

## Лекція 1

# ОСНОВИ ПРИКЛАДНОГО ПРОГРАМУВАННЯ

# Лекція 1. Основи системного програмування

## План

1. Теоретичні основи програмування
2. Прикладне та системне програмування
3. Мови програмування, що використовуються для створення прикладного програмного забезпечення
4. Література

# 1. Теоретичні основи програмування

**Програма** – це комбінація комп'ютерних інструкцій та даних, яка дозволяє апаратному забезпеченню обчислювальної системи виконувати розрахунки або функції управління (стандарт ISO/IEC/IEEE 24765:2010)

**Програмне забезпечення (ПЗ)** – це сукупність програм, призначених для керування комп'ютером.

**Класифікація ПЗ за сферою застосування:**

- наукове;
- для побутових пристроїв;
- для забезпечення роботи обладнання;
- для бізнесу;
- для організації роботи мережі;
- ...

**Класифікація ПЗ за призначенням:**

- прикладне;
- системне;
- інструментальне.

**Програмування** – процес розробки ПЗ, що поєднує в собі елементи науки (математики), інженерії та мистецтва. У вузькому значенні програмування – кодування алгоритму з допомогою деякої мови програмування (алгоритмічного мови). У широкому – тривалий та творчий процес створення програм.

# 1. Теоретичні основи програмування

Для управління будь-яким процесом необхідна систематизована, попередньо підготовлена достовірна **інформація**.

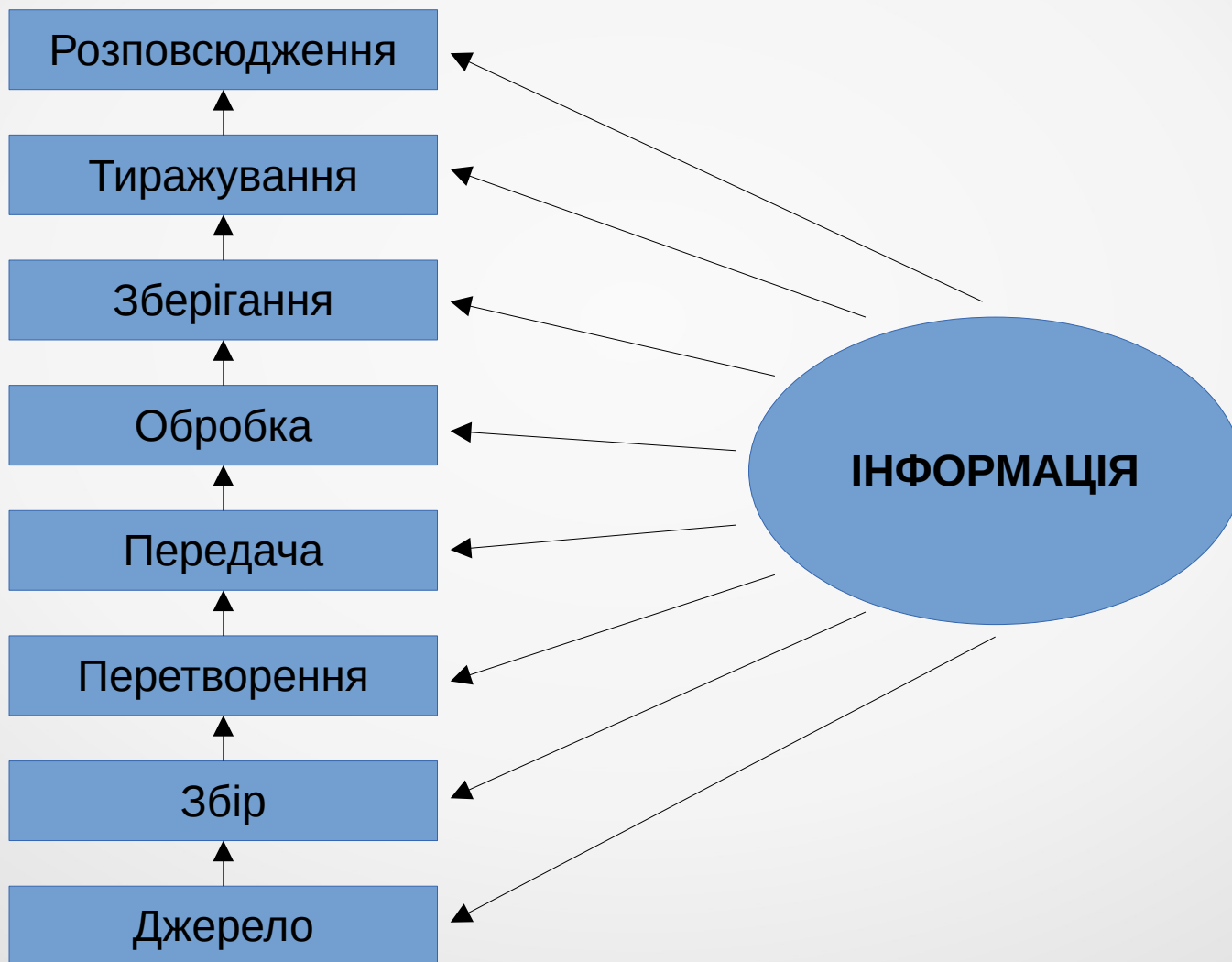
У загальному випадку **інформацією** називається певна сукупність відомостей про навколишній світ, події або діяльності.

