночас у промисловості та логістиці інтелектуальні системи оптимізують процеси, знижуючи витрати та підвищуючи продуктивність.

Отже, розвиток штучного інтелекту продовжує трансформувати різні аспекти нашого життя, створюючи нові можливості, покращуючи якість послуг і сприяючи загальному прогресу.

2. Напрями штучного інтелекту

Штучний інтелект охоплює широкий спектр напрямів, які активно розвиваються, вдосконалюються і знаходять нові застосування. Основні напрями включають:

1. **Машинне навчання (Machine Learning, ML):**
Це один із найважливіших напрямів ШІ, який дозволяє алгоритмам вчитися на основі даних і покращувати свої результати без прямого програмування. Машинне навчання поділяється на кілька підходів:

- **Навчання з учителем:** коли модель навчається на наборі даних, де кожен приклад має правильну відповідь. Наприклад, класифікація електронної пошти на спам та не спам.

- **Навчання без учителя:** коли модель аналізує дані, не маючи чітко визначених відповідей. Наприклад, кластеризація клієнтів для маркетингових кампаній.
- **Підкріплювальне навчання:** модель навчається шляхом проб і помилок у процесі взаємодії з середовищем. Наприклад, алгоритми, які грають в ігри або керують роботами.

2. **Глибоке навчання (Deep Learning):** Це підхід до машинного навчання, що базується на штучних нейронних мережах з багатьма шарами. Глибоке навчання використовується для вирішення складних завдань, таких як розпізнавання облич на зображеннях, автоматичний переклад текстів і навіть створення творчих контентів, як-от музика чи малюнки. Завдяки глибокому навчанні можливості ШІ значно розширилися в останні роки.

3. **Обробка природної мови (Natural Language Processing, NLP):** Цей напрям спрямований на те, щоб машини могли розуміти, аналізувати та генерувати текст природними мовами. Приклади включають автоматичний переклад (наприклад, Google Translate), чат-боти (як ChatGPT), голосові помічники (Siri, Alexa) і сервіси автоматичної класифікації тексту (наприклад, виявлення негативних відгуків в коментарях).

4. **Розпізнавання зображень і відео (Computer Vision):** Включає методи аналізу візуальної інформації для розпізнавання об'єктів, виявлення осіб, відстеження руху та багато іншого. Застосовується у відеоспостереженні, автомобілях з автономним керуванням, медичних дослідженнях (наприклад, аналіз медичних знімків для діагностики хвороб) і навіть у мистецтві (створення цифрових галерей та анімацій).

5. **Робототехніка:** Інтеграція ШІ в роботів дозволяє створювати автономних агентів, здатних виконувати складні фізичні завдання, адаптуватися до змін середовища та взаємодіяти з людьми. Сучасні приклади включають безпілотні автомобілі, роботи-прибиральники, роботи-маніпулятори на виробництвах та сервісні роботи в готелях.

6. **Планування та оптимізація:** ШІ допомагає знаходити оптимальні шляхи для досягнення цілей, вирішувати завдання розподілу ресурсів, керування логістикою або навіть розробляти складні стратегії в шахах чи бізнесі. Наприклад, компанії використовують алгоритми оптимізації для зменшення витрат на доставку товарів, зниження енергоспоживання в дата-центрах або планування ефективних розкладів виробництва.

7. **Мультиагентні системи:** Це підхід, при якому кілька агентів (програм, роботів, систем) взаємодіють, координують свої дії або навіть конкурують для досягнення певної мети. Застосування включає розумні транспортні системи, де декілька автономних автомобілів узгоджують свої маршрути, або енергетичні системи, де різні генератори і споживачі електроенергії автоматично оптимізують свої дії.

8. **Інтерфейси «людина – комп'ютер»:** Розробка інтуїтивних способів взаємодії між користувачами та інтелектуальними системами, що включає мову, жести, емоції. Приклади: Голосові помічники (Siri, Alexa, Google Assistant). Системи розпізнавання жестів. Емоційно чутливі ШІ, які аналізують настрій користувача.

Нові інтерфейси, орієнтовані на комфорт та природність, розвиваються з метою зробити взаємодію з технологіями більш зручною та ефективною. Наприклад, голосові помічники сьогодні вже не лише розпізнають слова, а й враховують інтонацію, щоб краще зрозуміти контекст запиту. Системи розпізнавання жестів активно використовуються у віртуальній і доповненій реальності, дозволяючи користувачам природно керувати віртуальними об'єктами без додаткових пристроїв.

Емоційно чутливі ШІ стають усе більш актуальними, особливо у сфері обслуговування клієнтів та освіти. Наприклад, програми для онлайн-навчання можуть адаптуватися до емоційного стану учня, пропонуючи більше підтримки чи змінюючи підхід до подачі матеріалу. У сфері медицини такі системи допомагають лікарям краще зрозуміти психоемоційний стан пацієнтів, що може сприяти вчасному виявленню тривожних розладів або депресії.

Таким чином, розвиток інтерфейсів «людина – комп’ютер» сприяє створенню більш природних і зручних способів взаємодії з технологіями, що дозволяє ефективніше використовувати потенціал сучасних інтелектуальних систем у різних галузях.

Найбільш популярні сьогодні напрями ШІ, що активно розвиваються наведені на рис. 9.1.

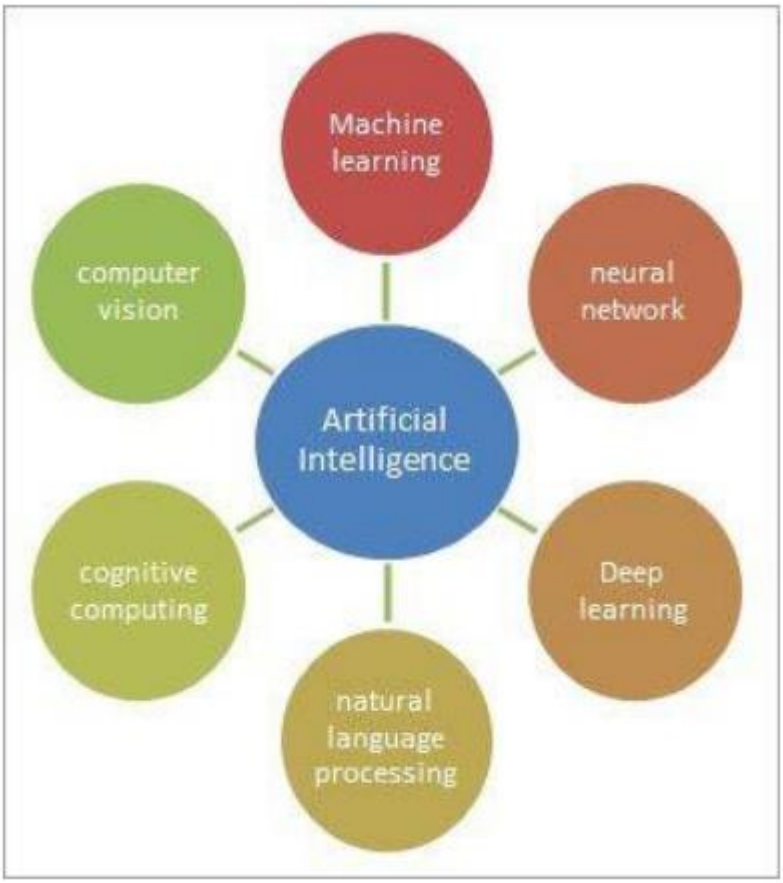

Рис.9.1. Технологічні напрями штучного інтелекту

Зображена на Рис.9.1. сукупність напрямів штучного інтелекту (machine learning, neural network, deep learning, natural language processing, cognitive computing, computer vision) підкреслює багатовимірність та взаємопов’язаність галузі ШІ. Кожен із цих напрямів вирішує специфічні завдання: від розпізнавання образів і обробки природної мови до глибинного аналізу даних та імітації людських когнітивних функцій. Об’єднання та взаємодія цих технологій створюють цілісні системи, здатні навчатися, приймати рішення, адаптуватися до нових умов і, зрештою, автоматизувати складні процеси. Така

синергія забезпечує постійний прогрес штучного інтелекту, відкриваючи нові можливості в науці, бізнесі та повсякденному житті.

Додаткові напрями та перспективи: Завдяки розвитку обчислювальних потужностей і нових підходів у науці про дані, з'являються нові напрями, такі як:

- **Штучний суперінтелект (Artificial Super Intelligence, ASI):** хоча це ще теоретичний концепт, його дослідження зосереджені на тому, як створити ШІ, що перевершує людський інтелект.
- **Етичні аспекти та безпека ШІ:** новий напрям, що займається питаннями забезпечення етичного використання ШІ, запобігання його зловживанням та захистом від небажаних наслідків.

