



# Бази даних та Інформаційні системи

## 1 Вступ, DF-моделювання

СумДУ, каф. КН  
2020

# Мета курсу

---

- ▶ Уміння робити практичні висновки з тверджень, понять, пов'язаних з базами даних;
- ▶ Розуміння принципів, що лежать в основі реляційної БД;
- ▶ Уміння спроектувати реляційну БД;
- ▶ Уміння працювати з реляційної БД;
- ▶ Базові знання та вміння використовувати мову SQL.

# Поняття БД

---

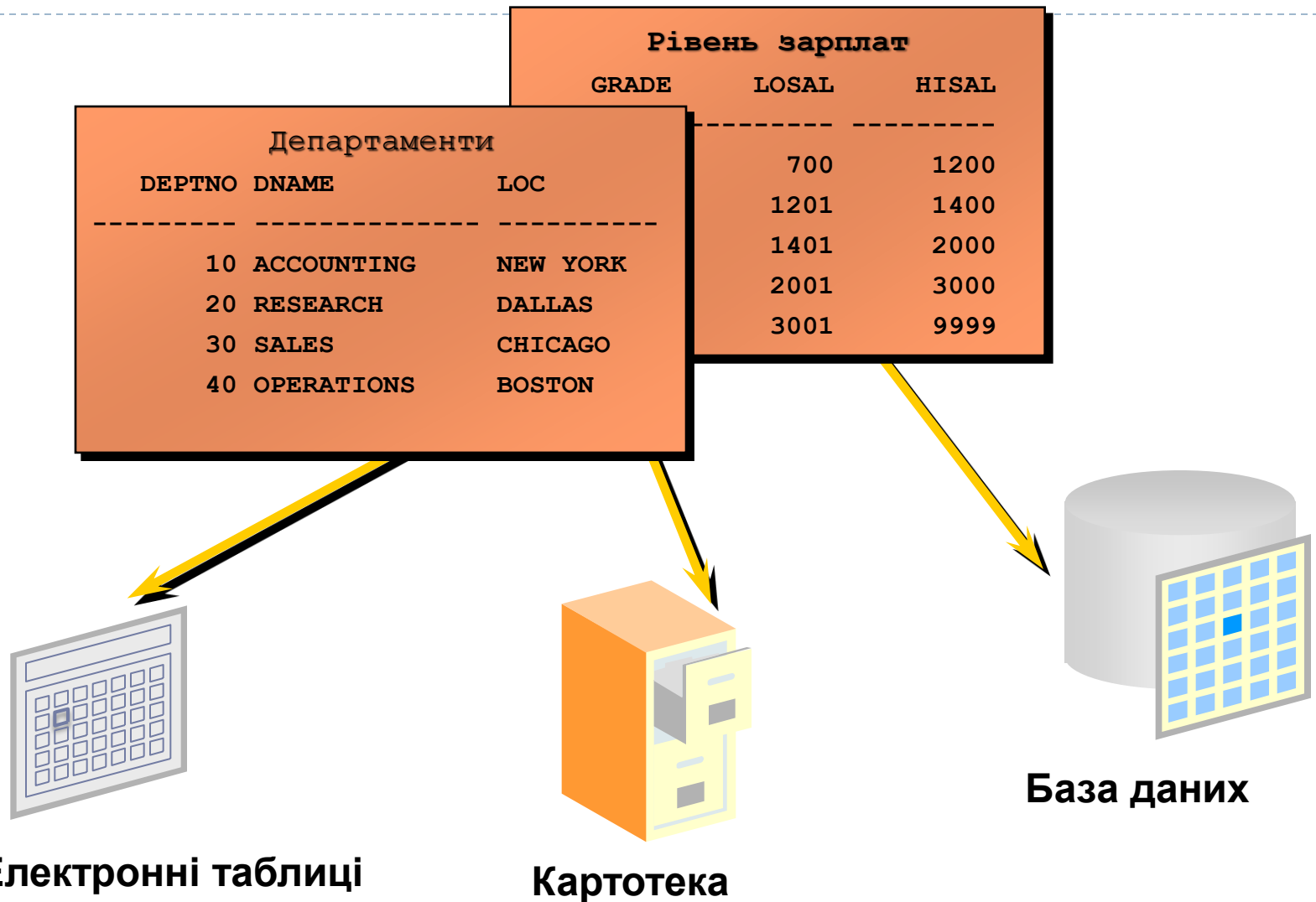
- “A **database** is a structured collection of data.”
  - [Wikipedia.en](#)
- «**База даних** — впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно, та призначені для задоволення інформаційних потреб користувачів»
  - [Wikipedia.ua](#)
- «**База данных** – представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины»
  - [Wikipedia.ru](#)
- —

# Приклади БД

---

- Телефонний довідник;
- Наповнення мобільного телефону:
  - Контакти;
  - СМС-ки;
  - Список пропущених дзвінків;
- Вміст браузера:
  - Список закладок;
  - Кеш завантажених сторінок;
  - Cookies;
- Google Maps;
- Список покупок.

# Зберігання даних на різних носіях





# Основні поняття курсу: БД, СКБД, ІС

---

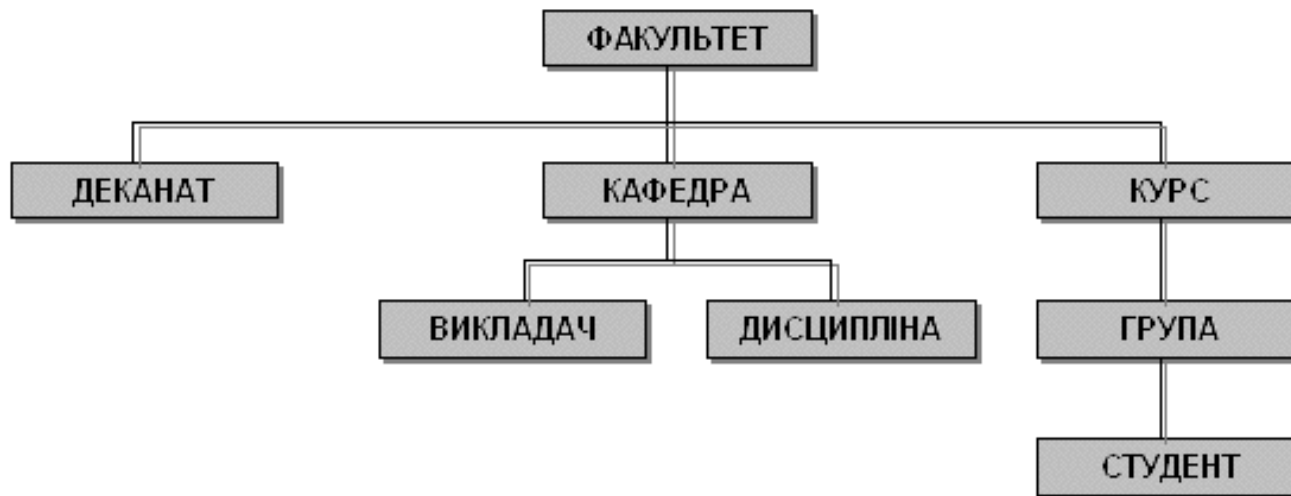
- **База Даних** - упорядкований набір даних.
- **Система Керування Базами Даних** – сукупність програмних і лінгвістичних засобів загального або спеціального призначення, що забезпечують керування створенням і використанням баз даних.
- **Інформаційна Система** – сукупність технічного, програмного та організаційного забезпечення, а також персоналу, призначена для збереження та обробки інформації з метою забезпечення інформаційних потреб користувача.

# Класифікація (СК)БД за моделями даних

---

«... В ієрархічній БД дані концептуально організовані в набори, які пов'язані між собою відношенням володіння ... »

1968 IBM IMS



# Класифікація (СК)БД за моделями даних

---

## Мережні БД

- Початок 70-х, робоча група Database Task Group;
- 2 основні конструкції: записи і зв'язки;
- «... Зв'язок це набір фізичних показників, які задають відносини володіння між об'єктами ... ».

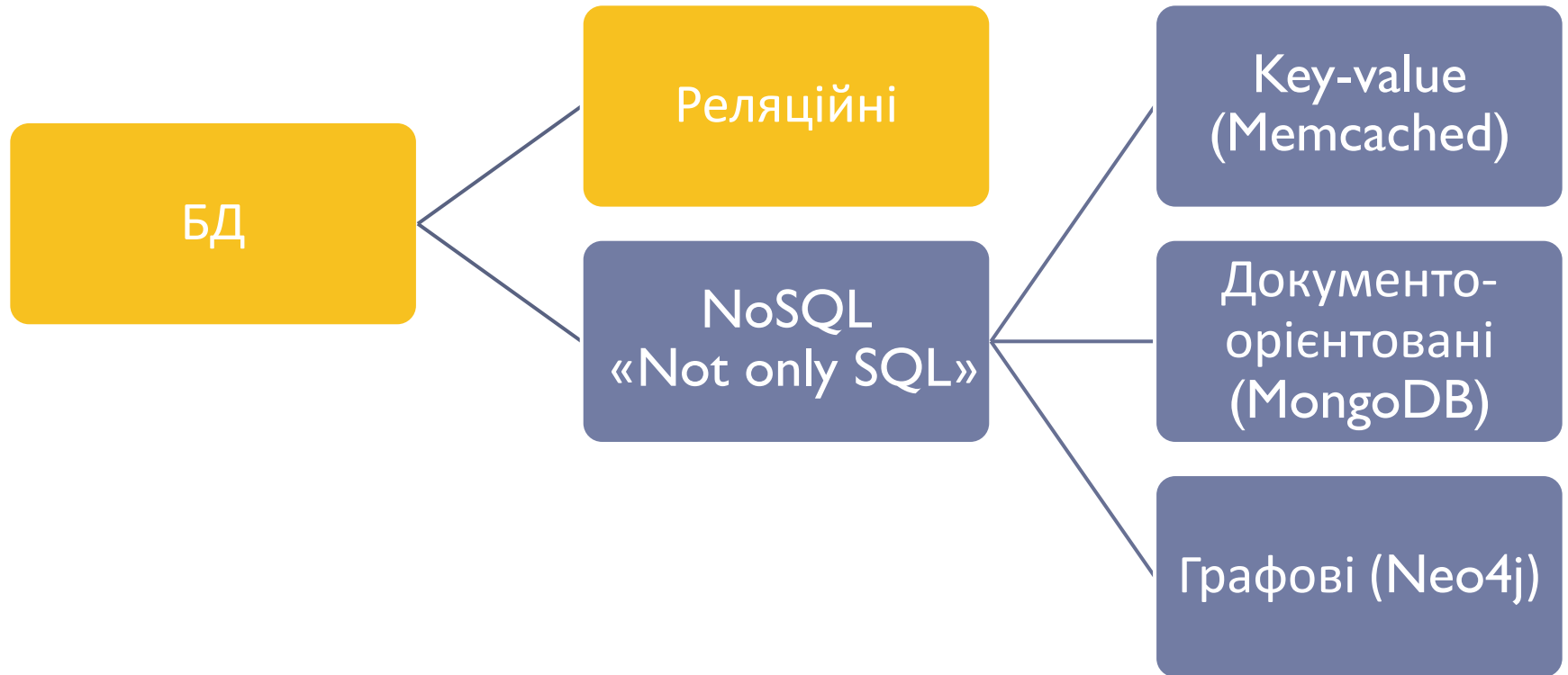
<https://uk.wikipedia.org/wiki/CODASYL>

У мережних моделях безпосередній доступ до будь-якого об'єкта може здійснюватися незалежно від його рівня.



