

## Практична робота №8. Механізм успадкування в C++

**Мета:** отримання практичних навичок при використанні успадкування мовою C++

### Актуалізація опорних знань

1. Що таке успадкування класів?
2. Який синтаксис успадкування класів?
3. Принцип керування доступом елементів класу при успадкуванні?

### Завдання до практичної роботи

1. Створити клас для роботи видавництва, що торгує книгами та аудіо-записами цих книг. Базовий клас *Publication* має містити назву (*title*), автора (*author*) і ціну (*price*) книги.
2. Від базового класу успадковуються ще два класи: клас *Book*, який містить інформацію про кількість сторінок у друкованій книзі (*page\_count*), і клас *Record*, який містить час запису книги в хвилинах (*time*).
3. У кожному з цих трьох класів повинен бути метод *enterData()*, через який можна отримувати дані від користувача з клавіатури, і *displayData()*, призначений для виведення цих даних.
4. Для класу *Book* додатково реалізувати метод, який перевірятиме чи ціна книги менше 200 грн. Для класу *Record* додатково реалізувати метод, який переводитиме хвилини в секунди.
5. Перевірити роботу створеної програми в *int main()*.

### Методичні рекомендації

**Успадкування (*inheritance*)** – це принцип, що дозволяє описати **новий** клас на основі вже **існуючого** (батьківського), при цьому властивості і функціональність батьківського класу запозичуються новим класом.

**Базовий** клас, **батьківський** клас або **клас-предок**, **суперклас** (*base class*, *perent class*, *superclass*) – клас, елементи якого успадковуються.

**Похідний** клас, **дочірній** клас, **клас-нащадок** або **підклас** (*derived class*, *child class*, *subclass*) – клас, який успадковує елементи іншого існуючого класу.

При успадкуванні **похідний** клас може наслідувати (**вміщувати в собі**) **поля** та **методи** базового класу, та **доповнювати** їх своїми власними.

#### Синтаксис успадкування:

```
class Base
{
    //тіло класу;
};

class Derived: модифікатор_доступу Base
{
    //тіло класу;
};
```

#### Особливості успадкування:

1. З одного базового класу можна створювати **багато** похідних класів.
2. Похідний клас може бути **базовим**, тобто батьківським для інших класів.
3. Похідний клас успадковує усі **поля** та **методи** базового класу крім конструкторів, деструктора та операції присвоєння.
4. Допускається не тільки успадкування методів базового класу, але також **додавання** нових і **перевизначення** існуючих методів.

Доступність елементів базового класу з класу-нащадку залежить від:

- **модифікатора доступу в базовому класі;**

Модифікатор доступу у базовому класі	Доступ у самому класі	Доступ у похідному класі	Доступ із «дружніх» класів та функцій	Доступ із інших класів та функцій
<b>public</b>	так	так	так	так
<b>protected</b>	так	так	так	<i>немає</i>
<b>private</b>	так	<i>немає</i>	так	<i>немає</i>

- **модифікатора успадкування.**

Залежно від **модифікатора успадкування**, зазначеного при оголошенні похідного класу, успадкування буває:

- ✓ *відкрите (найпоширеніше);*
- ✓ *захищене;*
- ✓ *закрите (за замовчуванням).*

Модифікатор доступу при оголошенні похідного класу **впливає** на модифікатори доступу у базовому класі:

Вид успадкування (модифікатор)	Модифікатор доступу в базовому класі		
	public	protected	private
Відкрите (public)	<b>public</b>	<b>protected</b>	<i>недоступний</i>
Захищене (protected)	<b>protected</b>	<b>protected</b>	<i>недоступний</i>
Закрите (private)	<b>private</b>	<b>private</b>	<i>недоступний</i>

### **Контрольні запитання:**

1. Які методи класу не успадковуються?
2. Види успадкування.
3. Що означає модифікатор доступу protected?