Форма подання інформації – **повідомлення**. Однак не будь-яке повідомлення може вважатися інформацією. Інформацією вважається тільки те повідомлення, що зменшує невизначеність у певній предметній області, до якої воно відноситься.



# 1. Теоретичні основи програмування

## Етапи обробки інформації



# 1. Теоретичні основи програмування

## Класифікація інформації

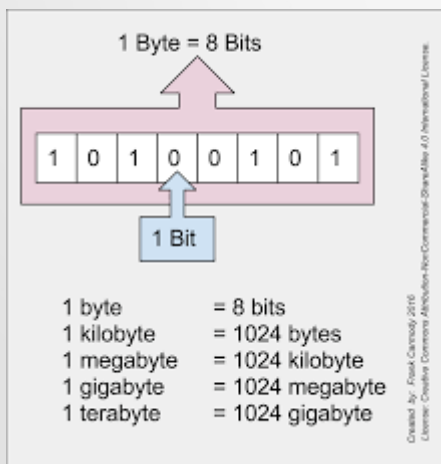
- 1) **За місцем виникнення** й напрямком руху інформація розрізняється як **вхідна** й **вихідна**; вхідну й вихідну інформацію розрізняють як **внутрішню** (виникає усередині об'єкта) й **зовнішню** (за межами об'єкта).
- 2) **За стадіями утворення** інформація підрозділяється на **первинну** й **вторинну**.
- 3) **Згідно процесу обробки** інформація класифікується на ту, що **оброблюється** і ту, що **не оброблюється**.
- 4) **За способом відображення даних** інформація класифікується як **текстова**, **графічна** або **мультимедійна**.
- 6) **За ознакою стабільності** інформація ділиться на **змінну** й **постійну**
- 7) **За способом передачі** (кур'єром; за допомогою електронних засобів телекомунікації тощо).

# 1. Теоретичні основи програмування

Важливою характеристикою інформації є її обсяг

Обсяг інформації вимірюється в таких одиницях:

- 1) **натуральних** – число повідомлень, документів, рядків, показників, слів, символів;
- 2) **телеграфних** – біти (базова одиниця виміру інформації, яка може приймати два значення – 0 або 1), байти (8 біт), кілобайти (1024 байт), мегабайти (1024 кілобайта) тощо.



Натуральні одиниці використовуються в разі обробки інформації людиною, а телеграфні – при організації автоматизованої обробки інформації за допомогою комп'ютерної техніки.

# 1. Теоретичні основи програмування

## Апаратне забезпечення

Технічну основу програмного забезпечення складають засоби **комп'ютерної, комунікаційної та організаційної** техніки.

Саме засоби комп'ютерної техніки є базисом всього комплексу технічних засобів програмного забезпечення і призначені для обробки різних видів інформації, що застосовується в повсякденній людській діяльності.





# 1. Теоретичні основи програмування

Сучасний **комп'ютер** – це електронний прилад, призначенням якого є автоматизація операцій з даними. Гнучкість автоматизації досягається завдяки тому, що операції з даними виконуються по заздалегідь заготовленим і легко змінюваним програмам. Універсальність комп'ютерів базується на тому принципі, що будь-які типи даних представляються в ньому за допомогою одноманітного двійкового кодування.

## Приклади класифікації комп'ютерів

### Міжнародний стандарт PC99

Згідно з PC99 виділяють наступні категорії персональних комп'ютерів:

- масовий ПК (consumer PC);
- офісний ПК (office PC);
- мобільний, переносний (mobile PC);
- робоча станція (workstation PC);
- розважальний ПК (entertainment PC).