# Які бувають (СК) БД?

---



**Прочитати: Wiki: NoSQL**

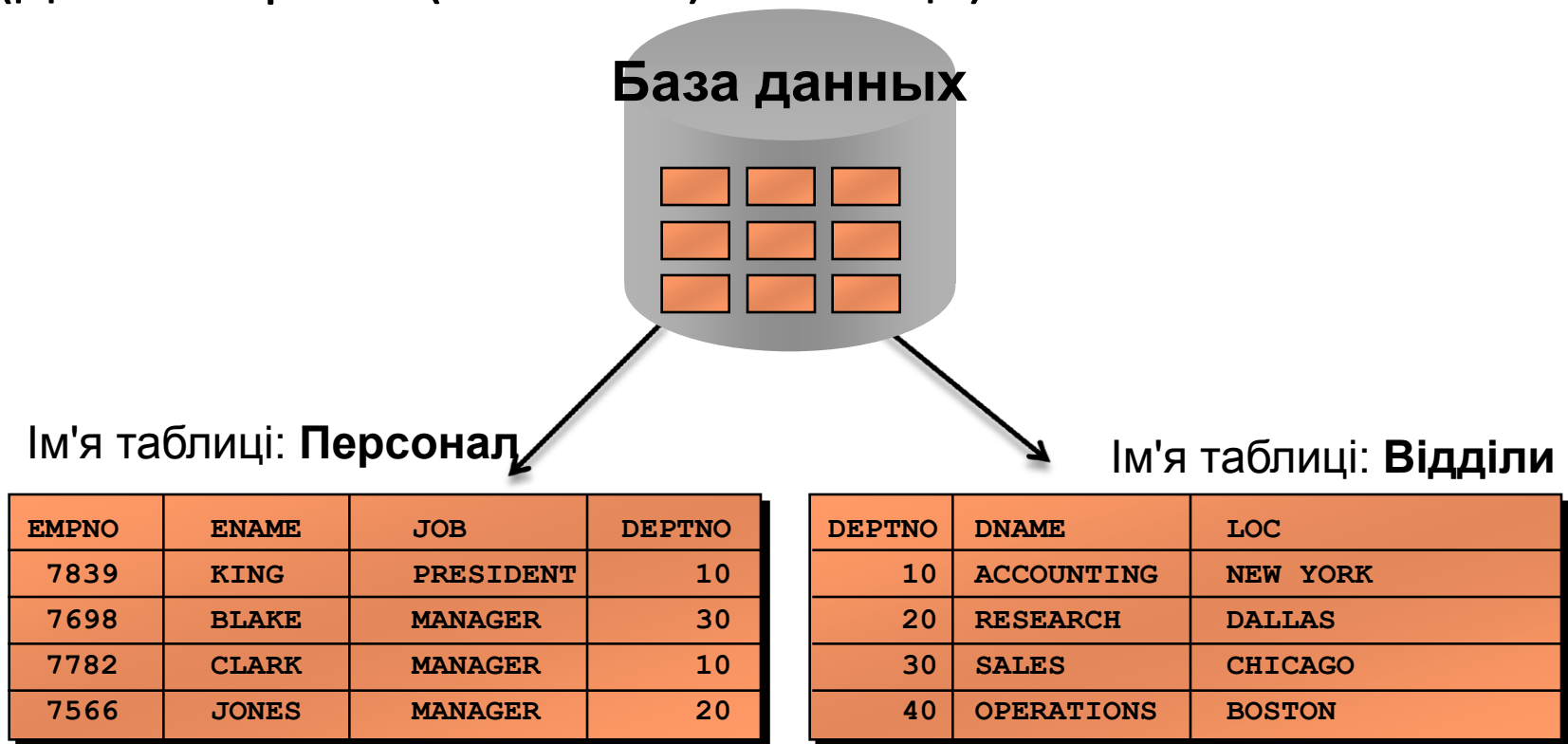
# Реляційний підхід

---

- Е. Кодд запропонував реляційну модель для БД в 1970 р
- Вона є основою реляційних СКБД (Relational database management system, RDBMS).
- Реляційна модель складається з:
  - Набору об'єктів (або відношень);
  - Набору операторів для взаємодії з відношеннями;
  - Засобів забезпечення цілісності.

# Реляційна БД

Реляційна база даних являє собою набір відношень (двовимірних (плоских) таблиць).



# Властивості реляційних БД

---

- Реляційна БД:
  - Містить набір таблиць, який не залежить від способу їх фізичної організації.
  - Доступ до даних і їх модифікація здійснюється за допомогою мови запитів, наприклад SQL.

**SQL** (англ. *Structured query language* — мова структурованих запитів)

# Чи обов'язково використовувати SQL?

---

## QBE

- Query By Example

## LinQ, Transact- SQL

## PL/...

- PL/SQL
- PL/pgSQL
- PL/Java
- ...
- SQL/PSM

# Реляційні БД

## OLTP

- Online transaction processing

	LCD	15"
Compaq	4159	56
GreenWood	23	349
HP	18	
HITACHI	34	984
IBM	119	718
LG	69	119
Philips	984	349
SONY	354	213
ViewSonic	56	

	LCD	15"
Compaq	1159	349
GreenWood		
HP	18	23
HITACHI	34	984
IBM	119	
LG	69	119
Philips		349
SONY	354	213
ViewSonic	56	984

## OLAP

- Online analytic processing

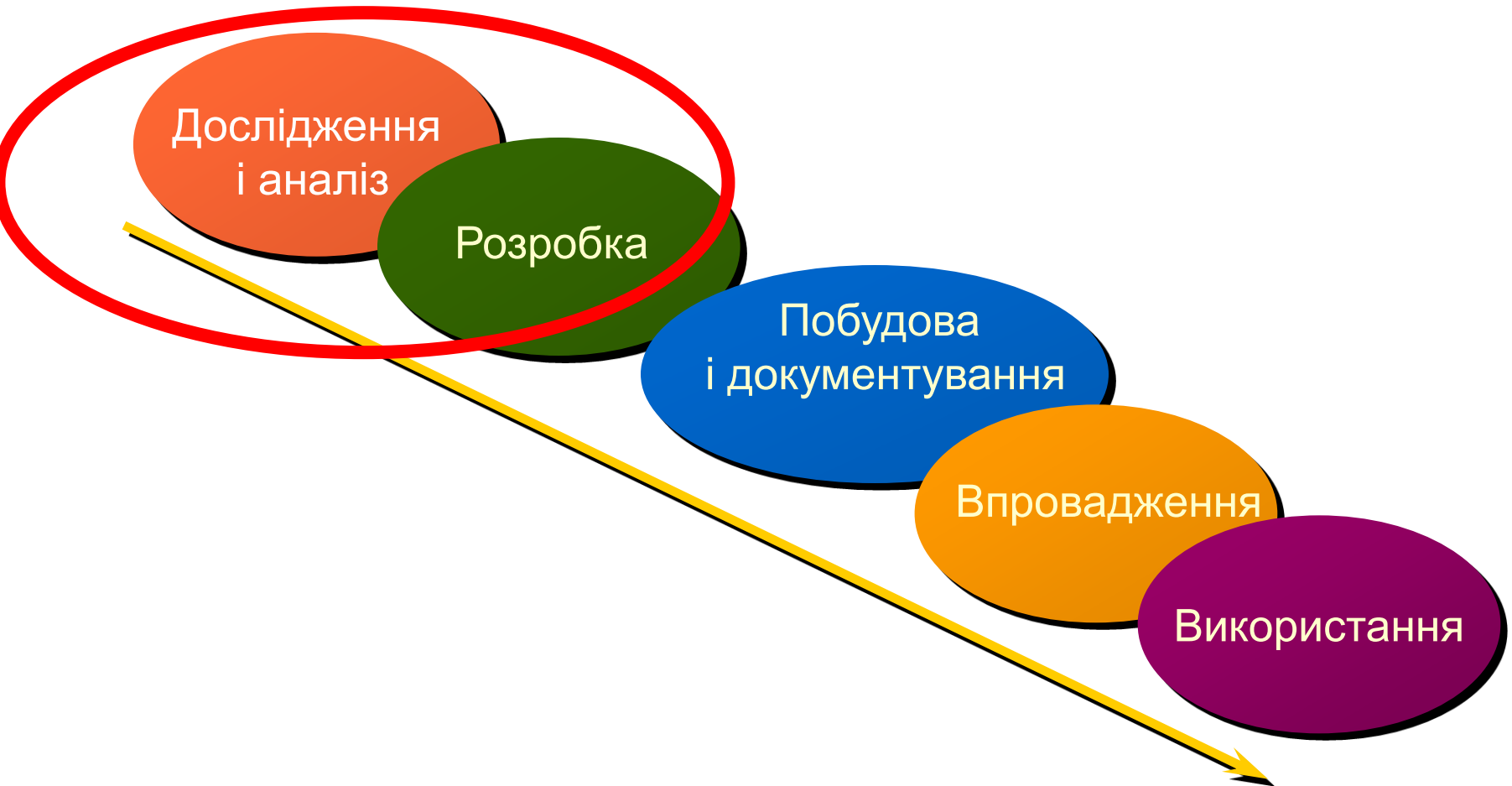
	LCD	15"
Compaq	1159	56
GreenWood	23	349
HP	18	
HITACHI	34	984
IBM	119	718
LG	69	119
Philips	984	349
SONY	354	213
ViewSonic	56	