Приклади застосування в реальному світі:

- Системи рекомендацій (Netflix, Amazon) покращують досвід користувачів завдяки аналізу їхніх уподобань.
- Автономні транспортні засоби (Tesla, Waymo) роблять перевезення безпечнішими й ефективнішими.
- Медичні системи, які використовують ШІ для виявлення ракових клітин на ранніх стадіях, рятують життя.
- Системи кібербезпеки, що використовують ШІ для автоматичного виявлення та блокування кібератак, захищають важливі дані.

Сучасний ШІ не просто обмежується одним напрямом, він розвивається у багатьох галузях, змінюючи те, як ми живемо, працюємо і взаємодіємо з навколишнім світом. Його напрями – це складна мережа взаємопов'язаних підходів, які в сукупності створюють базу для майбутніх інновацій.

Сьогодні штучний інтелект охоплює такі напрями, як *обробка природної мови, комп'ютерне бачення, машинне навчання, планування, інтелектуальні агенти та робототехніка*. Кожен із цих напрямів не лише виконує свої унікальні завдання, а й взаємодіє з іншими, створюючи комплексні рішення.

Наприклад, у галузі охорони здоров'я комп'ютерне бачення використовується для аналізу медичних знімків, а обробка природної мови

допомагає розробляти віртуальних помічників для лікарів та пацієнтів. У фінансовій сфері машинне навчання дозволяє виявляти шахрайські транзакції, прогнозувати ризики та автоматизувати управління активами. Водночас робототехніка та інтелектуальні агенти інтегруються в індустрію виробництва, підвищуючи точність і продуктивність процесів, які традиційно вимагали ручної роботи [8].

Ця багатогранність і взаємозв'язок напрямів ШІ сприяє створенню екосистем, які адаптуються до швидкозмінних умов сучасного світу. З розвитком технологій і зростанням обчислювальних потужностей ми бачимо, як ШІ починає не лише відповідати на поточні запити, а й формувати нові способи вирішення складних завдань, створюючи підґрунтя для інновацій у найрізноманітніших сферах життя.

3. Машинне навчання як складова ШІ

Машинне навчання (МН) — це один із ключових напрямів штучного інтелекту, що фокусується на розробці алгоритмів, здатних «навчатися» на основі даних. Американський дослідник у галузі ШІ Артур Самуель (Arthur Samuel) увів цей термін у 1959 році, визначивши машинне навчання як «здатність комп'ютера вчитися без чіткого програмування на кожен крок».

Головна мета машинного навчання полягає у тому, щоб на основі вхідних даних передбачити результат або виявити закономірності. Чим більш різноманітними є ці дані, тим легше алгоритму знайти залежності між ними і тим точніший підсумковий результат він може надати.

Приклади:

- **Виявлення спаму в електронній пошті:** алгоритм аналізує тисячі повідомлень і вчиться розпізнавати шаблони, характерні для спаму (наприклад, ключові слова чи підозрілі посилання).
- **Рекомендаційні системи:** сервіси на кшталт Netflix чи YouTube використовують машинне навчання для пропонування користувачам відео чи

фільмів, які, ймовірно, їх зацікавлять, ґрунтуючись на їхній попередній активності.

- **Фінансовий аналіз:** алгоритми можуть прогнозувати ризики кредитування, курс акцій або рух криптовалют, опираючись на історичні дані та поточні тенденції.

На наведеній (рис. 9.2) схемі зображені основні складові машинного навчання:

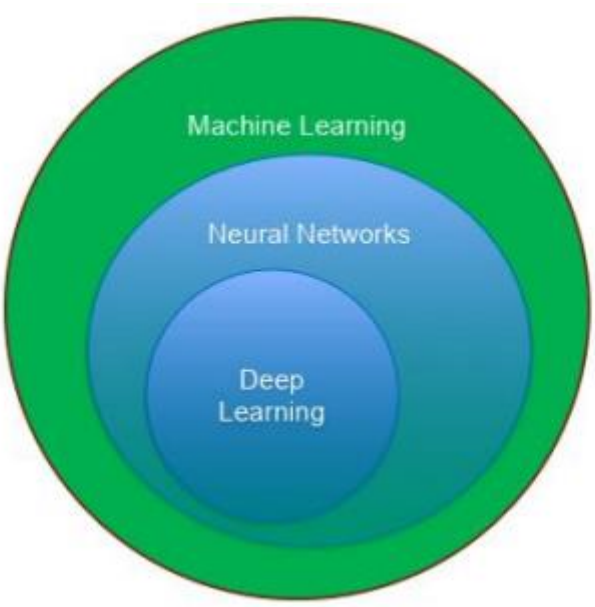

Рис. 9.2. Структура машинного навчання

1. **Збір даних:** якість та різноманітність зібраної інформації безпосередньо впливають на результат.

2. **Підготовка даних:** очищення, перетворення і нормалізація даних, щоб алгоритм міг коректно їх обробляти.

3. **Вибір та навчання моделі:** обрання відповідного алгоритму (наприклад, лінійна регресія, дерева рішень або нейронні мережі) та його навчання на зібраних даних.

4. **Оцінка та валідація:** перевірка точності та надійності моделі, аналіз її помилок і відсутніх випадків.

5. **Впровадження у виробниче середовище:** використання моделі для реальних завдань, моніторинг та регулярне оновлення при надходженні нових даних.

Основними складовими машинного навчання, які часто називають «три кити», є дані, ознаки та алгоритми.

Дані (Data):

Для вирішення конкретних завдань, таких як виявлення спаму, передбачення курсів акцій або аналіз інтересів користувача, потрібні релевантні дані. Наприклад, для виявлення спаму необхідні приклади спам-листів, для прогнозування цін акцій — історичні дані про курси, а для розуміння інтересів користувача — інформація про його лайки, пости чи коментарі. Чим більше даних, тим кращими стають результати моделювання. Дані можна збирати вручну — що забезпечує високу точність, але обмежене їх кількість, або автоматично, що дозволяє накопичити великі обсяги, але може містити помилки. Великі компанії, такі як Google, навіть використовують своїх користувачів для безкоштовної розмітки даних, адже за якісними наборами даних (datasets) завжди йде велике полювання.

Приклад: Для тренування алгоритмів фільтрації спаму система може автоматично збирати мільйони електронних листів, що містять як спам, так і легітимні повідомлення, а потім використовувати ручну розмітку для покращення точності класифікації.

Ознаки (Features):

Ознаки — це конкретні характеристики даних, з якими працює алгоритм. Вони можуть бути дуже різними: від пробігу автомобіля, статі користувача чи ціни акцій до частоти появи певних слів у тексті. Найкраще, коли дані організовані у вигляді таблиць, де назви колонок слугують ознаками. Проте надлишок ознак може зменшувати продуктивність моделі, адже обробка великої кількості ознак може бути повільною і неефективною. Водночас, коли розробник самостійно відбирає лише «правильні» ознаки, це може внести суб’єктивізм у модель, що також негативно позначається на її ефективності.

Приклад: У системі прогнозування кредитного ризику ознаками можуть бути вік позичальника, дохід, кредитна історія, кількість кредитів, зафіксованих у минулому. Неправильний або надмірний відбір ознак може призвести до

поганої роботи моделі, тоді як ретельний відбір допомагає підвищити точність прогнозів.

Алгоритми (Algorithms):

Для розв’язання однієї і тієї ж задачі існує багато алгоритмічних підходів, і саме від вибору методу залежить точність, швидкість роботи та розмір отриманої моделі. Кожна підгалузь машинного навчання має свої специфічні алгоритми: у розпізнаванні образів це можуть бути алгоритми класифікації, у роботі з текстами — методи обробки природної мови, у нейронних мережах — алгоритми навчання багатошарових моделей.

Приклад: Для розпізнавання рукописного тексту можна використовувати конволюційні нейронні мережі (CNN), які спеціально розроблені для аналізу зображень. Різні алгоритми, такі як SVM (машини опорних векторів) або метод головних компонент (PCA), можуть бути застосовані для вирішення тих же завдань, але з різною точністю та швидкістю роботи.

Від якості та кількості даних, правильного відбору ознак до ефективного вибору алгоритму залежить успішність моделі. Цей комплекс підходів дозволяє створювати системи, здатні вирішувати складні задачі автоматизації, прогнозування та класифікації, що знаходять застосування у різних галузях — від фінансових технологій до медицини і транспортних систем. В подальшому вдосконалення кожного з цих компонентів сприятиме ще більшій точності, швидкості та адаптивності систем машинного навчання.

