

Практична робота №1. Класи як основа ООП. Об'єкти типу клас

Мета: ознайомитись на практиці з класами, об'єктами та головними елементами об'єктного підходу, навчитись створювати і використовувати об'єкти типу клас.

Актуалізація опорних знань

1. Поняття ООП.
2. Основні принципи ООП.
3. Що таке клас?
4. Що таке об'єкт?
5. Які бувають модифікатори доступу?

Теоретичні відомості

Класи та об'єкти в C++ є основними концепціями об'єктно-орієнтованого програмування. **Об'єктно-орієнтоване програмування (ООП)** – це методологія програмування, заснована на записі програми у вигляді сукупності об'єктів, кожний з яких є екземпляром певного класу, а класи утворюють ієрархію спадкування.

Клас – визначений програмістом тип даних, що використовується для опису об'єктів. **Об'єкт** – це структурована змінна типу клас, що містить всю інформацію про деякий елемент, який реалізується в програмі. Створені об'єкти на основі одного класу називаються **екземплярами** цього класу. Ці об'єкти можуть мати різну поведінку, властивості, але все одно будуть об'єктами одного класу.

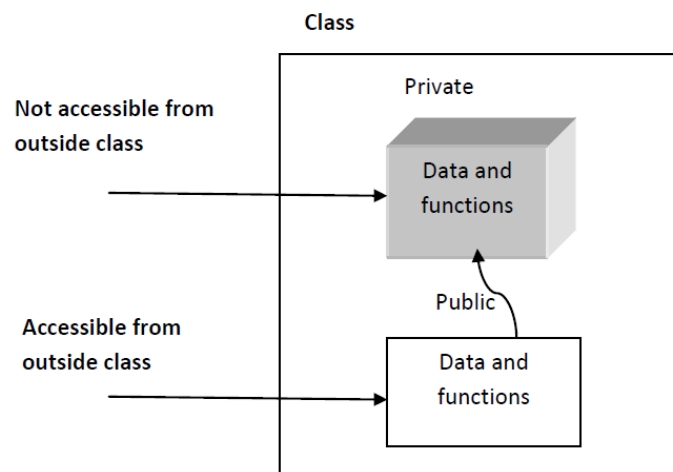
```
class MyClass
{
public:
    // елементи в цій секції доступні з будь-якої частини програми
    MyClass(); // конструктор
    ~MyClass(); // деструктор
protected:
    // елементи в цій секції доступні з класу та його нащадків
private:
    // елементи в цій секції доступні лише з класу;
    // це область доступу за замовчуванням
};
```

Дані (поля) класу – змінні, які оголошені безпосередньо в класі.

Функції-члени (методи, операції) – функції, які описані всередині класу.

Конструктор – це спеціальний метод класу, призначений для ініціалізації елементів класу деякими початковими значеннями.

Деструктор – спеціальний метод класу, який служить для знищення елементів класу.



Завдання до практичної роботи

1. Створити клас *Triangle* з полями *a*, *b* та *c*. Додати конструктор за замовчуванням та з конструктор параметрами. Описати метод *getArea()* для обчислення площі трикутника.
2. Створити клас *Box* з полями *width* (ширина), *length* (довжина) та *height* (висота), а також методами для встановлення та отримання значень полів. Описати метод *Volume()* для обчислення об'єму коробки *Box*. Визначити метод *Compare(Box box)* для порівняння об'ємів двох об'єктів типу *Box*.

Контрольні запитання:

1. Що таке поле класу?
2. Що таке метод класу?
3. Що таке конструктор?
4. Навіщо потрібен конструктор?
5. Як формується ім'я конструктора?
6. Що таке деструктор? Навіщо потрібен деструктор?
7. Як формується ім'я деструктора?
8. Що таке конструктор за замовчуванням?
9. Скільки конструкторів може мати клас?
10. Скільки деструкторів може мати клас?