19.10.02 8:15

19.10.02 12:20

# Життєвий цикл розробки ІС

---





# Концептуальна модель

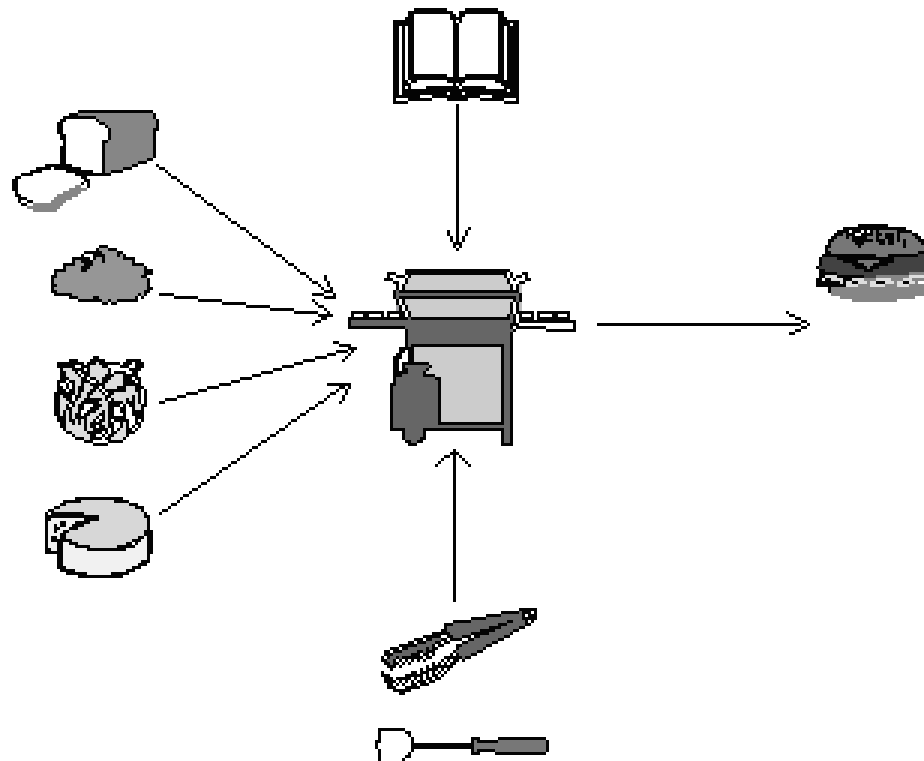
---

**Концептуальна модель** - це абстрактна модель, яка визначає структуру системи, що моделюється, властивості її елементів і причинно-наслідкові зв'язки, які притаманні системі і суттєві для досягнення мети моделювання.

- ▶ Результат абстрагування частини реального світу;
- ▶ Опис об'єкта, що моделюється, в прийнятій нотації;
- ▶ Опис об'єкта, що моделюється, властивих йому бізнес-правил і потоків, сховищ даних.

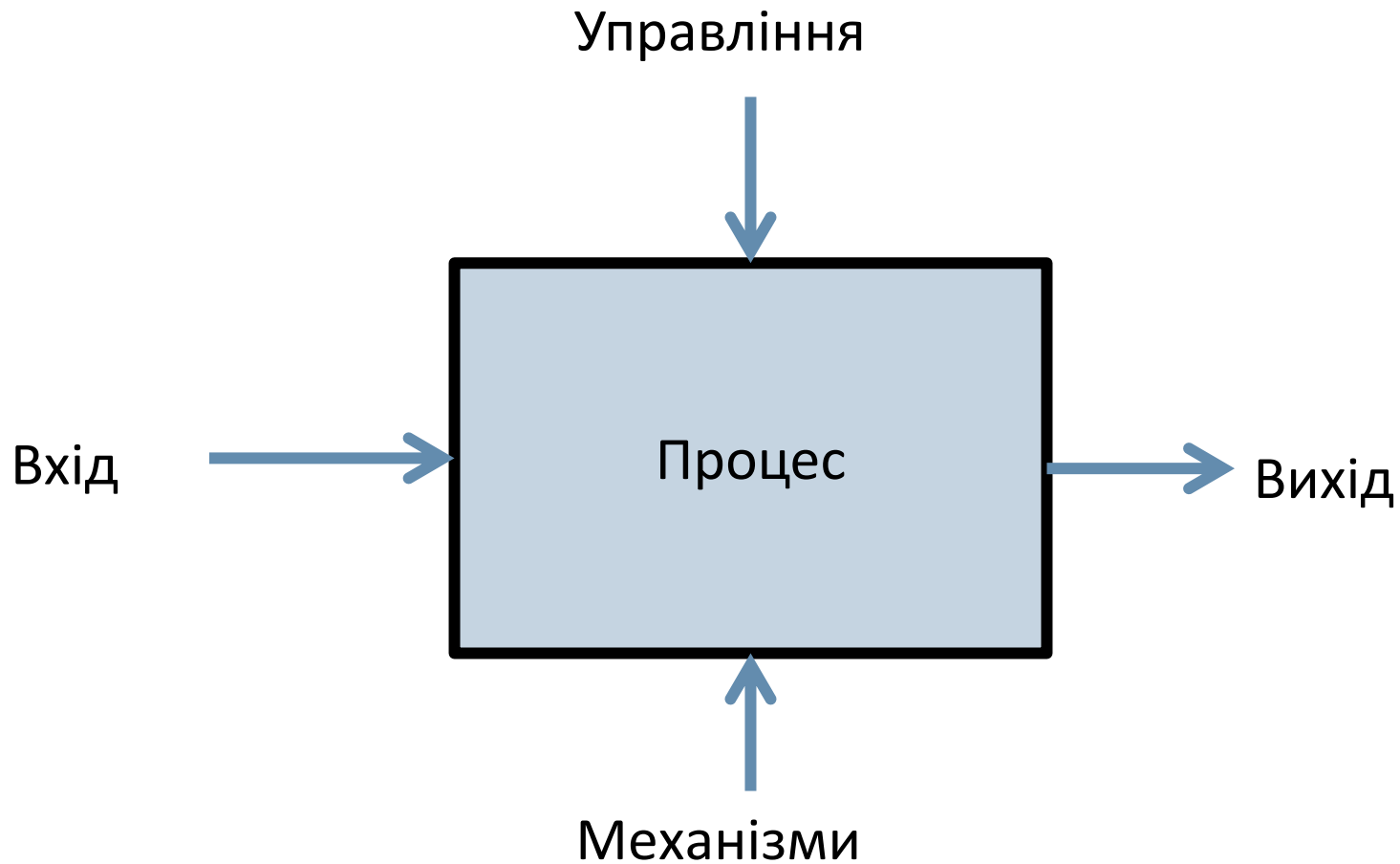
# IDEF0

IDEF0 - методологія функціонального моделювання і графічна нотація, призначена для формалізації і опису бізнес-процесів.



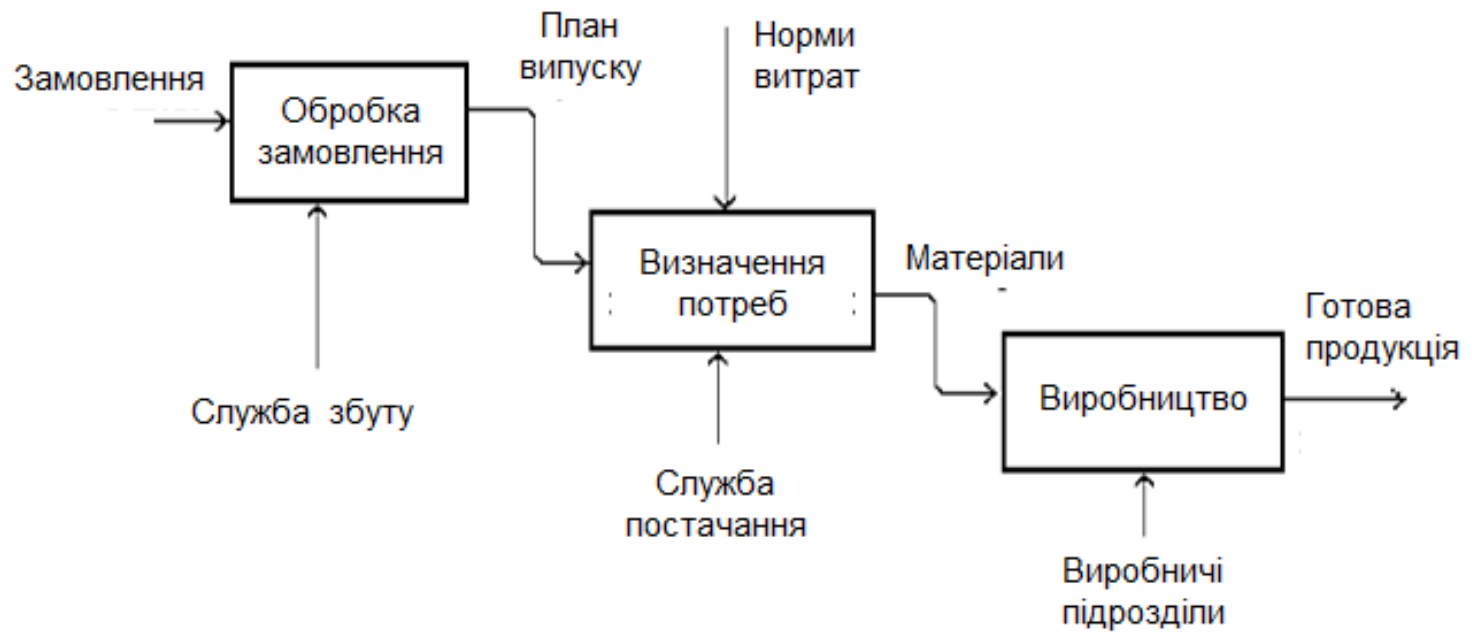
# IDEF0

---



# IDEF0

---


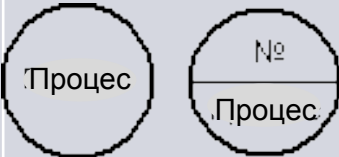
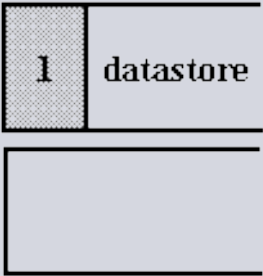