Однією з найважливіших сфер застосування машинного навчання є розпізнавання різноманітних образів, що охоплює як візуальні, так і аудіо, ситуаційні та стано-орієнтовані задачі. Це дозволяє системам не лише "бачити" та "чути", а й розуміти контекст подій і визначати поточний стан середовища.

Типи машинного навчання для розпізнавання образів:

- **Image Recognition (Розпізнавання зображень):** Системи, що здатні аналізувати та ідентифікувати об'єкти на зображеннях. Наприклад, у медичній діагностиці нейронні мережі можуть розпізнавати ознаки захворювань на основі

рентгенівських знімків або МРТ, а в безпеці — розпізнавати обличчя для ідентифікації осіб.

- **Speech Recognition (Розпізнавання мовлення):** Технології, які дозволяють системам розуміти та обробляти людське мовлення. Голосові асистенти, такі як Siri, Alexa чи Google Assistant, використовують розпізнавання мовлення для перетворення голосових команд у текст і подальшого аналізу запитів.
- **Situation Recognition (Розпізнавання ситуацій):** Це здатність систем аналізувати комплекс інформації з різних джерел (наприклад, зображень, аудіо, сенсорних даних) для визначення контексту ситуації. Наприклад, в автономних транспортних засобах системи розпізнавання ситуацій допомагають визначати дорожню обстановку, прогнозувати поведінку інших учасників руху та приймати рішення в режимі реального часу.
- **State Recognition (Розпізнавання станів):** Цей тип системи зосереджується на аналізі даних для визначення поточного стану об’єкта або середовища. Наприклад, в промисловості системи моніторингу можуть аналізувати дані з сенсорів для виявлення стану обладнання, виявлення аномалій або передбачення потенційних поломок.

Приклади застосування:

- **Image Recognition:** Автоматизовані системи виявлення дефектів на виробничих лініях, де камери фіксують зображення продукції, а алгоритми виявляють відхилення від стандарту, що дозволяє зменшити кількість браку та покращити якість продукції.
- **Speech Recognition:** Системи для автоматичного транскрибування дзвінків у кол-центрах, які дозволяють оперативно аналізувати розмови та підвищувати рівень обслуговування клієнтів.
- **Situation Recognition:** У безпілотних автомобілях системи аналізують дані з камер і сенсорів, щоб розпізнати дорожні умови, поведінку інших транспортних засобів та пішоходів, забезпечуючи безпечне керування автомобілем.

- **State Recognition:** У енергетичних мережах системи розпізнавання станів використовуються для моніторингу технічного стану обладнання, що дозволяє вчасно виявляти можливі несправності та проводити планове технічне обслуговування [8].

Таким чином, основи штучного інтелекту та машинного навчання лежать у здатності систем аналізувати дані, навчатися та адаптуватися для виконання завдань, які традиційно вимагали людського інтелекту. Розпізнавання зображень, мовлення, ситуацій та станів є яскравими прикладами практичного застосування цих технологій, що змінюють спосіб, у який ми взаємодіємо з технологічними системами в різних галузях. Від автономних автомобілів до медичних діагностичних систем, можливості машинного навчання постійно розширюються, створюючи фундамент для майбутніх інновацій.

Контрольні питання

1. Що таке штучний інтелект і які його основні компоненти та принципи?
2. Як історично розвивалися основи ШІ і чому вони є важливими для сучасних інформаційних систем?
3. Які основні напрями штучного інтелекту (наприклад, машинне навчання, обробка природної мови, розпізнавання зображень, робототехніка тощо) ви можете виділити, і чим вони відрізняються один від одного?
4. Які практичні приклади застосування різних напрямів ШІ можна навести у медицині, фінансах, освіті та інших галузях?
5. Що таке машинне навчання і як воно інтегрується в систему штучного інтелекту?
6. Які типи машинного навчання існують (з учителем, без учителя, підкріплювальне) і для яких задач вони використовуються?
7. Як алгоритми машинного навчання допомагають автоматизувати розв’язання завдань, які традиційно потребували людського інтелекту?
8. Які переваги та виклики пов’язані з впровадженням машинного навчання в сучасні інформаційні системи?

Тема 10.
ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ТЕЛЕКОМУНІКАЦІЯХ

Метою є ознайомлення слухачів з сучасними тенденціями використання технологій штучного інтелекту (ШІ) в галузі телекомунікацій, демонстрація практичних прикладів застосування ШІ для вирішення актуальних завдань, а також формування розуміння потенціалу та перспектив розвитку цієї сфери. Лекція має на меті розкрити можливості ШІ для оптимізації мереж, покращення якості обслуговування, автоматизації процесів та створення нових послуг у телекомунікаційній галузі.

План лекції:

1. Напрями застосування Штучного інтелекту в телекомунікаціях.
2. Штучний інтелект в покращенні якості обслуговування.
3. Перспективи розвитку та виклики.

1. Напрями застосування ШІ в телекомунікаціях

Штучний інтелект активно змінює сферу телекомунікацій, дозволяючи автоматизувати багато процесів, покращувати якість зв’язку, підвищувати продуктивність мереж та оптимізувати роботу операторів. Сучасні телекомунікаційні компанії впроваджують алгоритми штучного інтелекту для аналізу великих обсягів даних, прогнозування мережевих збоїв, управління трафіком та покращення взаємодії з користувачами. Більшість застосувань ШІ зосереджені на вдосконаленні певних характеристик і параметрів телекомунікаційних систем. До основних напрямів його використання відносяться:

- **Автоматизація управління мережею** – використання ШІ для моніторингу трафіку, оптимізації навантаження на мережу та автоматичного усунення несправностей.

- **Прогнозування відмов обладнання** – машинне навчання допомагає аналізувати роботу пристроїв та прогнозувати можливі несправності, що дозволяє запобігати аварійним ситуаціям.
- **Покращення якості обслуговування клієнтів** – чат-боти та віртуальні асистенти на базі штучного інтелекту можуть швидко відповідати на запити користувачів, допомагати у вирішенні технічних проблем та рекомендувати послуги.
- **Оптимізація маршрутизації трафіку** – інтелектуальні алгоритми автоматично знаходять найоптимальніші шляхи для передавання даних, що зменшує затримки та покращує стабільність з'єднання.
- **Виявлення та запобігання кібератакам** – штучний інтелект аналізує поведінку в мережі та знаходить аномалії, які можуть свідчити про потенційні загрози.
- **Інтелектуальне керування 5G-мережами** – ШІ допомагає адаптувати мережі до зміни навантаження, розподіляти ресурси ефективніше та підтримувати високий рівень якості зв'язку.
- **Персоналізація послуг** – аналізуючи поведінку користувачів, штучний інтелект дозволяє операторам пропонувати індивідуальні тарифи та додаткові послуги.
- **Розвиток Інтернету речей (IoT)** – ШІ допомагає ефективно керувати мільйонами підключених пристроїв, забезпечуючи швидку обробку даних і підвищену безпеку.

Завдяки інтеграції штучного інтелекту у сферу телекомунікацій з'являються нові можливості для підвищення ефективності роботи мереж, вдосконалення послуг для користувачів та впровадження інноваційних технологій. У перспективі ШІ стане ще більш важливим компонентом телекомунікаційних систем, допомагаючи адаптувати їх до нових викликів [8].

Сучасні застосування штучного інтелекту в основному спрямовані на вдосконалення конкретних характеристик і параметрів. Це спеціалізовані програми, розроблені для виконання певних завдань, зокрема:

1. Оптимізація параметрів радіозв'язку

На сьогоднішній день методи машинного навчання (ML) активно використовуються для оптимізації потоку даних між базовою станцією (BTS) і мобільними пристроями в телекомунікаційних мережах. Основні параметри радіозв'язку залежать від відстані до користувачів, кількості активних підключень, а також впливу факторів навколишнього середовища.

Одним із ключових аспектів цієї оптимізації є ефективний розподіл доступних ресурсів спектра, що визначає максимальну швидкість передавання даних. Крім того, рівень перешкод впливає на якість зв'язку, тому необхідно координувати радіоресурси між мікро- та макросотами.

Штучний інтелект відіграє важливу роль у цьому процесі, оскільки спеціалізовані алгоритми можуть динамічно визначати, яка частина спектра має бути використана для конкретного користувача, з якими параметрами та на якій частоті. Використання ШІ дозволяє здійснювати гнучке налаштування цих параметрів у реальному часі, що суттєво підвищує ефективність мобільних мереж та забезпечує кращу якість обслуговування користувачів.

