

## Лабораторна роботи №12

*Створення нових типів даних на основі класів C++. Основи ООП.  
Наслідування. Інкапсуляція. Конструктори і деструктори.*

*Мета роботи: знайомство з основними поняттями та принципами об'єктно-орієнтованого програмування.*

### **Завдання**

Реалізувати базовий клас відповідно до варіанта. Передбачити необхідні функції-члени класу, конструктори. Реалізувати власний деструктор та перевірити його виконання в головній програмі. При проектуванні класу врахувати, що в наступному завданні потрібно буде реалізовувати вказаний у варіанті завдання похідний клас. В головній функції створити декілька екземплярів класу та протестувати роботу усіх реалізованих функцій-членів класу.

### **Варіанти завдань**

**Варіант 1.** Створити клас ПАРА ЦЛИХ ЧИСЕЛ. Визначити потрібні конструктори, функції доступу до полів. Створити похідний клас ЗВИЧАЙНИЙ ДРІБ з полями: чисельник, знаменник. Визначити потрібні конструктори, деструктор, функції введення-виведення та оператори порівняння дробів.

**Варіант 2.** Створити клас ПАРА ЧИСЕЛ. Визначити потрібні конструктори, функції доступу до полів, введення-виведення даних. Створити похідний клас ПРЯМОКУТНИЙ ТРИКУТНИК з полями-катетами. Визначити конструктори з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення площі та гіпотенузи трикутника.

**Варіант 3.** Створити клас ТРІЙКА ЧИСЕЛ. Визначити конструктори, деструктор, функції доступу до полів, введення-виведення та обчислення суми чисел. Створити похідний клас ВЕКТОР з полями-координатами. Визначити конструктори із різним числом параметрів, деструктор, функції доступу до полів, введення-виведення. Перевантажити оператори додавання, віднімання та множення (скалярного) векторів.

**Варіант 4.** Створити клас ПАРА ЧИСЕЛ. Визначити потрібні конструктори, функції доступу до полів, введення-виведення. Створити похідний клас ГЕОМЕТРИЧНА ПРОГРЕСІЯ з полями: перший член та знаменник прогресії. Визначити конструктори і з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення суми для заданої кількості елементів прогресії.

С. М. Ментинський, Я. М. Пелех. Основи програмування на C++.

**Варіант 5.** Створити клас ТРІЙКА ЧИСЕЛ. Визначити конструктори, функції доступу до полів, введення-виведення. Створити похідний клас ПАРАЛЕЛЕПІПЕД з полями-сторонами. Визначити конструктори і з різним числом параметрів, деструктор, функції, введення-виведення, обчислення об'єму та площі поверхні.

**Варіант 6.** Створити клас ТРІЙКА ЦІЛИХ ЧИСЕЛ. Визначити конструктори, функції доступу до полів, введення-виведення чисел. Створити похідний клас ЧАС з полями: година, хвилина і секунда. Визначити конструктори і з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення. В конструкторах та при вводі перевіряти коректність даних. Перевантажити оператори додавання та віднімання для двох моментів часу.

**Варіант 7.** Створити клас ПАРА ЦІЛИХ ЧИСЕЛ. Визначити потрібні конструктори, функції доступу до полів. Створити похідний клас ЗВИЧАЙНИЙ ДРІБ з полями: чисельник, знаменник. Визначити потрібні конструктори, деструктор, функції введення-виведення та оператори додавання, віднімання, множення та ділення дробів.

**Варіант 8.** Створити клас ТРІЙКА ЧИСЕЛ. Визначити конструктори, функції доступу до полів, введення-виведення. Створити похідний клас КВАДРАТНЕ РІВНЯННЯ з полями-коефіцієнтами. Визначити конструктори із різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення коренів рівняння.

**Варіант 9.** Створити клас ПАРА ЧИСЕЛ. Визначити конструктори, деструктор, функції доступу до полів, введення-виведення та обчислення суми чисел. Створити похідний клас ВЕКТОР з полями-координатами. Визначити конструктори із різним числом параметрів, деструктор, функції доступу до полів, введення-виведення. Перевантажити оператори додавання, віднімання та множення (скалярного) векторів.