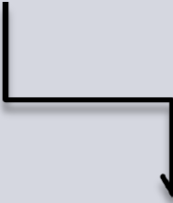


# Data Flow Diagrams

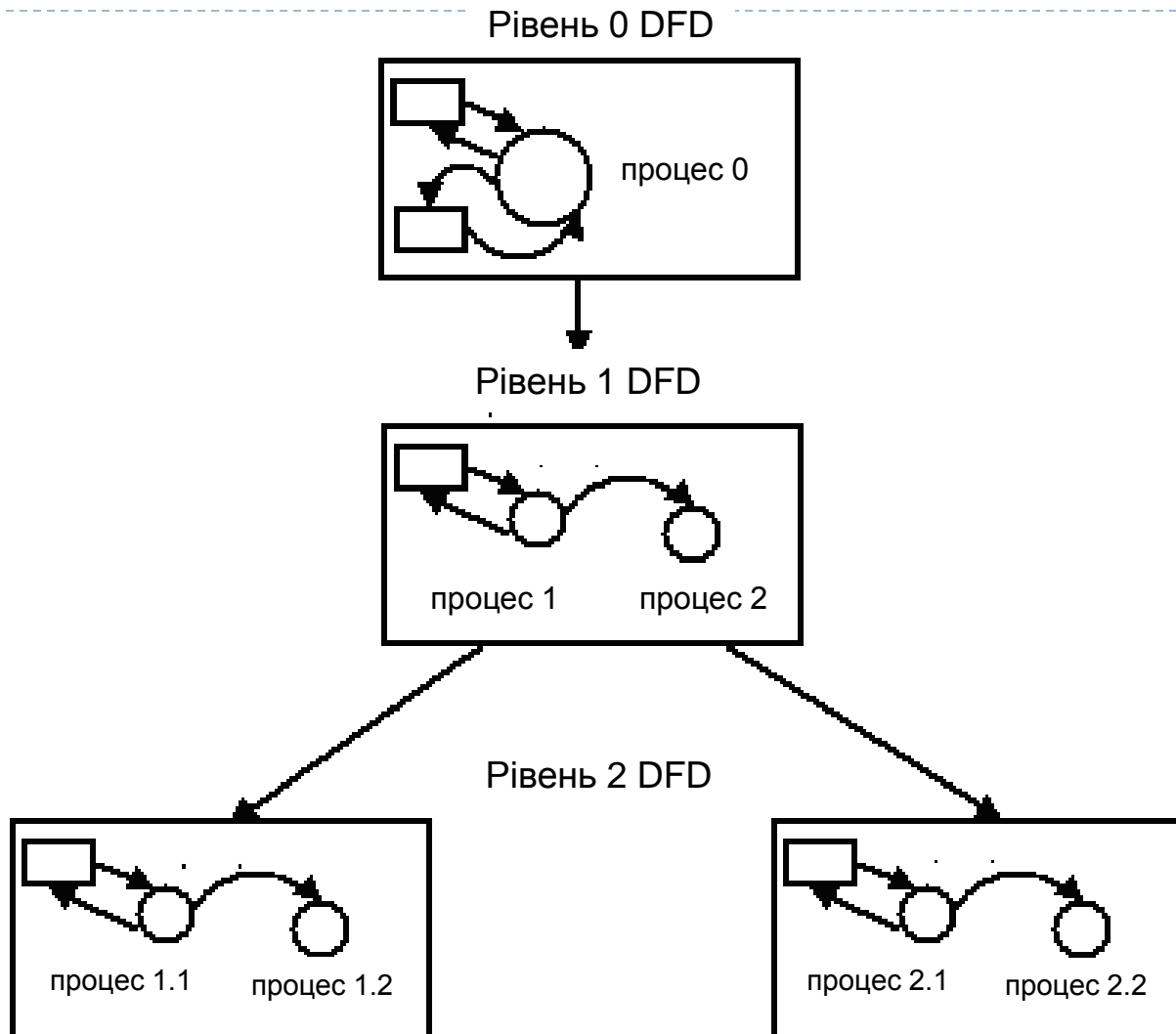
---

- **DFD** (Діаграми потоків даних) - методологія графічного структурного аналізу, що описує зовнішні по відношенню до системи джерела даних, отримувачів даних, логічні функції, потоки даних і сховища даних, до яких здійснюється доступ.
- Призначена для моделювання інформаційних систем з точки зору зберігання, обробки і передачі даних.

# DFD

	Зовнішня сутність (External Entity)	Процес (Process )	Сховище даних (Datastore)	Потік даних (Dataflow)
Gane- Sarson				
Yourdon- Coad				

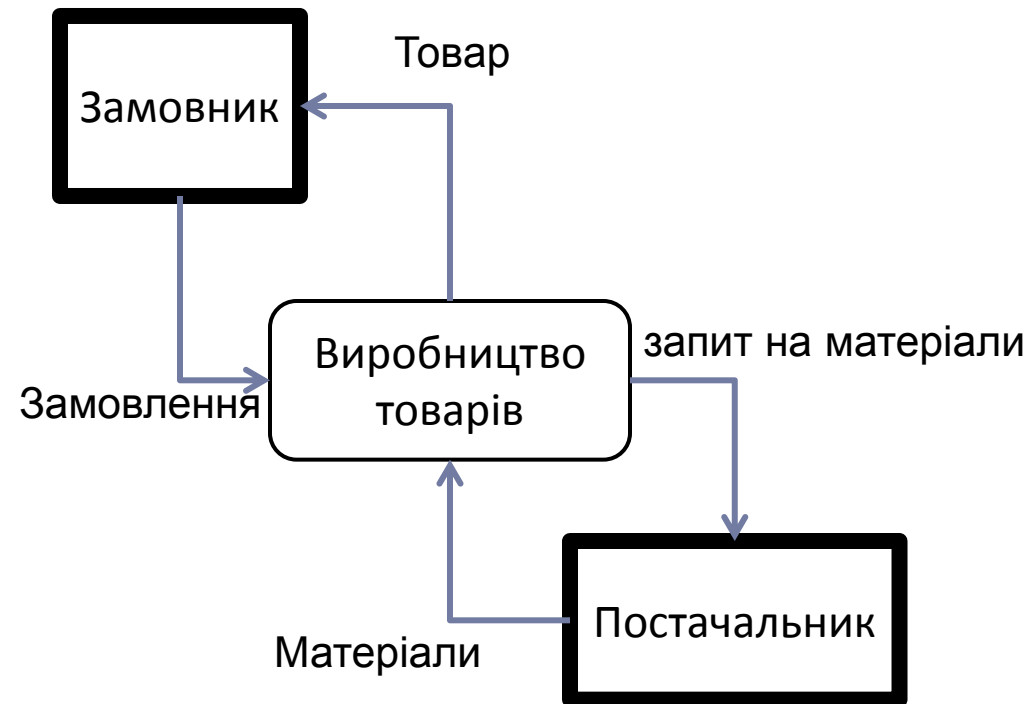
# Рівні діаграми



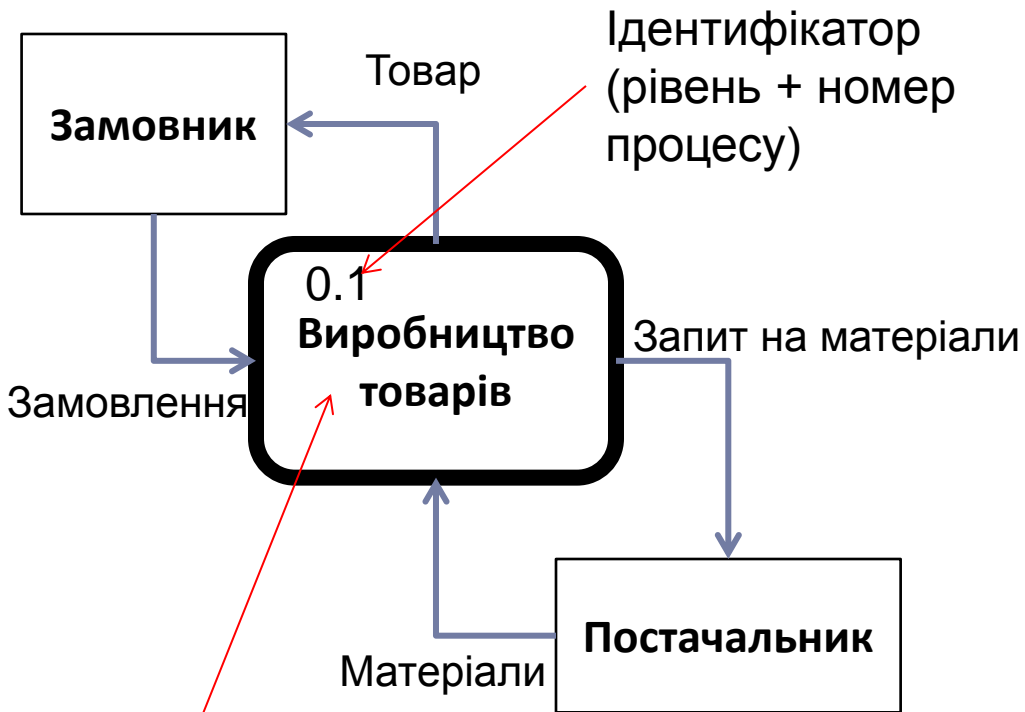


# Зовнішні сутності

- ▶ Зовнішня сутність - об'єкт, який взаємодіє з системою, але не належить до неї;
- ▶ Повинні бути відображені на рівнях **0** і **1** DFD діаграми;
- ▶ Не повинні бути присутніми на рівнях 2 (3, ...) діаграми;
- ▶ Повинні мати змістовні імена;
- ▶ Можуть мати дублікати на діаграмі.