2. Оптимізація енергоспоживання в телекомунікаційних мережах

Методи машинного навчання активно застосовуються для підвищення енергоефективності мобільних мереж. Вони допомагають зменшити витрати електроенергії базових станцій за рахунок адаптивного регулювання потужності сигналу та спрямованості антен.

Цей підхід базується на аналізі великого обсягу даних, включаючи метеорологічні умови, кількість активних користувачів та їхнє географічне розташування. На основі цих параметрів алгоритми штучного інтелекту автоматично налаштовують характеристики сигналу, що дозволяє зменшити витрати енергії в періоди низького навантаження, наприклад, вночі, коли споживання трафіку зменшується.

Крім того, адаптивне управління енергоспоживанням дозволяє ефективніше розподіляти навантаження між базовими станціями, забезпечуючи покриття лише в необхідних зонах. Це не тільки знижує експлуатаційні витрати

операторів зв’язку, а й сприяє екологічному використанню ресурсів, зменшуючи вуглецевий слід телекомунікаційної інфраструктури.

3. Оцінка якості передачі сигналу

Якість передачі сигналу в телекомунікаційних мережах є критичним фактором для забезпечення стабільного з’єднання. Штучний інтелект та машинне навчання дозволяють передбачати потенційні проблеми ще до їхнього виникнення, що дає змогу вжити необхідні заходи для мінімізації ризиків втрати сигналу або деградації його якості.

В оптоволоконних мережах сигнал може спотворюватися або перериватися через фізичні пошкодження кабелів, вплив зовнішніх електромагнітних завад, застаріле обладнання чи надмірне навантаження. Алгоритми машинного навчання аналізують різні параметри, такі як довжина кабелю, рівень затухання сигналу, щільність інших сигналів у волоконному середовищі, а також вік і технічний стан обладнання. На основі цих даних ШІ прогнозує оптимальний маршрут передачі, забезпечуючи найкращу якість зв’язку.

У бездротових мережах штучний інтелект може оцінювати рівень шумів та інтерференцій, визначаючи оптимальні параметри корекції помилок або необхідність повторної передачі даних. Завдяки цьому можливо значно покращити продуктивність мережі, зменшити затримки та знизити навантаження на інфраструктуру.

Експертні системи, розподілений штучний інтелект (Distributed AI) та методи машинного навчання активно впроваджуються в телекомунікаційній сфері. У перспективі їхнє застосування допоможе операторам зв’язку більш ефективно використовувати доступні ресурси, зменшити експлуатаційні витрати та покращити якість послуг для користувачів.

Програми штучного інтелекту в телекомунікаціях

Штучний інтелект (ШІ) має широкий спектр застосування в телекомунікаційній галузі, сприяючи автоматизації процесів, підвищенню

якості послуг та оптимізації роботи мереж. Одним із найпоширеніших підходів є Closed-loop scenario (сценарій замкнутого циклу).

Сценарій замкнутого циклу (Closed-loop scenario)

Цей сценарій передбачає повністю автоматизоване прийняття рішень системою ШІ без втручання людини. Алгоритми аналізують вхідні дані, визначають найкращий курс дій та реалізують їх у режимі реального часу. Єдиною можливою взаємодією людини із системою є можливість вимкнення або перезапуску (Рис. 10.1).

Приклади застосування:

- **Розпізнавання мовлення та голосові асистенти:** Голосові помічники, такі як **Google Assistant, Siri, Alexa**, автоматично обробляють голосові запити та виконують команди без необхідності підтвердження користувачем.
- **Автоматичне налаштування мережі:** Системи моніторингу мережевої інфраструктури можуть самостійно змінювати частотні параметри, розподіляти навантаження між вузлами мережі та усувати збої. Наприклад, **Cisco AI Network Analytics** використовує ШІ для покращення продуктивності мережевих з’єднань.
- **Автоматичне керування трафіком:** У 5G-мережах ШІ оптимізує трафік відповідно до змін у навантаженні та умовах середовища, забезпечуючи безперервність зв’язку та мінімізуючи затримки.
- **Системи виявлення шахрайства:** ШІ-алгоритми аналізують шаблони дзвінків та інтернет-трафік, автоматично блокуючи підозрілу активність, наприклад, шахрайські виклики або спроби отримати доступ до персональних даних.

Сценарії замкнутого циклу дозволяють значно підвищити ефективність телекомунікаційних систем, зменшити людський фактор та автоматизувати ключові процеси, що сприяє розвитку технологічно просунутих мереж нового покоління.

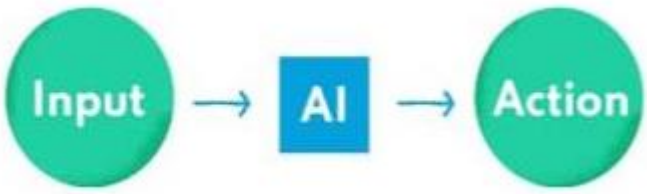

Рисунок 10.1 – Сценарій замкнутого циклу

Сценарій відкритого циклу (Open-loop scenario)

Сценарій відкритого циклу передбачає, що система штучного інтелекту (ШІ) відіграє допоміжну роль у прийнятті рішень (Рис.10.2). Вона аналізує дані, формує висновки та пропонує рекомендації, проте кінцеве рішення залишається за людиною. У цьому випадку користувач може прийняти, відхилити або доповнити рекомендації ШІ на основі додаткової інформації або власного досвіду.

Приклади застосування:

- **Експертні системи в медицині:** ШІ аналізує симптоми пацієнта, історію хвороб і лабораторні тести, пропонуючи можливі діагнози. Наприклад, **IBM Watson Health** допомагає лікарям у діагностуванні онкологічних захворювань, але остаточне рішення про лікування ухвалює спеціаліст.
- **Системи підтримки водіїв (ADAS – Advanced Driver Assistance Systems):** ШІ аналізує дорожню ситуацію, попереджає водія про можливі загрози (наприклад, наближення до перешкоди або перевищення швидкості), але не втручається у керування транспортним засобом.
- **Кібербезпека та антивірусні системи:** Аналітичні ШІ-системи виявляють потенційні загрози у мережевому трафіку або файлах, але остаточне рішення про блокування або видалення загрози приймає фахівець.
- **Фінансові рекомендаційні системи:** Алгоритми аналізують ринок та пропонують інвесторам варіанти купівлі-продажу акцій, але трейдер сам ухвалює рішення. Наприклад, **Bloomberg Terminal** надає фінансову аналітику та прогнозування, допомагаючи інвесторам приймати обґрунтовані рішення.

Таким чином, **сценарій відкритого циклу** ідеально підходить для сфер, де потрібен баланс між автоматизацією та людським контролем, особливо у **медицині, фінансах, транспорті та безпеці**.

Рисунок 10.2 – Сценарій відкритого циклу

Сценарій замкнутого циклу, обмеженого правилами (Rule-Constrained Closed-Loop Scenario)

У **rule-constrained closed-loop scenario** система штучного інтелекту (ШІ) має автономний контроль над процесами, однак її дії обмежені заздалегідь визначеними правилами та умовами (Рис.10.3). Це означає, що система може виконувати завдання без участі людини, але у випадку виникнення непередбаченої або критичної ситуації – вимикається або переходить у безпечний режим.

*Приклади застосування***:**

- **Автономні транспортні засоби:**
  Автомобілі з автопілотом, такі як **Tesla, Waymo, Cruise**, працюють у **замкнутому циклі** та можуть самостійно приймати рішення щодо швидкості, напрямку руху та гальмування. Проте вони мають **вбудовані правила безпеки**: якщо система виявляє загрозу (наприклад, перешкоду на дорозі або складні погодні умови), вона активує **аварійне гальмування або передає керування водієві**.
- **Промислова робототехніка:**
  Роботи, що працюють на заводах **(наприклад, у Tesla, BMW, Amazon Robotics)**, виконують операції (збирання деталей, пакування, переміщення об’єктів) у повністю автономному режимі, але у разі

порушення правил безпеки або відхилення від допустимих параметрів – **зупиняються, щоб уникнути аварії**.

- **Автоматизовані фінансові системи:**
  Алгоритмічні торгові системи **(HFT – High-Frequency Trading)** на біржах виконують мільйони угод за секунду. Проте у разі виявлення **аномальних коливань ринку або ризику серйозних фінансових втрат** – такі системи мають вбудовані обмеження, що блокують торги або переводять їх у безпечний режим.
- **Безпілотні дрони та військові системи:**

Дрони, такі як **Skydio X2 або DJI Enterprise**, працюють автономно для моніторингу територій або виконання місій. Проте вони обмежені алгоритмами безпеки, які забороняють політ у заборонених зонах або активують автоматичне повернення додому при низькому заряді батареї.