**Варіант 10.** Створити клас ТРІЙКА ЧИСЕЛ. Визначити потрібні конструктори, функції доступу до полів, та виведення на консоль. Створити похідний клас ТРКУТНИК з полями – довжинами сторін. Визначити конструктори із різним числом параметрів, деструктор, функції доступу до полів, введення-виведення. Перевантажити оператори порівняння двох трикутників, вважаючи, що більшим є трикутник, у якого більший периметр.

**Варіант 11.** Створити клас ПАРА ЧИСЕЛ. Визначити конструктори, функції доступу до полів, виведення чисел та обчислення їх добутку. Створити похідний клас ПРЯМОКУТНИК з полями-сторонами. Визначити конструктори із різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення площі та периметра прямокутника.

**Варіант 12.** Створити клас ПАРА ЧИСЕЛ. Визначити потрібні конструктори, функції доступу до полів, виведення на консоль. Створити похідний клас АРИФМЕТИЧНА ПРОГРЕСІЯ з полями: перший елемент та різниця прогресії. Визначити конструктори із різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення суми для заданої кількості елементів прогресії.

**Варіант 13.** Створити клас ТРІЙКА ЦЛИХ ЧИСЕЛ. Визначити конструктори, функції доступу до полів, введення-виведення чисел, та оператор порівняння двох трійок на рівність. Створити похідний клас ЧАС з полями: година, хвилина і секунда. Визначити конструктори і з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення. В конструкторах та при вводі перевіряти коректність даних. Перевантажити оператори порівняння для двох моментів часу.

**Варіант 14.** Створити клас ТРІЙКА ЦЛИХ ЧИСЕЛ. Визначити конструктори, функції доступу до полів, та виводу на консоль. Створити похідний клас ДАТА з полями: рік, місяць і день. Визначити потрібні конструктори, деструктор, функції доступу до полів та вводу даних з клавіатури, перевантажити оператори порівняння дат.

**Варіант 15.** Створити клас ТРІЙКА ЧИСЕЛ. Визначити конструктори, деструктор, функції доступу до полів, введення-виведення та обчислення суми чисел. Створити похідний клас ВЕКТОР з полями-координатами. Визначити конструктори із різним числом параметрів, деструктор, функції доступу до полів, функції введення-виведення та функцію визначення модуля вектора. Перевантажити оператор множення для отримання векторного добутку двох векторів.

#### *Приклад виконання завдання*

**Варіант 16.** Створити клас ТРІЙКА ЦЛИХ ЧИСЕЛ. Визначити конструктори, функції доступу до полів, та виводу на консоль. Створити похідний клас ДАТА з полями: рік, місяць і день. Визначити потрібні конструктори, деструктор, функції доступу до полів та вводу даних з клавіатури, перевантажити оператори порівняння дат.

```
#include <iostream>
#include <cmath>
using namespace std;
class int_triplet{
protected:
    int first, second, third;
```

С. М. Ментинський, Я. М. Пелех. Основи програмування на C++.

```
public:
    int_triplet();
    int_triplet(int first, int second, int third);
    int get_first()
    {
        return first;
    }
    int get_second()
    {
        return second;
    }
    int get_third()
    {
        return third;
    }
    void print();
    void input(int first, int second, int third);
    //перевірка рівності наслідується похідними класами
    bool operator==(int_triplet other)
    {
        return other.first==this->first &&
            other.second==this->second &&
            other.third==this->third;    }
    bool operator!=(int_triplet other)
    {
        return !(*this == other);
    }
};
```

С. М. Ментинський, Я. М. Пелех. Основи програмування на C++.

```
int main()
{
    int_triplet a(2,3,4), b(1,1,1);
        a.print();
        b.print();
        cout<<endl;
    if(a!=b)
    {
        a.print();
        cout<<" != ";
        b.print();
        cout<<endl;
    }
    b.input(2,3,4); //мають бути рівні
    if(a==b)
    {
        a.print();
        cout<<" == ";
        b.print();
        cout<<endl;
    }

    // Реалізація членів класу "Три числа" int_triplet
    int_triplet::int_triplet()
    {
        first=0;
        second=0;
        third=0;
    }
    int_triplet::int_triplet(int first,
```

```
        int second, int third)
    {
        this->first = first;
        this->second = second;
        this->third = third;
    }

void int_triplet::print()
    {
        cout<<"{ "<<first<<" ";
        <<second<<" "; <<third<<" }";
    }

void int_triplet::input(int first,
        int second, int third)
    {
        this->first = first;
        this->second = second;
        this->third = third;
    }
```