# Процес

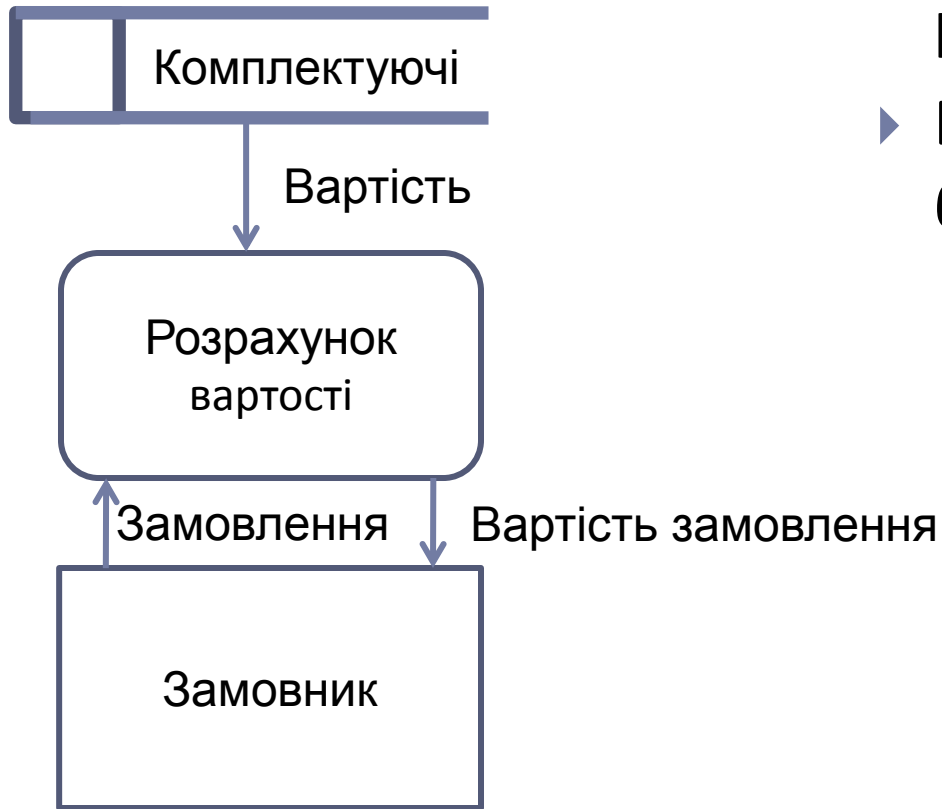


Назва: дієслово  
(Створити, Обчислити, ...) + Об'єкт

- ▶ Описує дії з інформацією.
- ▶ Повинні бути представлені на всіх рівнях діаграми:
  - ▶ Рівень **0** - один процес, що описує систему;
  - ▶ Рівень **1+** - рекомендується розміщувати не більше 7 процесів.
- ▶ Дублікати не дозволені.

# Сховища даних

---



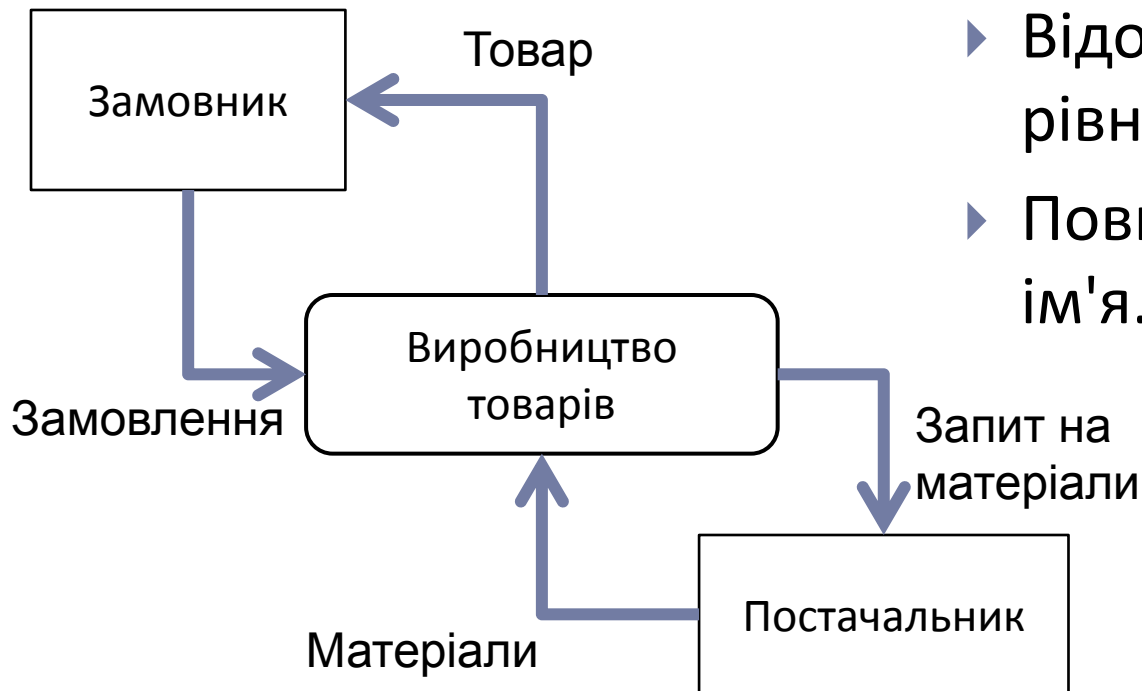
- ▶ Вказує вид інформації;
- ▶ Не відображаються на рівні **0**;
- ▶ Всі сховища повинні бути вказані на рівні **1**.



# Потоки даних

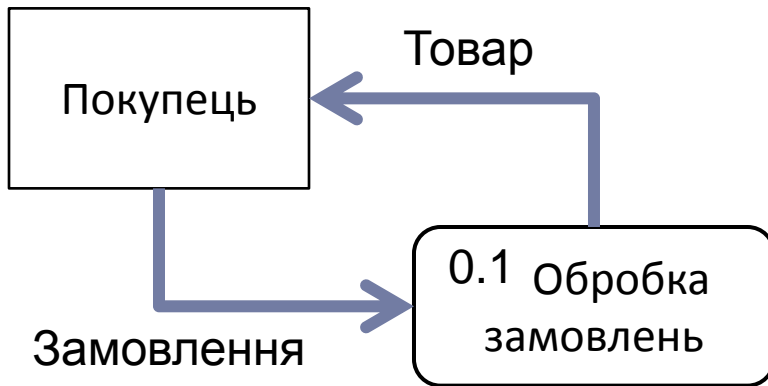
---

- ▶ Відображають напрямок передачі даних;
- ▶ Відображаються на всіх рівнях діаграми;
- ▶ Повинні мати змістовне ім'я.

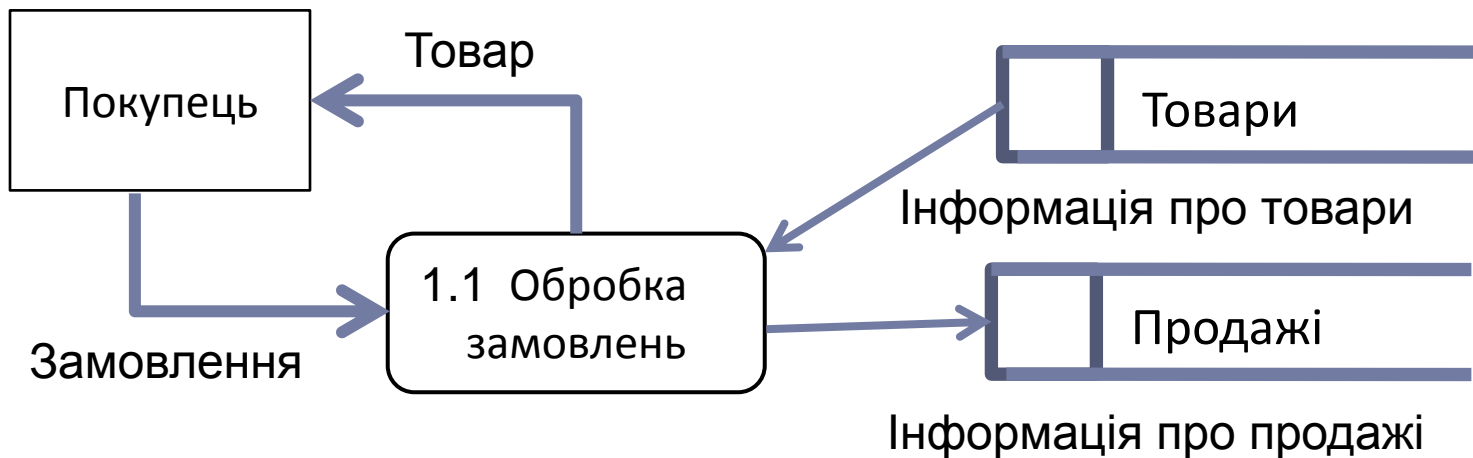


# Приклад 1

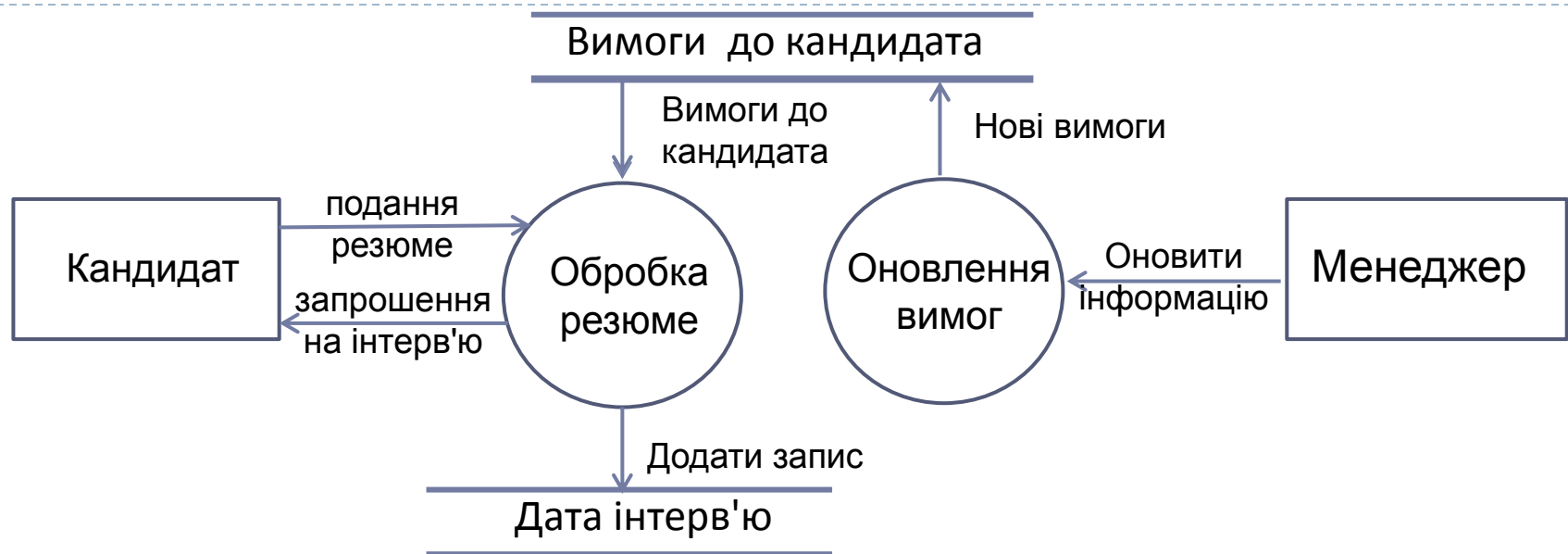
## ► Рівень 0



## ► Рівень 1



## Приклад 2



- ▶ Компанія отримує резюме.
- ▶ Перевіряються вимоги до кандидата, якщо людина підходить його запрошують на інтерв'ю.
- ▶ Вимоги до кандидатів можуть змінюватися менеджментом компанії.