Рисунок 10.3 – Сценарій, обмежений правилами

Сценарії взаємодії людини та ШІ в телекомунікаціях

Штучний інтелект у телекомунікаційних системах працює за різними сценаріями взаємодії з людьми та іншими ШІ-системами. Два важливі підходи – це "Людина в циклі" (Human in the Loop) та "Кілька систем ШІ в циклі" (Multiple AI Systems in the Loop).

1. Human in the Loop (Людина в циклі)

У цьому сценарії **ШІ може працювати автономно**, проте людина має можливість **втручатися, зупиняти або коригувати дії системи**. Такий підхід застосовується у випадках, коли повна автоматизація небажана або коли рішення ШІ потребують людського контролю.

Приклади використання:

- **Автопілот у транспорті:**

У сучасних автомобілях із системами **ADAS (Advanced Driver Assistance Systems)**, як у **Tesla, BMW, Mercedes**, водій зобов'язаний тримати руки на кермі. ШІ допомагає керувати авто, але в разі непередбачуваної ситуації людина має **взяти керування на себе**.

- **Медичні телекомунікаційні системи:**

У системах дистанційної діагностики **(наприклад, IBM Watson for Oncology)** ШІ аналізує медичні дані, але **остаточний діагноз залишається за лікарем**.

- **Кібербезпека мереж:**

У системах виявлення загроз **(SOC – Security Operations Center)** ШІ виявляє підозрілі активності, але рішення щодо блокування трафіку чи ізоляції серверів приймають **фахівці з безпеки**.

- **Модерація контенту в соціальних мережах:**

ШІ на платформах **Facebook, YouTube, TikTok** знаходить порушення правил (образливий контент, фейки), але остаточне рішення про блокування залишається **за модераторами**.

2. Multiple AI Systems in the Loop (Кілька систем ШІ в циклі)

Цей сценарій передбачає, що **рішення однієї системи ШІ перевіряється та контролюється іншими системами ШІ**. Такий підхід підвищує **надійність і точність** рішень, а також зменшує ризик помилок.

*Приклади використання***:**

- **Телекомунікаційні мережі 5G:**

У мобільних мережах **AI-платформи Ericsson, Huawei, Nokia** автоматично керують трафіком, оптимізують використання спектра, але додаткові системи ШІ перевіряють результати й коригують стратегію маршрутизації.

- **Фінансові та банківські системи:**

У сфері фінансових транзакцій **(наприклад, Visa, Mastercard, PayPal)** ШІ аналізує транзакції на предмет шахрайства, але його рішення перевіряє **інший ШІ, який використовує більш глибокі методи аналізу**.

Автономні бойові системи:

У військовій сфері **ШІ керує дронами та безпілотними системами**, але вищі рівні ШІ аналізують їхні дії, визначаючи ризики та приймаючи стратегічні рішення.

Обидва сценарії відіграють ключову роль у розвитку інтелектуальних телекомунікаційних систем. "Людина в циклі" потрібна там, де рішення ШІ мають критичний вплив і потребують людського контролю. Натомість "Кілька ШІ в циклі" використовується у високонавантажених системах, де важливо забезпечити багаторівневу перевірку та мінімізувати ризики.

Ці підходи стають основою для майбутніх інновацій у телекомунікаціях, безпеці та автономних системах.

Штучний інтелект (ШІ) відіграє все більш значущу роль у сфері телекомунікацій, сприяючи підвищенню ефективності мереж, оптимізації бізнес-процесів та покращенню якості послуг для користувачів. Використання AI-технологій дозволяє операторам зв’язку впроваджувати передові підходи до управління мережами, аналізу великих обсягів даних та забезпечення кібербезпеки.

Одним із ключових напрямів застосування ШІ є оптимізація параметрів радіосигналу, де алгоритми машинного навчання допомагають налаштовувати частотні спектри, зменшувати інтерференцію та підвищувати якість передачі даних. Управління енергоспоживанням за допомогою інтелектуальних систем дозволяє значно скоротити витрати електроенергії, автоматично регулюючи потужність антен відповідно до попиту.

Також важливим напрямом є оцінка якості передачі сигналу, де алгоритми ШІ прогнозують можливі збої та оптимізують маршрути трафіку, забезпечуючи стабільність з’єднання. Крім того, експертні системи та методи машинного навчання активно застосовуються для виявлення шахрайства, прогнозування попиту на послуги, автоматичного обслуговування клієнтів через чат-боти та віртуальних помічників.

ШІ інтегрується у телекомунікаційні системи за допомогою різних сценаріїв, включаючи замкнуті та відкриті цикли управління, коли рішення

приймаються як автоматично, так і за участі людини. Особливу роль відіграють **автономні системи**, здатні самостійно налаштовувати та оптимізувати роботу мереж, що є важливим кроком у розвитку інтелектуальних телекомунікаційних технологій [8].

З огляду на стрімкий розвиток AI-рішень у сфері телекомунікацій, прогнозується, що до 2026 року впровадження ШІ принесе багатомільярдні прибутки компаніям цієї галузі. Подальший розвиток технологій відкриє ще більше можливостей для автоматизації, покращення якості зв'язку та персоналізації телекомунікаційних послуг, що в перспективі змінить не лише бізнес-моделі операторів, а й сам підхід до управління інформаційними потоками.

2. Штучний інтелект в покращенні якості обслуговування

Штучний інтелект (ШІ) відіграє все більш важливу роль у покращенні якості обслуговування в різних галузях, зокрема й у телекомунікаціях. Ось деякі ключові аспекти застосування ШІ в цій сфері:

1. Чат-боти та віртуальні асистенти:

- ШІ-чат-боти здатні надавати швидкі та ефективні відповіді на поширені запитання клієнтів, зменшуючи час очікування та навантаження на операторів.
- Віртуальні асистенти можуть допомагати клієнтам у вирішенні складніших проблем, надаючи персоналізовану підтримку.

2. Персоналізація послуг:

- ШІ може аналізувати дані про клієнтів (історію дзвінків, використання послуг, демографічні дані) для створення персоналізованих пропозицій та рекомендацій.
- Це дозволяє компаніям надавати більш релевантні послуги та підвищувати задоволеність клієнтів.

3. Виявлення та запобігання шахрайству:

- ШІ може виявляти підозрілу активність та запобігати шахрайським діям, захищаючи клієнтів від фінансових втрат.
- Алгоритми машинного навчання здатні аналізувати великі обсяги даних та виявляти аномалії, які можуть свідчити про шахрайство.

4. Аналіз емоцій клієнтів:

- ШІ може аналізувати голосові та текстові дані для визначення емоційного стану клієнтів.
- Це дозволяє компаніям реагувати на негативні емоції та покращувати якість обслуговування.

5. Автоматизація обслуговування:

- ШІ може автоматизувати рутинні завдання, такі як обробка платежів, зміна тарифних планів та надання технічної підтримки.
- Це дозволяє компаніям скоротити витрати та підвищити ефективність обслуговування.

Переваги використання ШІ в обслуговуванні:

- Підвищення швидкості та ефективності обслуговування.
- Персоналізація послуг та підвищення задоволеності клієнтів.
- Зменшення витрат на обслуговування.
- Підвищення безпеки та запобігання шахрайству.
- Аналіз великого обсягу даних.

6. Проактивна підтримка та прогнозування проблем:

- ШІ може аналізувати дані з мережі та пристроїв клієнтів для виявлення потенційних проблем до того, як вони вплинуть на якість обслуговування.
- Наприклад, ШІ може прогнозувати перевантаження мережі та автоматично перерозподіляти ресурси, щоб уникнути збоїв.
- Також, ШІ може виявляти проблеми з обладнанням клієнтів (наприклад, слабкий сигнал Wi-Fi) та пропонувати їм рішення до того, як вони звернуться до служби підтримки.

7. Голосові інтерфейси та обробка природної мови (NLP):

- Голосові інтерфейси, керовані ШІ, дозволяють клієнтам отримувати підтримку за допомогою голосових команд, що робить обслуговування більш зручним та доступним.
- NLP дозволяє ШІ розуміти та інтерпретувати людську мову, що робить спілкування з чат-ботами та віртуальними асистентами більш природним та ефективним.

