

## **Лабораторна робота**

**№6**

*Тема: «Розробка програм з користувачькими класами. Робота з класами та об'єктами.»*

**Виконав:** : Шалівський Віталій

**Група:** ALK-43

**Київ 2025р.**

## **Відповіді на контрольні питання.**

### **1. Що називають полями класу?**

Поля класу — це змінні всередині класу, які зберігають стан об'єкта. Вони можуть бути будь-якого типу: прості (int, double тощо), інші класи чи вказівники. Поля описують характеристики об'єкта.

### **2. Що називають методами класу?**

Методи класу — це функції, які визначені всередині класу і призначені для обробки його полів або зовнішніх даних. Методи реалізують поведінку об'єкта класу, тобто алгоритми опрацювання даних.

### **3. Якими властивостями володіють класи?**

Класи володіють такими основними властивостями:

- **Інкапсуляція** — об'єднання даних (полів) і методів в одному блоці; контроль доступу до полів через специфікатори доступу (*private*, *protected*, *public*).
- **Наслідування** — можливість створення похідних класів, що успадковують поля і методи базового класу.
- **Поліморфізм** — здатність методів з однаковою назвою поводитися по-різному для різних класів.

### **4. Що таке наслідування класів?**

Наслідування — це механізм, за допомогою якого один клас (похідний) може використовувати поля та методи іншого класу (базового). Це дозволяє створювати ієрархічні структури класів та повторно використовувати код.

### **5. Що називають поліморфізмом методів?**

Поліморфізм методів — це здатність методів з однаковою назвою в різних класах виконуватися по-різному. Тобто один метод може мати різну реалізацію

## залежно від класу об'єкта. **6. Як в C++ реалізується перевантаження операторів?**

Перевантаження операторів здійснюється шляхом створення спеціальних методів класу типу:

```
<тип> operator<символ>(<формальні параметри>) {  
    /* тіло методу */  
}
```

Це дозволяє змінювати поведінку стандартних операторів (+, -, \*, /, =, () тощо) для об'єктів користувачьких класів.

## **7. Для чого призначений конструктор класу?**

Конструктор класу — це спеціальний метод, який викликається при створенні об'єкта для його ініціалізації.

- Може мати параметри або не мати їх (конструктор за замовчуванням).
- Не повертає значень.
- Використовується для надання початкових значень полям класу.

### Завдання 1

Варіант 11. Створити клас точок. Членами класу є координати точки на площині. Методами класу є: ввід точки з клавіатури; вивід координат точки на екран; обчислення та вивід номера квадранта системи координат, в якому точка знаходитьться. Доповнити клас перевантаженими операціями "+ " (додаються відповідні координати двох точок); "- " (віднімаються відповідні координати двох точок). Написати програму, що демонструє роботу з класом.

### **Опис класу**

## Назва класу: Point

### Поля:

- x — координата по осі X;
- y — координата по осі Y.

### Методи:

- input() — введення координат точки;
- output() — виведення координат точки;
- quadrant() — визначення чверті розташування точки;
- operator+() — додавання двох точок;
- operator-() — віднімання двох точок.

### Код програми:

```
#include <iostream>
#include <windows.h>
using namespace std;

class Point {
private:
    double x, y;

public:
    Point() : x(0), y(0) {}
    Point(double xVal, double yVal) : x(xVal), y(yVal) {}

    void input() {
        cout << "Введіть координати точки (x y): ";
        cin >> x >> y;
    }

    void output() const {
        cout << "(" << x << ", " << y << ")";
    }

    int quadrant() const {
        if (x > 0 && y > 0) return 1;
        if (x < 0 && y > 0) return 2;
        if (x < 0 && y < 0) return 3;
        if (x > 0 && y < 0) return 4;
        return 0; // на осіях
    }

    Point operator+(const Point& other) const {
        return Point(x + other.x, y + other.y);
    }

    Point operator-(const Point& other) const {
        return Point(x - other.x, y - other.y);
    }
};
```

```

int main() {
    // Установка українського кодування для Windows-консолі
    SetConsoleOutputCP(1251);
    SetConsoleCP(1251);

    cout << "==== Демонстрація роботи з класом Точка ====\n\n";

    Point A, B;
    cout << "Введення першої точки:\n";
    A.input();

    cout << "Введення другої точки:\n";
    B.input();

    cout << "\n--- Введені точки ---\n";
    cout << "A = "; A.output(); cout << endl;
    cout << "B = "; B.output(); cout << endl;

    cout << "\n--- Квадранти ---\n";
    cout << "A знаходиться у ";
    switch (A.quadrant()) {
        case 1: cout << "I чверті"; break;
        case 2: cout << "II чверті"; break;
        case 3: cout << "III чверті"; break;
        case 4: cout << "IV чверті"; break;
        default: cout << "на осях координат"; break;
    }
    cout << endl;

    cout << "B знаходиться у ";
    switch (B.quadrant()) {
        case 1: cout << "I чверті"; break;
        case 2: cout << "II чверті"; break;
        case 3: cout << "III чверті"; break;
        case 4: cout << "IV чверті"; break;
        default: cout << "на осях координат"; break;
    }
    cout << endl;

    Point sum = A + B;
    Point diff = A - B;

    cout << "\n--- Операції з точками ---\n";
    cout << "A + B = "; sum.output(); cout << endl;
    cout << "A - B = "; diff.output(); cout << endl;

    cout << "\n==== Кінець роботи програми ====\n";
}

```

```

cout << "B знаходиться у ";
switch (B.quadrant()) {
    case 1: cout << "I чверті"; break;
    case 2: cout << "II чверті"; break;
    case 3: cout << "III чверті"; break;
    case 4: cout << "IV чверті"; break;
    default: cout << "на осях координат"; break;
}
cout << endl;

Point sum = A + B;
Point diff = A - B;

cout << "\n--- Операції з точками ---\n";
cout << "A + B = "; sum.output(); cout << endl;
cout << "A - B = "; diff.output(); cout << endl;

cout << "\n==== Кінець роботи програми ====\n";
return 0;
}

```

Результат виконання програми:

```

Microsoft Visual Studio Debug Console
=====
Демонстрація роботи з класом Точка ====
Введення першої точки:
Введіть координати точки (x y): 3 4
Введення другої точки:
Введіть координати точки (x y): -2 5

--- Введені точки ---
A = (3, 4)
B = (-2, 5)

--- Квадранти ---
A знаходиться у I чверті
B знаходиться у II чверті

--- Операції з точками ---
A + B = (1, 9)
A - B = (5, -1)

==== Кінець роботи програми ====
D:\Desktop\студент\ооп\на 22 10 25\пробний\x64\Debug\пробний.exe (process 11392) exited with code 0 (0x0).
Press any key to close this window . .

```