### Класифікація за розмірами

- настільні (desktop, nettop);
- портативні (notebook, netbook);
- кишенькові (palmtop).

## 2. Прикладне та системне програмування

### Програмне забезпечення

Для автоматизації обробки інформації із застосуванням комп'ютера необхідне відповідне програмне забезпечення, яке можна розділити на дві окремі великі групи:

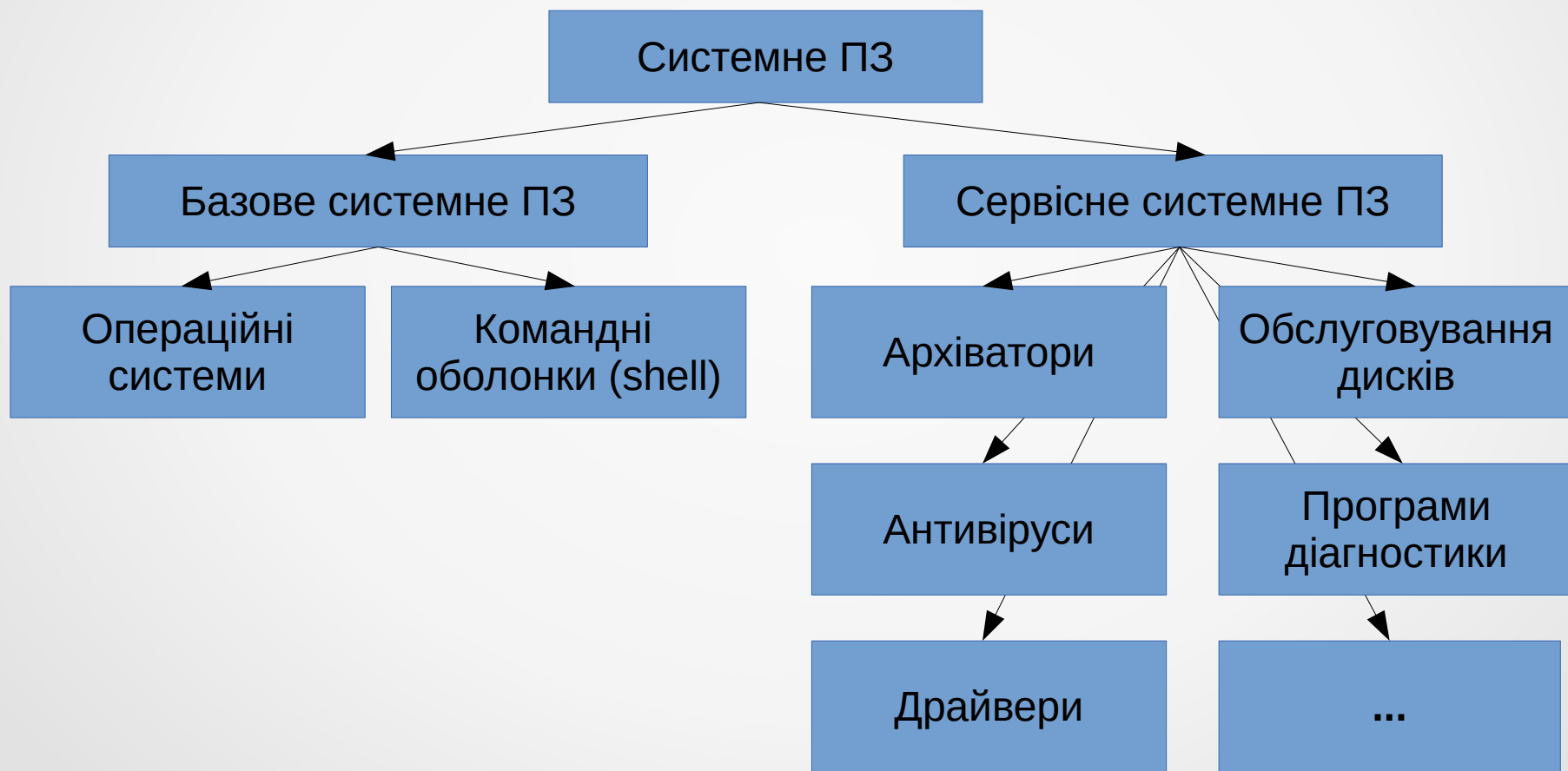
- 1) **системне;**
- 2) **прикладне.**

**Системне ПЗ** організовує процес обробки інформації на комп'ютері і забезпечує нормальне робоче середовище для прикладних програм.