8. Оптимізація роботи контакт-центрів:

- ШІ може автоматично розподіляти дзвінки між операторами на основі їхньої кваліфікації та доступності, що скорочує час очікування та покращує якість обслуговування.
- ШІ може надавати операторам підказки та рекомендації в реальному часі, що допомагає їм швидше та ефективніше вирішувати проблеми клієнтів.
- ШІ може аналізувати записи дзвінків для оцінки якості обслуговування та виявлення областей для покращення.

9. Розвиток самообслуговування:

- ШІ може допомогти клієнтам самостійно вирішувати проблеми за допомогою інтерактивних посібників, відеоінструкцій та баз знань.
- Це дозволяє клієнтам отримувати швидкі відповіді на свої запитання без необхідності звертатися до служби підтримки.

10. Етичні аспекти та виклики:

- Важливо враховувати етичні аспекти використання ШІ в обслуговуванні, такі як конфіденційність даних та потенційна упередженість алгоритмів.
- Необхідно забезпечити прозорість та зрозумілість роботи ШІ-систем для клієнтів.
- Важливо постійно навчати та адаптувати ШІ-системи, щоб вони відповідали змінам у потребах клієнтів та технологічному прогресі.

Використання штучного інтелекту в телекомунікаціях не лише покращує якість обслуговування, але й відкриває нові можливості для інновацій та розвитку. Компанії, які успішно впроваджують ШІ, зможуть забезпечити своїм

клієнтам більш якісне та персоналізоване обслуговування, що стане ключовим фактором конкурентоспроможності в майбутньому.

Отже, інтеграція штучного інтелекту в телекомунікаційні системи не обмежується лише окремими функціями, а передбачає створення цілісної екосистеми, де ШІ стає невід'ємною частиною кожного етапу взаємодії з клієнтом. Це включає в себе не тільки автоматизацію рутинних завдань, але й передбачення потреб клієнтів, персоналізацію досвіду та проактивне вирішення потенційних проблем.

Одним з ключових аспектів є створення інтелектуальних платформ, які об'єднують дані з різних джерел - від історії дзвінків та даних про використання послуг до інформації з соціальних мереж та відгуків клієнтів. Ці платформи дозволяють ШІ аналізувати великі обсяги даних в реальному часі та надавати операторам та клієнтам персоналізовану підтримку. Наприклад, коли клієнт телефонує до служби підтримки, ШІ може миттєво ідентифікувати його, переглянути історію взаємодії та надати оператору підказки щодо вирішення проблеми.

Крім того, ШІ може використовуватися для створення інтелектуальних систем самообслуговування, які дозволяють клієнтам самостійно вирішувати проблеми без необхідності звертатися до служби підтримки. Це може включати в себе інтерактивні посібники, відеоінструкції, бази знань та віртуальних асистентів, які можуть надавати персоналізовані рекомендації.

Важливим аспектом є також використання ШІ для моніторингу та оптимізації роботи мережі. ШІ може аналізувати дані про трафік, стан обладнання та інші параметри для виявлення потенційних проблем та автоматичного перерозподілу ресурсів. Це дозволяє забезпечити безперебійну роботу мережі та високу якість послуг.

Однак, впровадження ШІ в телекомунікаційні системи також пов'язане з певними викликами. Одним з них є забезпечення конфіденційності даних клієнтів. Необхідно розробити надійні механізми захисту даних та забезпечити прозорість використання ШІ.

Іншим викликом є необхідність постійного навчання та адаптації ШІ-систем. Технології ШІ швидко розвиваються, і компанії повинні постійно оновлювати свої системи, щоб вони відповідали останнім досягненням.

Незважаючи на ці виклики, потенціал ШІ для покращення якості обслуговування в телекомунікаціях є величезним. Компанії, які успішно впроваджують ШІ, зможуть забезпечити своїм клієнтам більш якісне, персоналізоване та зручне обслуговування, що стане ключовим фактором конкурентоспроможності в майбутньому.

3. Перспективи розвитку та виклики

Перспективи штучного інтелекту в телекомунікаціях

Застосування штучного інтелекту (ШІ) у телекомунікаційній галузі відкриває нові можливості для обробки величезних масивів даних, які раніше залишалися недосяжними для аналітики. Тепер ці дані можна ефективно використовувати для підвищення якості послуг, персоналізації клієнтського досвіду, автоматизації процесів та оптимізації бізнес-операцій.

За результатами аналітичних досліджень, у 2025 року очікується, що ШІ принесе телекомунікаційним компаніям майже 11 мільярдів доларів прибутку. Ця цифра є лише початком глобальної трансформації індустрії, адже впровадження ШІ продовжує розширюватися, змінюючи всі аспекти телекомунікаційного ринку (Рис. 10.4)

Згідно зі звітом «Глобальний штучний інтелект у телекомунікаційному секторі: розмір, стан і прогноз на 2021–2027 рр.», очікується значне зростання інвестицій у ШІ. Основні напрями цього розвитку включають:

- **Розширення аналітичних можливостей для прогнозування поведінки користувачів**
- **Автоматизацію мережевих процесів та управління ресурсами**
- **Зниження операційних витрат завдяки самонавчальним алгоритмам**
- **Покращення обслуговування клієнтів через AI-асистентів та чат-боти**

Ці тенденції вказують на невпинний розвиток телекомунікаційної сфери завдяки впровадженню інтелектуальних систем, які не лише підвищують продуктивність компаній, а й покращують якість обслуговування користувачів.

Розвиток штучного інтелекту у телекомунікаціях є невідворотним трендом, який зумовлює глобальну цифрову трансформацію галузі. Очікується, що штучний інтелект стане основою для наступного покоління телекомунікаційних сервісів, забезпечуючи ефективність, швидкість і якість зв’язку на новому рівні.

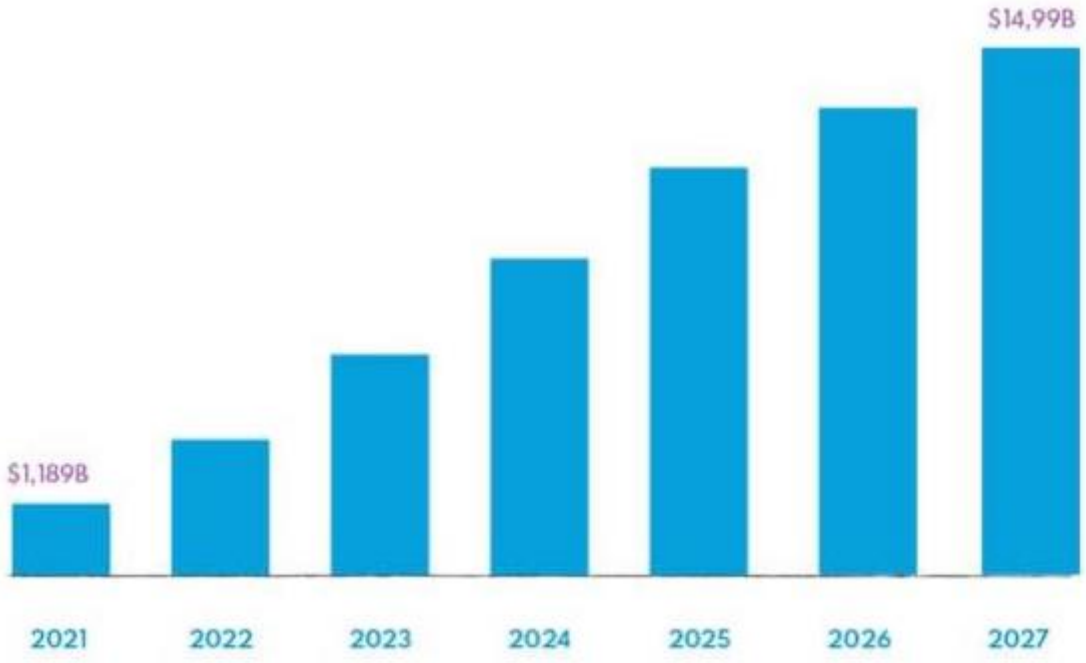

Рисунок 10.4. Прогнозоване зростання світового ринку штучного інтелекту в телекомунікаціях.

1. Автоматизація обслуговування клієнтів

Одним із ключових напрямів використання ШІ є автоматизація роботи контакт-центрів і персоналізоване обслуговування клієнтів.

Віртуальні асистенти та чат-боти

- Інтелектуальні чат-боти, такі як **Google Dialogflow, IBM Watson Assistant, Microsoft Bot Framework**, здатні розпізнавати запити користувачів, проводити діалог і навіть вирішувати певні технічні проблеми без втручання оператора.

- ШІ-асистенти значно скорочують час очікування клієнтів і знижують навантаження на службу підтримки.