# Алгоритм побудови діаграми

---

- ▶ Визначте вхідні запити або події, на які повинна реагувати система. Визначте, що саме система відповідає на них;
- ▶ Визначте хто формує ці запити і отримує відповідь (це зовнішні сутності);
- ▶ Побудуйте діаграму рівня **0** - на ній відображені тільки зовнішні сутності і головний процес системи;
- ▶ Побудуйте діаграму рівня **1** - на ній повинні бути відображені всі зовнішні сутності, все сховища даних, головні процеси і потоки інформації між ними;
- ▶ Проаналізуйте і оптимізуйте діаграму рівня **1**.  
Рекомендується на одному рівні діаграми розміщувати  $7 \pm 2$  процесів;
- ▶ Якщо потрібно, деталізуйте окремі процеси на рівні 2 (3,4 ...).

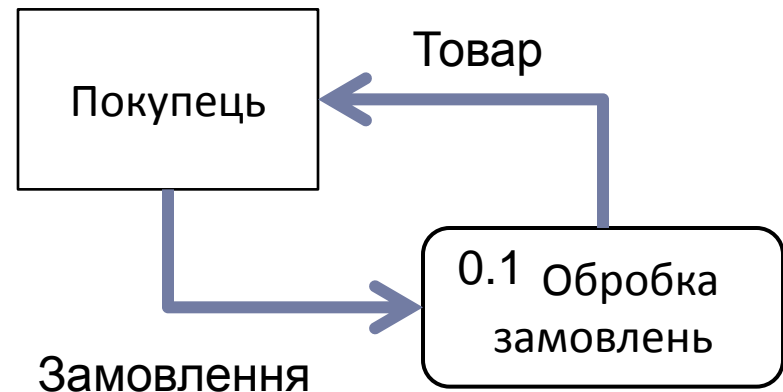




## Пам'ятайте:

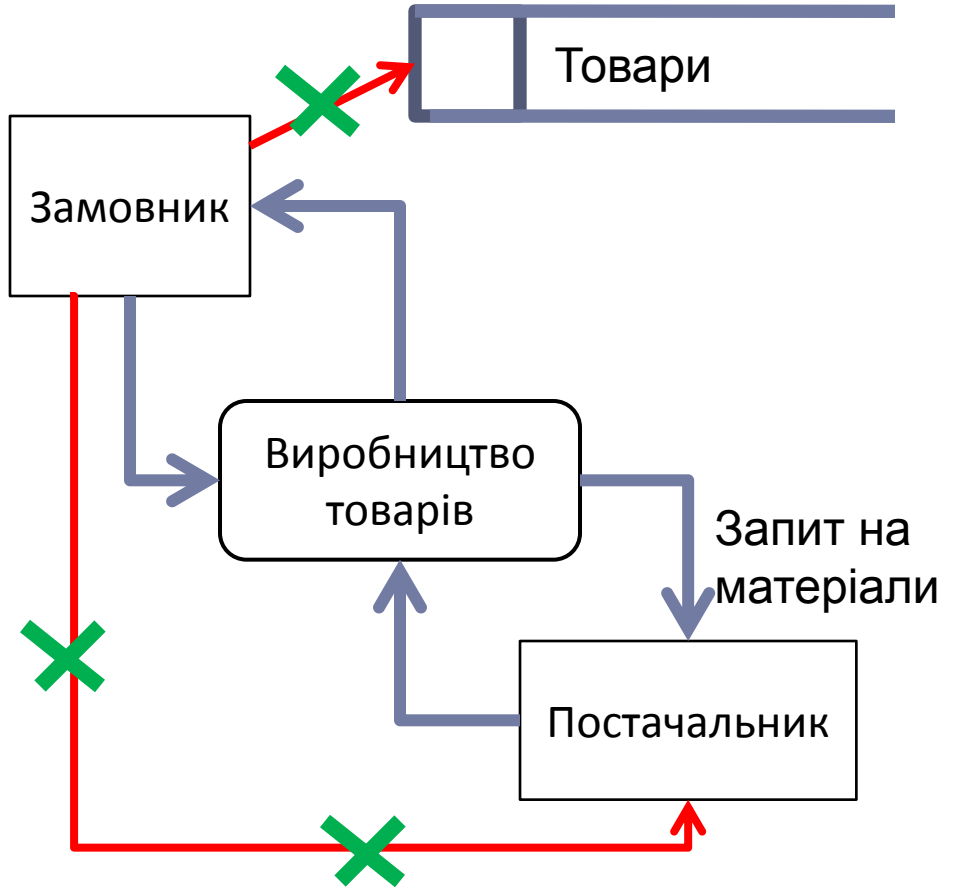
---

- ▶ У кожного процесу, сховища даних, зовнішньої сутності і потоку даних має бути ім'я.
- ▶ На DFD діаграмі не відображаються точки початку і закінчення, рішення (if) і цикли.



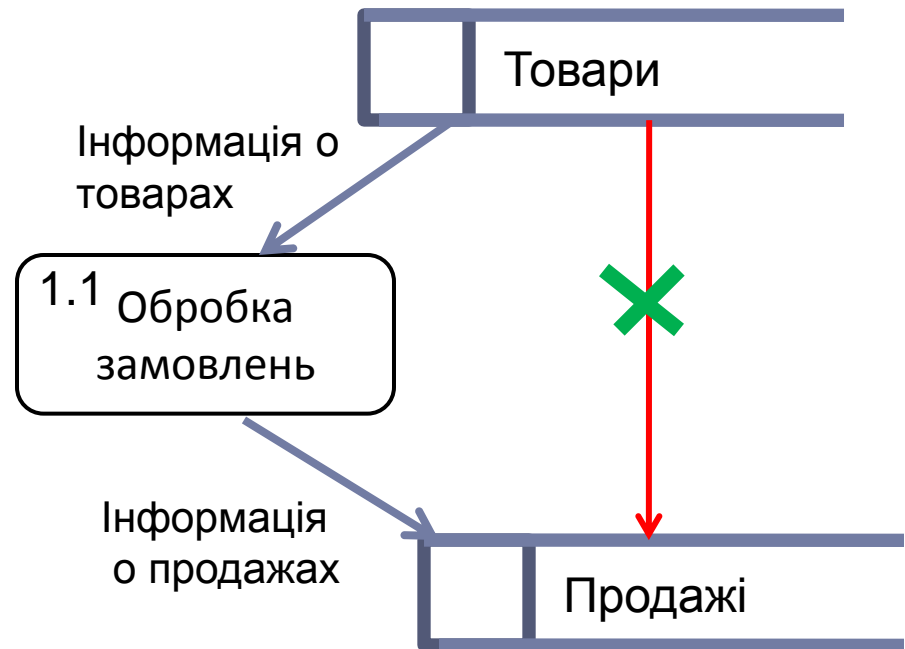
## Пам'ятайте:

- ▶ Зовнішні сутності не можуть бути зв'язані потоками інформації (це виходить за рамки системи, що моделюється ).
- ▶ Зовнішні сутності не можуть безпосередньо звертатися до сховищ даних.



## Пам'ятайте:

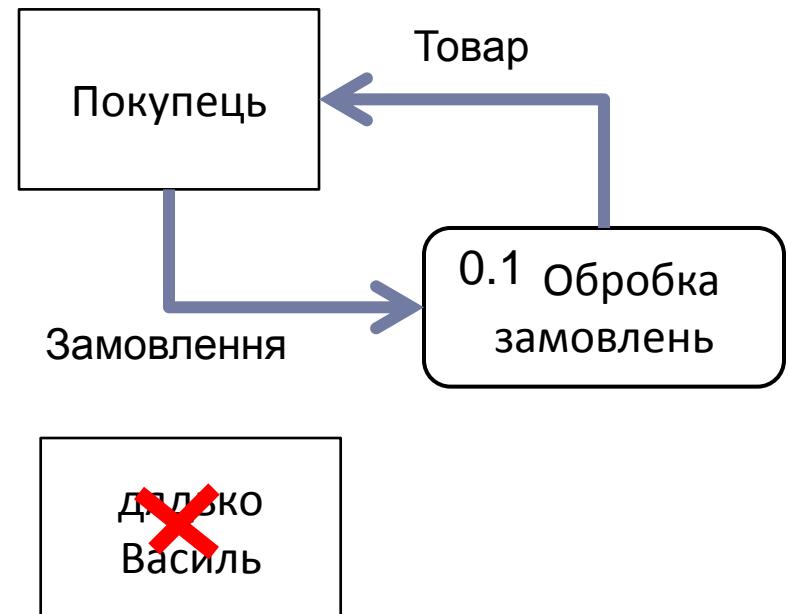
- ▶ Сховище даних не може безпосередньо звертатися до іншого сховища. Між ними повинен бути процес.
- ▶ **Уникайте** сховищ в які тільки записується інформація або з яких тільки читається.