До складу системного ПЗ входять:

- операційні системи;
- сервісні програми;
- програми технічного обслуговування;
- транслятори;
- ...

## 2. Прикладне та системне програмування



## 2. Прикладне та системне програмування

**Операційна система (ОС)** – це сукупність програм, що управляє апаратними і програмними ресурсами комп'ютера, а також забезпечує виконання команд користувача.

**Сервісне ПЗ** це сукупність програмних продуктів, що надають користувачам додаткові послуги в роботі з комп'ютером і розширюють можливості ОС.

**Програми технічного обслуговування** призначені для діагностики і виявлення помилок в процесі роботи комп'ютера або обчислювальної системи в цілому.

**Транслятором** називається програма, що здійснює переклад тексту програми з однієї мови програмування в іншу. Розрізняють два типи трансляторів: **компілятори й інтерпретатори**.

## 2. Прикладне та системне програмування

**Прикладне ПЗ** призначено для вирішення конкретних завдань користувача і організації обчислювального в цілому.

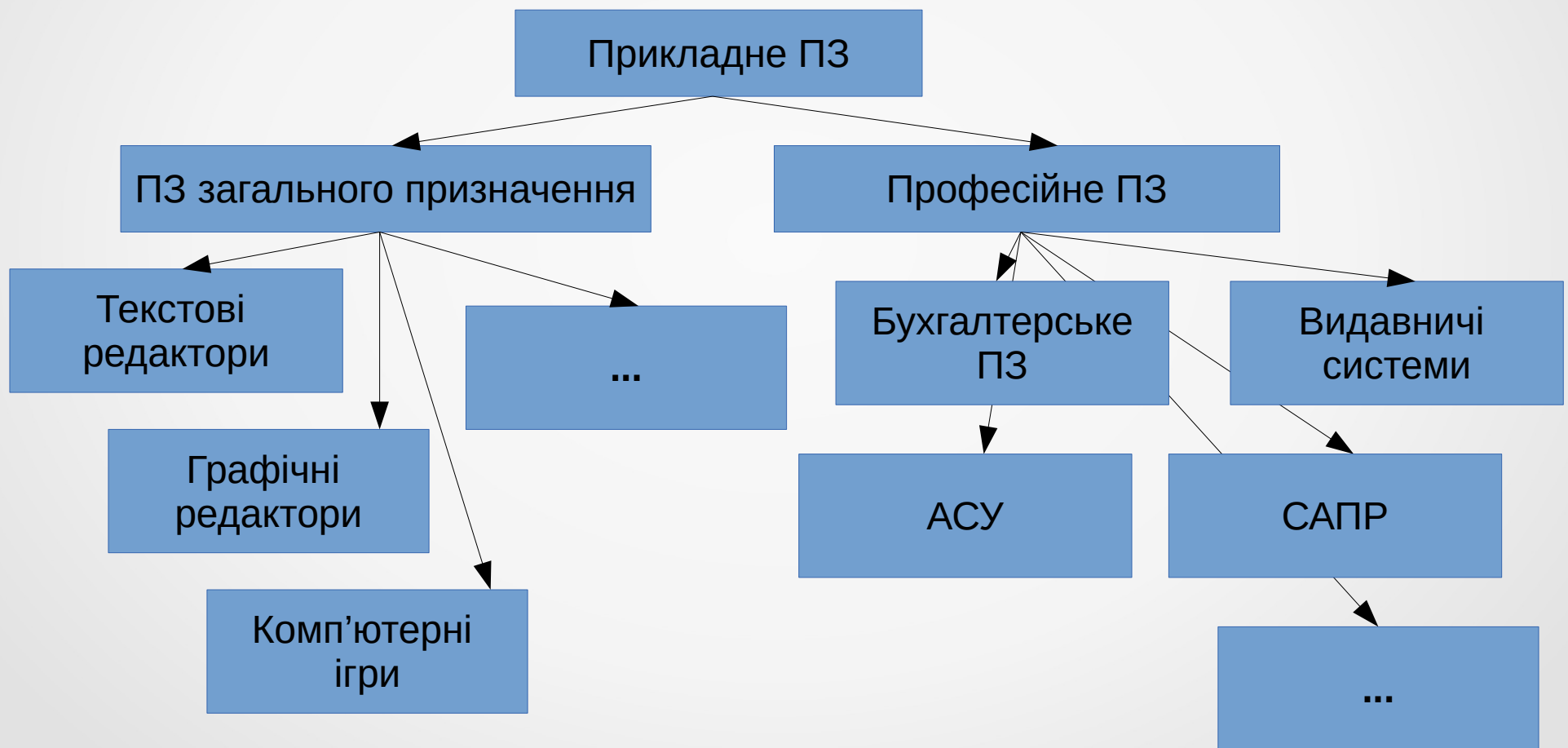
Прикладні програми дозволяють розробляти і виконувати завдання користувача, наприклад, при роботі з інформаційними системами, при бухгалтерському обліку, управлінню персоналом, автоматизації створення креслень і т. д.