Персоналізація обслуговування

- ШІ аналізує історію взаємодії користувача з оператором і прогнозує його потреби.
- Система автоматично пропонує персоналізовані послуги, тарифні плани або спеціальні акції на основі поведінки абонента.

2. Предиктивна аналітика та управління мережею

Штучний інтелект використовується для аналізу великих обсягів даних і прогнозування можливих проблем у мережі.

Передбачення технічних несправностей

- AI-алгоритми можуть запобігати збоям у роботі мережі, аналізуючи дані про навантаження, погодні умови та поведінку обладнання.
- Наприклад, оператори можуть отримувати попередження про можливий вихід із ладу станцій або серверів, що дає змогу оперативно вжити заходів для усунення проблеми.

Розумне управління навантаженням мережі

- Використання AI дозволяє оптимізувати розподіл трафіку, динамічно налаштовуючи параметри радіозв'язку залежно від завантаженості базових станцій.
- Це особливо корисно для 5G-мереж, де необхідна висока швидкість та мінімальні затримки передачі даних.

3. Оптимізація витрат та енергозбереження

ШІ допомагає знижувати енергоспоживання та оптимізувати витрати на технічне обслуговування інфраструктури.

Економія електроенергії

- Завдяки AI-аналізу активності абонентів, **базові станції можуть зменшувати потужність у нічний час** або в регіонах із низьким навантаженням.

- Це дозволяє **значно скоротити витрати операторів** на електроенергію та зменшити негативний вплив на екологію.

Автоматизація моніторингу обладнання

- AI-системи можуть проводити діагностику телекомунікаційних пристроїв у реальному часі, знижуючи витрати на їх обслуговування та ремонт.
- Завдяки передбачуваному технічному обслуговуванню (Predictive Maintenance), компанії можуть зменшити кількість аварійних ситуацій.

4. Покращення кібербезпеки в телекомунікаціях

Збільшення використання цифрових технологій вимагає посилення заходів безпеки, оскільки оператори мобільного зв'язку є потенційними цілями для хакерських атак [8].

Виявлення загроз та запобігання атакам

- ШІ-алгоритми можуть автоматично виявляти аномалії у трафіку, що може свідчити про DDoS-атаку або несанкціоноване втручання в мережу.
- Такі системи аналізують поведінкові патерни користувачів, що дозволяє запобігти шахрайським операціям (наприклад, фішинговим атакам).

Захист персональних даних

- AI використовується для контролю доступу до конфіденційної інформації та запобігання витокам даних.
- Штучний інтелект допомагає автоматично ідентифікувати підозрілі дії та блокувати потенційно небезпечні запити.

Перспективи розвитку ШІ в телекомунікаціях

Інтеграція ШІ у 6G-мережі

- Очікується, що ШІ відіграватиме ключову роль у розвитку 6G-мереж, автоматично оптимізуючи якість зв'язку та безпеку передачі даних.
- Передбачається, що AI-управління трафіком дозволить досягти швидкості до 1 Тб/с.

Розвиток автономних мереж

- У майбутньому телекомунікаційні компанії зможуть створювати повністю автономні мережі, які самостійно виявлятимуть та виправлятимуть помилки без втручання людини.

Збільшення ролі IoT та AI в смарт-містах

- AI-технології будуть використовуватися для оптимізації міських телекомунікаційних інфраструктур, розумного керування трафіком, автоматизації громадських сервісів та покращення якості життя.

Інтеграція штучного інтелекту у телекомунікації змінює правила гри, відкриваючи нові можливості для автоматизації, прогнозування, кібербезпеки та оптимізації бізнес-процесів.

Основні переваги впровадження ШІ в телекомунікаціях:

- Підвищення швидкості обробки даних та оптимізація трафіку
- Автоматизація роботи контакт-центрів та персоналізоване обслуговування
- Зниження енергоспоживання та оптимізація витрат
- Покращення безпеки та боротьба з кіберзлочинністю
- Підготовка до впровадження 6G-мереж та автономних систем

З кожним роком роль ШІ в телекомунікаціях буде зростати, і компанії, які вже сьогодні інвестують у ці технології, отримають значну конкурентну перевагу у майбутньому.

Таким чином, штучний інтелект уже зараз відіграє ключову роль у розвитку телекомунікаційних систем, забезпечуючи оптимізацію мережевого трафіку, покращення якості зв’язку, персоналізацію користувацького досвіду та ефективний моніторинг безпеки. Завдяки алгоритмам машинного навчання та передовим AI-технологіям, телекомунікаційні компанії можуть автоматизувати складні процеси, знижувати експлуатаційні витрати та покращувати ефективність обслуговування клієнтів.

Подальший розвиток штучного інтелекту в галузі телекомунікацій відкриває перспективи для створення автономних мереж, що здатні самостійно адаптуватися до змін у навантаженні, автоматично виявляти та усувати

несправності, а також покращувати кібербезпеку шляхом прогнозування потенційних загроз. Впровадження AI-рішень у телекомунікації дозволить значно підвищити надійність мережевих інфраструктур, скоротити час відгуку в обробці запитів та забезпечити безперебійну передачу даних навіть у найскладніших умовах.

Крім того, розвиток 5G і майбутніх 6G-мереж значною мірою залежатиме від використання штучного інтелекту для динамічного управління спектром, оптимізації навантаження та забезпечення мінімальних затримок у передачі даних. Завдяки впровадженню AI-алгоритмів телекомунікаційні компанії зможуть ефективніше використовувати радіочастотний спектр, що особливо важливо у зв'язку з постійним зростанням кількості підключених пристроїв та інтернету речей (IoT).

У довгостроковій перспективі штучний інтелект стане невід'ємною частиною телекомунікаційних технологій, сприяючи створенню «розумних» мереж, які зможуть не лише реагувати на поточні виклики, а й прогнозувати майбутні потреби користувачів та автоматично підлаштовуватися під змінні умови. Зважаючи на стрімкий розвиток AI-рішень у сфері телекомунікацій, компанії, які вже сьогодні впроваджують ці технології, отримають значну конкурентну перевагу, знижуючи витрати та покращуючи якість надання послуг. В результаті використання штучного інтелекту сприятиме не лише технологічному прогресу в галузі зв'язку, а й створенню більш ефективного, безпечного та інтелектуального цифрового середовища.

Однак, попри всі переваги, впровадження штучного інтелекту в телекомунікаціях супроводжується низкою викликів, які потребують ретельного опрацювання.

1. Кібербезпека та конфіденційність даних

Збільшення обсягів обробки даних і автоматизація мереж призводять до зростання ризиків витоку конфіденційної інформації, кібератак та несанкціонованого доступу. AI-алгоритми потребують значних обсягів даних для навчання, що може створювати загрозу порушення приватності

користувачів. Без належних заходів захисту існує ризик зловживання персональною інформацією або маніпуляції з боку зловмисників.

2. Високі витрати на впровадження

Впровадження AI-рішень у телекомунікаційні мережі вимагає значних інвестицій у розробку, інфраструктуру та адаптацію існуючих технологій. Не всі компанії можуть дозволити собі масштабні фінансові вкладення в модернізацію систем, що створює бар'єри для масового впровадження штучного інтелекту, особливо для малих і середніх операторів зв’язку.

3. Відсутність стандартизації

Попри активний розвиток AI-технологій, на сьогоднішній день немає єдиних стандартів щодо інтеграції ШІ в телекомунікаційні мережі. Відсутність загальноприйнятих норм і протоколів може створювати проблеми сумісності між різними AI-рішеннями, а також ускладнювати взаємодію операторів та постачальників технологій.

4. Потреба в кваліфікованих кадрах

Робота з AI-системами вимагає глибоких знань у сфері машинного навчання, аналізу великих даних і мережевих технологій. Нестача фахівців, які мають досвід у впровадженні штучного інтелекту в телекомунікаціях, є одним із ключових обмежень для розвитку галузі.

5. Надійність та прозорість алгоритмів

AI-системи приймають рішення на основі аналізу великих обсягів даних, однак механізм ухвалення таких рішень не завжди є прозорим. "Чорна скринька" AI може спричинити ситуації, коли оператори не можуть пояснити, чому система працює саме так, а не інакше. Це може викликати проблеми довіри як з боку клієнтів, так і регуляторів.

6. Регуляторні та етичні питання

Штучний інтелект у телекомунікаціях може бути використаний для моніторингу поведінки користувачів, аналізу їхніх звичок та прогнозування дій. Це викликає етичні питання щодо відповідального використання даних, а також

регуляторні виклики щодо дотримання міжнародних норм та законодавства про захист інформації [8].