## Пам'ятайте:

---

- ▶ На діаграмі не повинно бути нез'єднаних об'єктів;
- ▶ Уникайте процесів, які не мають вхідних потоків інформації;
- ▶ Уникайте процесів які не мають вихідних потоків інформації;



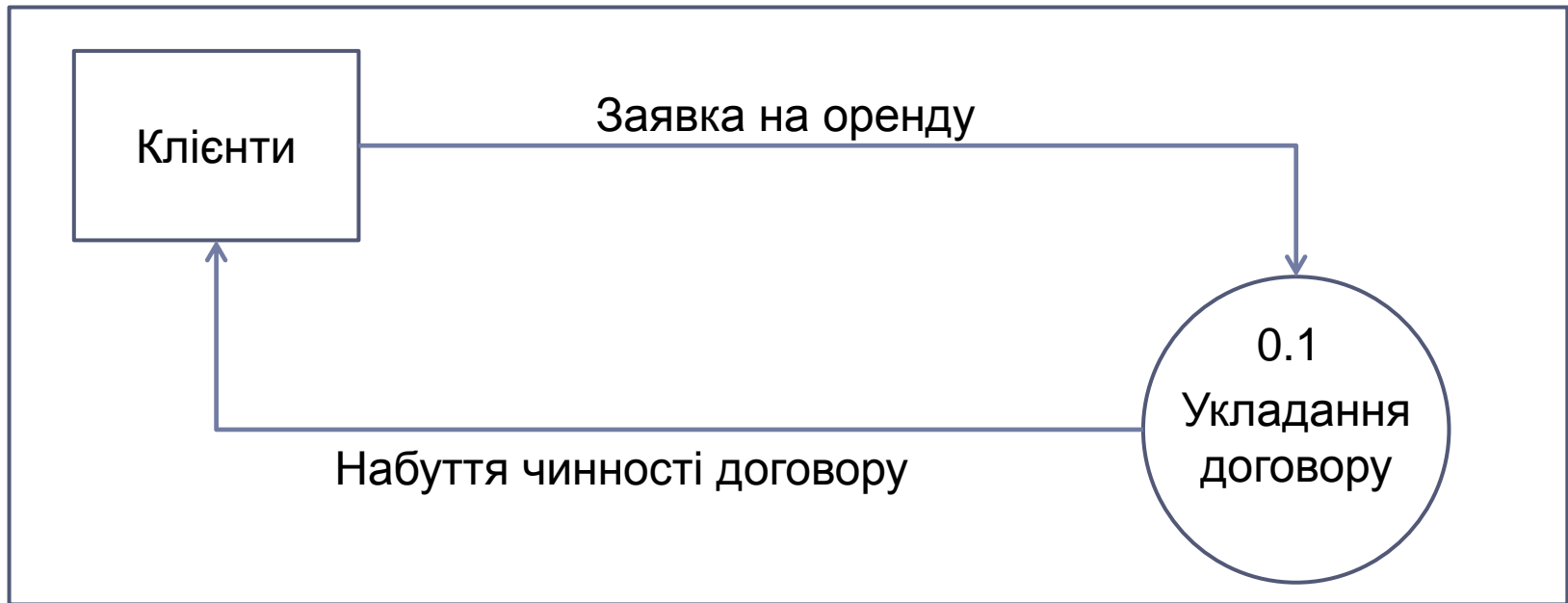
# Задача:

---

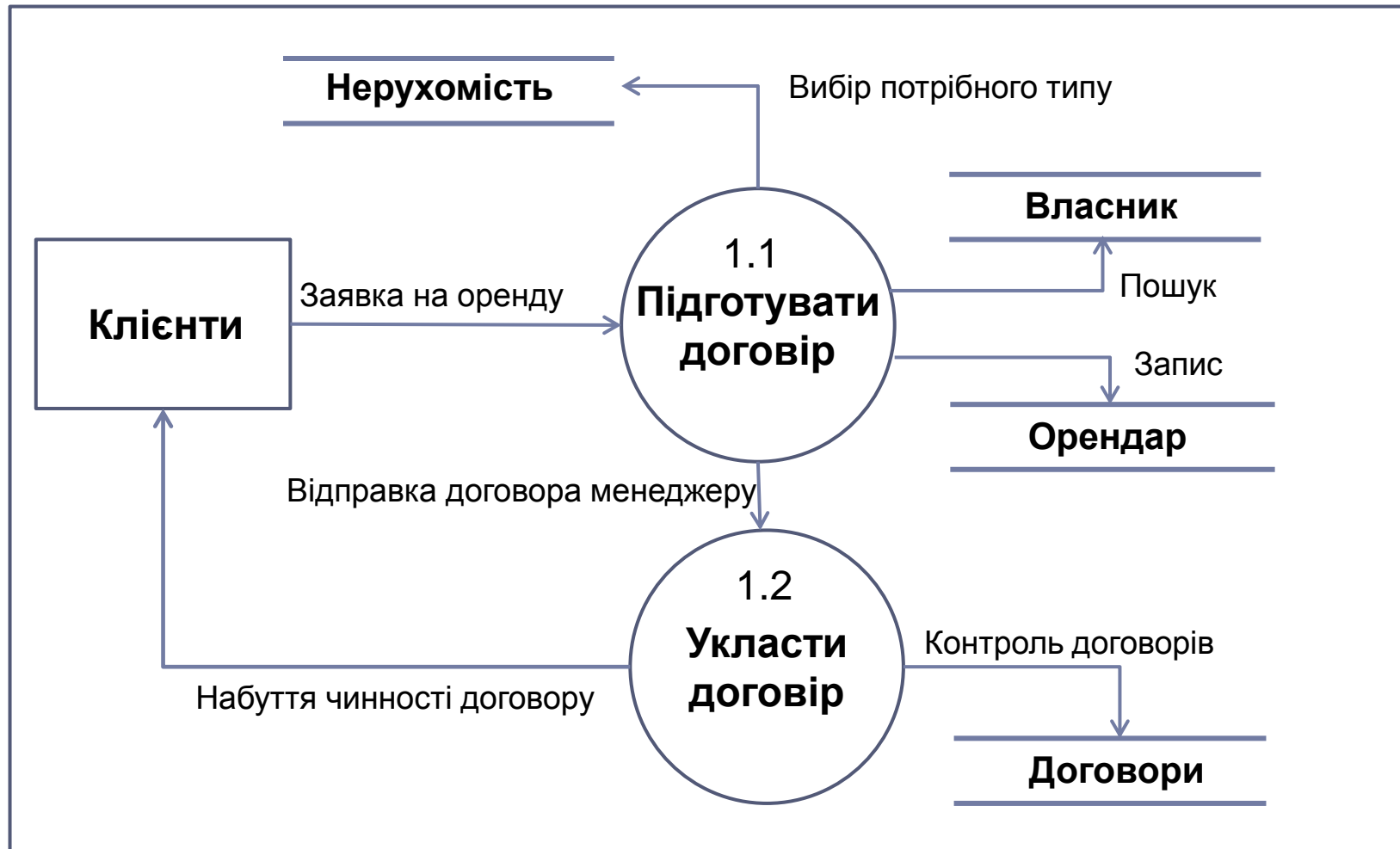
- ▶ Інформаційна система ріелторської контори:
  - ▶ Замовники можуть залишати заявки на оренду;
  - ▶ Менеджер шукає нерухомість і готує проект договору;
  - ▶ Клієнт підписує договір.

# Побудова DFD-діаграми (рівень 0)

---



# Побудова DFD-діаграми (рівень 1)





# Використання CASE Studio

---

Демо-ролик про створення DFD

# Підсумки

---

- ▶ Ввели поняття:
  - ▶ База даних;
  - ▶ Система управління базами даних;
  - ▶ Інформаційна система;
- ▶ Типи БД;
- ▶ OLTP / OLAP;
- ▶ Проектування ІС;
- ▶ Основні етапи проектування;
- ▶ Навіщо потрібно проектувати;
- ▶ Методологія:
  - ▶ IDEF0;
  - ▶ DFD;
  - ▶ UML;

# Домашнє читання: UML

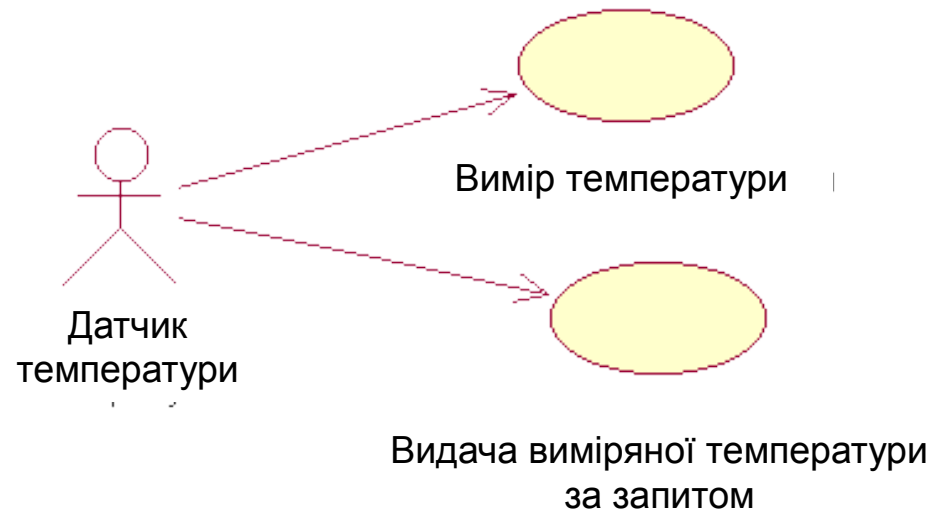
---

- ▶ Use case diagram (діаграми прецедентів);
- ▶ Deployment diagram (діаграми топології);
- ▶ State machine diagram (діаграми станів)
  - ▶ Statechart diagram (діаграма станів)
  - ▶ Activity diagram (діаграми активності)
- ▶ Interaction diagram (діаграми взаємодії);
  - ▶ Sequence diagram (діаграми послідовностей дій);
  - ▶ Collaboration diagram (діаграми співпраці);
- ▶ Class diagram (діаграми класів);
- ▶ Component diagram (діаграми компонентів).