Прикладні програми працюють під управлінням системного ПЗ.

До складу прикладного ПЗ входять програми:

- загального призначення;
- функціонального (професійного) призначення.

## 2. Прикладне та системне програмування



## 2. Прикладне та системне програмування

**Програми загального призначення** це універсальні програмні продукти, що призначені для автоматизації розробки і експлуатації функціональних завдань користувача в цілому. До них зазвичай відносяться:

- текстові редактори
- графічні редактори;
- електронні таблиці;
- системи управління базами даних (СУБД);
- ...

**Програми функціонального призначення** орієнтовані на автоматизацію функцій користувача в конкретній сфері його діяльності:

- системи автоматизації проектувальних робіт (САПР);
- автоматизовані системи управління (АСУ);
- корпоративні інформаційні системи (КІС);
- системи документообігу;
- ...

## 2. Прикладне та системне програмування

**Прикладним програмуванням** називається процес розробки прикладного ПЗ (для вирішення конкретних прикладних задач).

**Системним програмуванням** називається процес розробки системного ПЗ (призначеного для управління компонентами обчислювальної системи: процесора, пам'яті, пристрої введення-виводу тощо).

На практиці можна вживати таку умовну класифікацію: прикладна програма переважно взаємодіє з користувачем, а системна – з апаратним (hardware) і програмним (software) забезпеченням комп'ютера.

Очевидно, що не існує чіткої межі між прикладним та системним програмуванням, тому що будь-яка прикладна програма тим чи іншим способом взаємодіє з “залізом” комп'ютера, а системна – з користувачем. Однак, при розробці системного програмного забезпечення програміст, як правило, зосереджений на врахуванні специфічних апаратно-залежних особливостей конкретної обчислювальної системи, для якої розробляється програмне забезпечення.



### 3. Мови програмування, що використовуються для створення прикладного програмного забезпечення

**Машинний код** – орієнтована на конкретну апаратну платформу система команд, що безпосередньо виконується її процесором.

Приклад програми, що виводить на екран рядок «Hello, world!» для процесора архітектури x86 (у 16-ти річному форматі):

```
BB 11 01 B9 0D 00 B4 0E 8A 07 43 CD 10 E2 F9 CD 20 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 21
```

**Мова асемблера** – машинозалежна мова програмування низького рівня.

Приклад програма «Hello, world!» для процесора архітектури x86 та операційної системи Linux:

```
section      .text
global      _start                ;must be declared for linker (ld)
_start:     ;tell linker entry point
    mov     edx,len                ;message length
    mov     ecx,msg                ;message to write
    mov     ebx,1                  ;file descriptor (stdout)
    mov     eax,4                  ;system call number (sys_write)
    int     0x80                   ;call kernel
    mov     eax,1                  ;system call number (sys_exit)
    int     0x80                   ;call kernel
section      .data
msg          db  'Hello, world!',0xa ;our dear string
len          equ $ - msg           ;length of our dear string
```

### 3. Мови програмування, що використовуються для створення прикладного програмного забезпечення

**Мова програмування високого рівня** – незалежна від апаратної платформи мова програмування (алгоритмічна мова), розроблена для швидкості та зручності при написанні програм. Основною рисою високорівневих мов є їх абстракція від особливостей конкретної обчислювальної системи, що досягається за рахунок введення смислових конструкцій, що коротко описують такі структури даних та операції над ними, опис яких на машинному кодї (або іншій низькорівневій мові програмування) дуже довгі і складні для розуміння.