Попри всі виклики, потенціал штучного інтелекту в телекомунікаціях є величезним. Розвиток AI-технологій дозволяє значно підвищити ефективність мереж, покращити якість обслуговування та створити інноваційні сервіси для користувачів. Однак подолання бар'єрів, пов'язаних із безпекою, регулюванням, прозорістю алгоритмів та доступністю кваліфікованих фахівців, є важливим завданням для майбутнього розвитку галузі. Впровадження чітких стандартів, етичних норм та ефективних заходів захисту стане ключем до успішного використання штучного інтелекту в телекомунікаціях.

Контрольні питання

1. Які основні напрями застосування штучного інтелекту в телекомунікаціях?
2. Як машинне навчання використовується для оптимізації параметрів радіосигналу?
3. У чому полягає роль ШІ у процесі управління енергоспоживанням у телекомунікаційних мережах?
4. Як ШІ допомагає оцінювати якість передачі сигналу?
5. Які методи ШІ застосовуються для виявлення шахрайства у телекомунікаційних системах?
6. Які основні типи сценаріїв використання ШІ в телекомунікаціях?
7. Чим відрізняється сценарій відкритого циклу (Open-loop) від замкнутого циклу (Closed-loop)?
8. Яку роль відіграють автономні системи ШІ у телекомунікаційній індустрії?
9. Які перспективи впровадження ШІ у сфері телекомунікацій прогнозуються на найближчі роки?
10. Які виклики та ризики пов’язані із застосуванням ШІ у телекомунікаційних системах?

Список використаних джерел:

1. Ананьїн В. О. Національна безпека держави в сучасних умовах: монографія / В.О. Ананьїн, О. В. Ананьїн, В. В. Горлинський; за загальною редакцією В.О. Ананьїна. – К.: КПІ ім. Ігоря Сікорського, 2021. – 345 с.
2. Вишня В. Б. Інформаційні системи та технології [Текст]: підручник. – Дніпро: Дніпропетровський державний університет внутрішніх справ, 2021. – 279 с.
3. Гончар О. І. Інформаційні системи: теорія та практика. Навчальний посібник. Харків: Фоліо, 2023. 240 с.
4. Гуржій А. М., Возненко Л. І., Поворознюк Н. І., Самсонов В. В. Основи інформаційних технологій [Текст]: навч. посіб. – Київ : Літера ЛТД, 2023. – 288 с.https://lib.imzo.gov.ua/wa-data/public/site/books2/posibnyky-prof-tech/Osnovy_inform_tehnologiy.pdf
5. Задорожній В. В. Інформаційні системи та технології: Навчальний посібник. Львів: Львівська політехніка, 2019. 276 с.
6. Зачек О. І., Сеник В. В., Магеровська Т. В. та ін. Інформаційні технології [Текст] : навч. посіб. / за ред. О. І. Зачека. – Львів : Львівський державний університет внутрішніх справ, 2022. – 432 с. https://vstup.htek.com.ua/wp-content/uploads/2024/10/20.3-InformacTexnolog.pdf
7. Зінов'єва О. Г., Шаров С. В. Проектування інформаційних систем [Текст]: конспект лекцій. – Запоріжжя: ТДАТУ, 2024. – 148 с. http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/proektuvannja_informacijnyh_system_praktykum.pdf
8. Зінов’єва О.Г. Проектування інформаційних систем: Лабораторний практикум/О. Г. Зінов’єва, С.В. Шаров, Г.В. Гешева. – Запоріжжя, 2023. – 160с.
9. Звенігородський О. С., Зінченко О. В., Чичкарьов Є. А., Кисіль Т. М. Штучний інтелект. Вступний курс [Текст] : навч. посіб. – Київ : ДУТ, 2022. – 193 с.

10. Ігнатенко В. В. Системи управління базами даних: Access / В. В. Ігнатенко, О. С. Яновська. Навчальний посібник – Харків : ХНЕУ, 2020. 280 с.

11. Каштан В.Ю., Іванов Д.В. Конспект лекцій з дисципліни “Бази даних в інформаційних системах”. Для студентів галузі знань 12 “Інформаційні технології” спеціальності 126 “Інформаційні системи та технології”. – Д.: НТУ «ДП», 2021. – 58 с.

12. Климчук О. В. Інформаційні системи і технології в управлінні [Текст]: конспект лекцій. – Вінниця : ДонНУ імені Василя Стуса, 2021. – 160 с. https://r.donnu.edu.ua/bitstream/123456789/1673/1/%D0%9A%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82_%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%97_%D0%86%D0%A1%D0%A2%D0%A3%D0%A1%D0%A1_2021_%D0%B2%D0%B5%D1%80%D1%81%D1%82%D0%BA%D0%B0.pdf

13. Левченко М. П. Основи комп'ютерних систем та технологій. Підручник. Запоріжжя: Видавництво "Прем'єр", 2022. 350 с.

14. Лубко Д.В., Шаров С.В. Методи та системи штучного інтелекту: лабораторний практикум. Запоріжжя, 2023. 162 с.

15. Марченко А.В. Проектування інформаційних систем. Київ, 2016. 90 с.

16. Мушеник І.М., Марчук Н.А. Курс лекцій з навчальної дисципліни «Інформаційні технології та вища математика» для здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю 201 «Агрономія» / І. М. Мушеник., Н.А. МАРЧУК. Кам’янець-Подільський: ЗВО «ПДУ», 2024. 246 с. http://188.190.43.194:7980/jspui/handle/123456789/14035

17. Мушеник І.М. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни «Основи інформаційних технологій» для здобувачів першого (бакалаврського) рівня вищої освіти для спеціальності 126 «Інформаційні системи і технології». /І.М. Мушеник, Л.Ю. Збаравська // 2024. 82 с. http://188.190.43.194:7980/jspui/handle/123456789/13163

18. Олевський В. І., Олевська Ю. Б., Соколова Н. О. Конспект лекцій з дисципліни «Моделювання інформаційних систем» [Електронний ресурс] : для

здобувачів ступеня бакалавра спеціальності 126 «Інформаційні системи та технології» / М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Електрон. дані. – Дніпро : НТУ «ДО», 2024. – 499 с.

19. Основи теорії інформації та кодування [Текст] / І. А. Прокопишин, Р. Є. Рикалюк, В. Ф. Чекурін, К. А. Червінка. – Львів : ЛНУ ім. Івана Франка, 2023. – 156 с.

20. Проектування інформаційних систем: загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс] : навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського ; уклад. О. С. Коваленко, Л. М. Добровська. – Електрон. текст. дані (1 файл : 2,02 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2020. – 192 с. https://ela.kpi.ua/server/api/core/bitstreams/c136860d-44cb-4f05-adaf-dcdd20830483/content/

21. Риндюк Д. В., Пешко В. А. Інформаційні технології: конспект лекцій [Текст]: навч. посіб. / Київ: КПІ ім. Ігоря Сікорського, 2022. – 180 с. https://fpk.in.ua/images/biblioteka/2fmb_finansy/Informatsiini_tekhnolohii_lektsii-2022.pdf

22. Соколова Н. О. Бази даних в інформаційних системах [Електронний ресурс] : конспект лекцій з дисципліни «Бази даних в інформаційних системах» для здобувачів ступеня бакалавра освітньо-професійної програми «Інформаційні системи та технології» зі спеціальності 126 «Інформаційні системи та технології» / уклад. Н. О. Соколова ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2024. – 233 с.

23. Соколова Н. О., Коробко О. В. Прикладні інформаційні технології (за професійним спрямуванням) [Електронний ресурс] : навч. посіб. / М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДО», 2024. – 100 с.

24. Ситник Г. П., Орел М. Г. Національна безпека в контексті європейської інтеграції України: підручник / Г. П. Ситник, М. Г. Орел; за ред.

Г. П. Ситника. – К.: Міжрегіональна Академія управління персоналом, 2021. – 372 с.

25. Холодняк Ю.В., Зінов’єва О.Г. Інформаційні технології: практикум. Таврійський державний агротехнологічний університет імені Дмитра Моторного, 2023. 152 с. http://www.tsatu.edu.ua/kn/wp-content/uploads/sites/16/praktykum_informacijni_tehnolohiyi.pdf

26. Шпек В. В. Апаратне забезпечення комп'ютерних систем. Київ: Видавництво "Торсінг", 2022. 295 с.

МУШЕНИК Ірина Миколаївна

«Теорія інформаційних систем»

Курс лекцій
для здобувачів першого (бакалаврського) рівня вищої освіти
за спеціальністю 126 «Інформаційні системи і технології»