# 1. Use case (діаграми прецедентів)

---

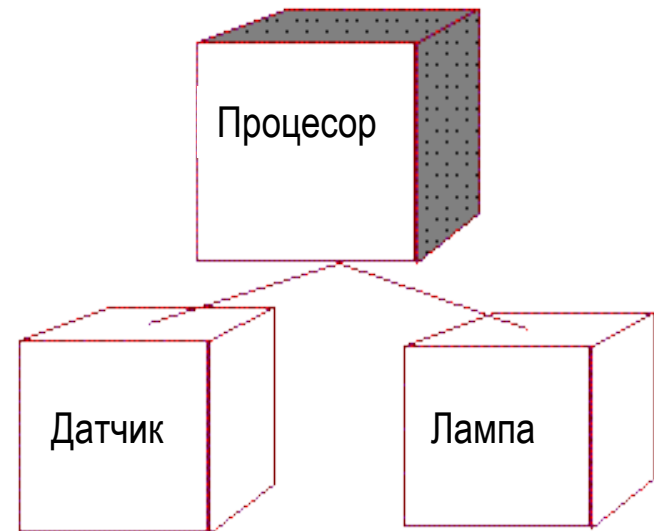
- ▶ Показує список операцій, які виконує система.
- ▶ Задає список вимог до системи і визначає безліч функцій, які виконуються системою.
- ▶ Use case - це опис сценарію поведінки, якого дотримуються діючі особи (Actors).



## 2. Deployment diagram (діаграми топології)

---

- ▶ Призначена для аналізу апаратної частини системи.
- ▶ Для кожної моделі створюється тільки одна така діаграма
- ▶ Відображає процесори (Processor), пристрої (Device) і їхні з'єднання.

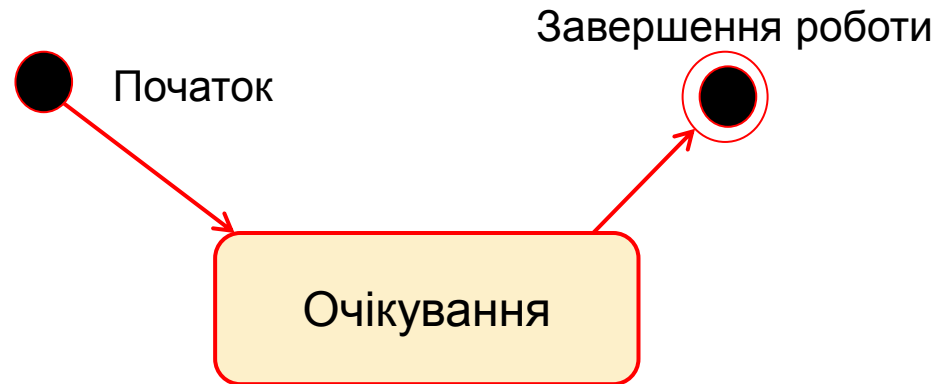


## 3. State Machine diagram (діаграми станів)

---

### 3.1 Statechart diagram (діаграма станів)

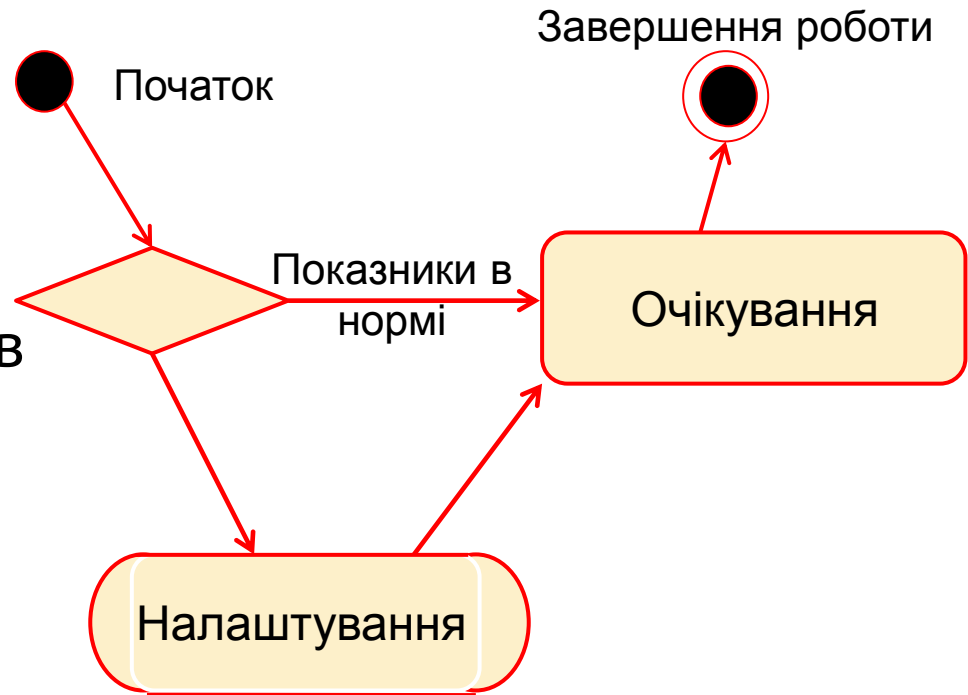
- ▶ Відображає стани об'єктів системи, що мають складну модель поведінки



### 3. State Machine diagram (діаграми станів)

#### 3.2 Activity diagram (діаграми активності)

- ▶ Відображає бізнес-процеси об'єкта
- ▶ Використовується для проектування алгоритмів (згадайте блок-схеми)

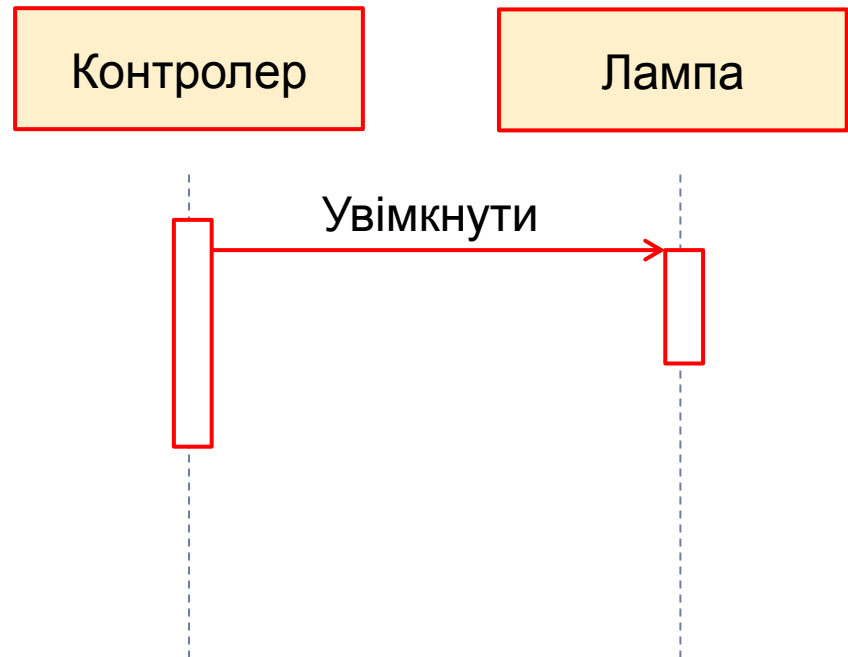


## 4. Interaction diagram (діаграми взаємодії)

---

### 4.1 Sequence diagram (Діаграми послідовностей дій)

- ▶ Відображає послідовність передачі повідомлень між об'єктами
- ▶ Акцентує увагу на послідовності прийому / передачі повідомлень



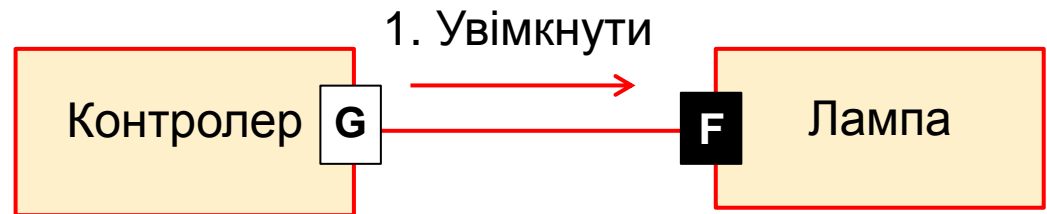


## 4. Interaction diagram (діаграми взаємодії)

---

### 4.2 Collaboration diagram (Діаграми співробітництва)

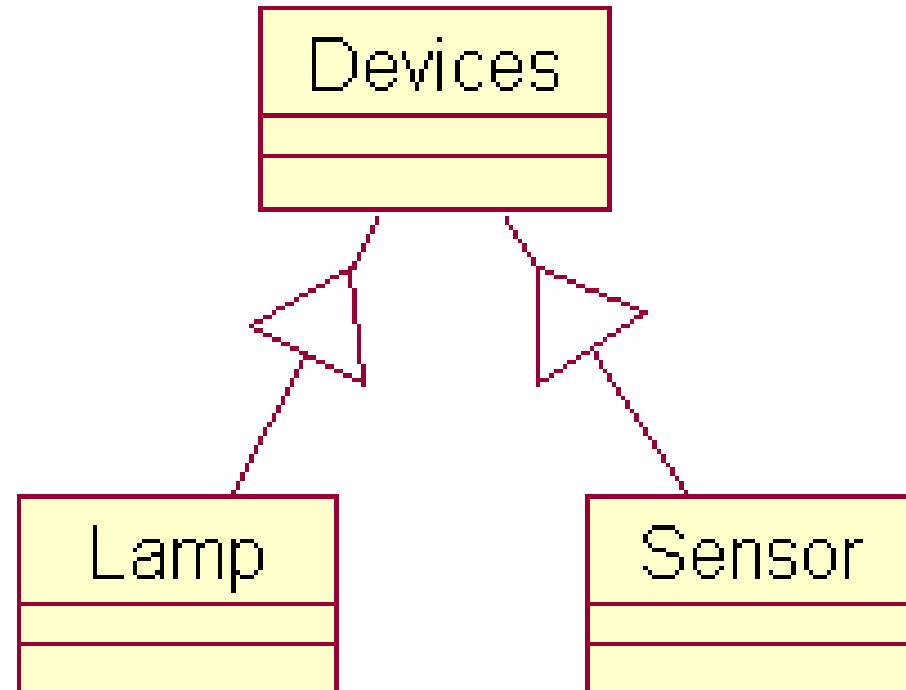
- У компактному вигляді відображає **всі** прийняті і передані повідомлення конкретного об'єкта і типи цих повідомлень



## 5. Class diagram (Діаграми класів)

---

- ▶ Дозволяє створювати логічне представлення системи, на основі якої створюється вихідний код описаних класів.



## 6. Component diagram (діаграма компонентів)

- ▶ *призначена для розподілу класів і об'єктів за компонентами при фізичному проектуванні системи*