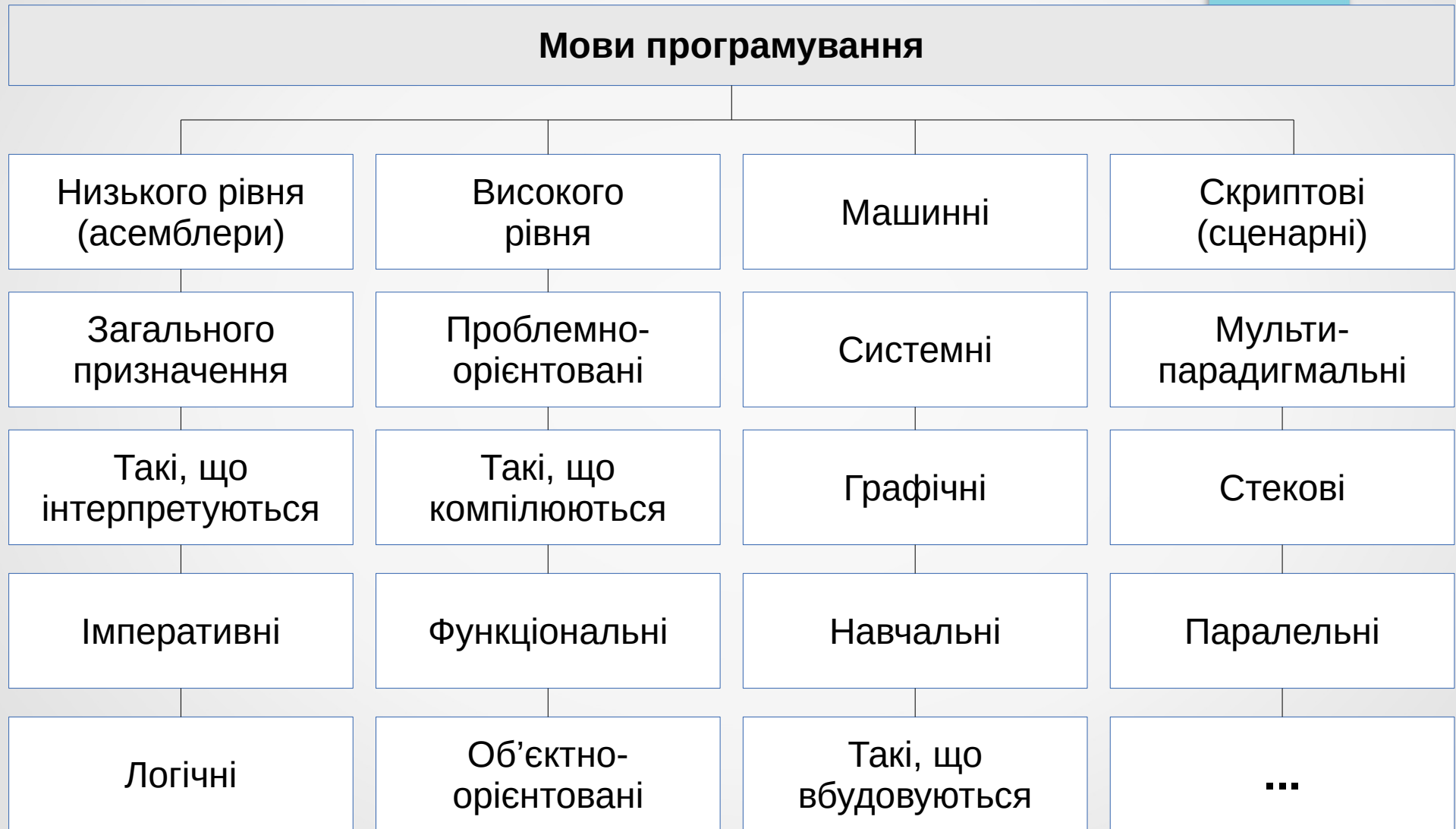
Приклад програми «Hello, world!» мовою програмування C:

```
#include <stdio.h>

int main()
{
    printf("Hello, world\n");
    return 0;
}
```

Для системного програмування найчастіше використовують такі високорівневі мови, як C і C++ (об'єктно-орієнтоване розширення мови C).

### 3. Мови програмування, що використовуються для створення прикладного програмного забезпечення



## 4. Література

1. Керниган Б., Ритчи Д. Язык программирования Си. М.: Вильямс, 2007. 304 с.
2. Прата С. Язык программирования С: Лекции и упражнения. М.: Вильямс, 2006. 960 с.
3. Страуструп Б. Программирование: принципы и практика с использованием С++. М.: ООО Вильямс, 2016. 1328 с.
4. Вандевурд Д., Джосаттис Н.М., Грегор Д. Шаблоны С++. Справочник разработчика. СПб.: Альфа-книга, 2018. 848 с.
5. Мейерс С. Эффективный и современный С++: 42 рекомендации по использованию С++ 11 и С++14. М.: Вильямс, 2016. 304 с.