**Лабораторна робота №4**

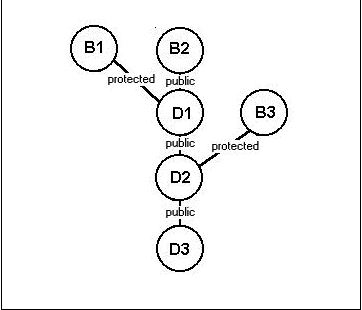
***Побудова множинного наслідування класів***

**Мета.** навчитися організовувати класи та похідні класи з своїми даними та властивостями: визначати структуру класу, типи даних та методів, організовувати необхідні конструктори та деструктори, використовувати специфікатори доступу; будувати ієрархію класів, використовуючи множинне наслідування; визначати та оперувати об’єктами цих класів; отримати практичні вміння та навички проектування та побудови ієрархії класів.

**Постановка задачі:**

**Завдання № 1**

Необхідно побудувати ієрархію класів що відповідає схемі наслідування, наведеній у варіанті завдання. Кожен клас повинен містити конструктор-ініціалізатор для виведення значень.

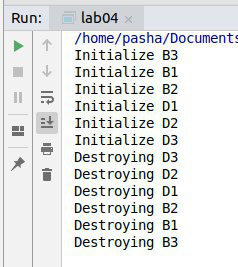


**Код пограми С++:**

#include **<iostream>**#include **<bits/stdc++.h>  
  
using namespace** std;  
  
**class** B1 {  
 **public**:  
 B1() { cout << **"Initialize B1"** << endl; }  
 ~B1() { cout << **"Destroying B1"** << endl; }  
};  
  
**class** B2 {  
 **public**:  
 B2() { cout << **"Initialize B2"** << endl; }  
 ~B2() { cout << **"Destroying B2"** << endl; }  
};

**class** B3 {  
**public**:  
 B3() { cout << **"Initialize B3"** << endl; }  
 ~B3() { cout << **"Destroying B3"** << endl; }  
};  
  
**class** D1 : **protected** B1, **public** B2 {  
 **public**:  
 D1() { cout << **"Initialize D1"** << endl; }  
 ~D1(){ cout << **"Destroying D1"** << endl; }  
};  
  
**class** D2 : **protected** B3, **public** D1 {  
 **public**:  
 D2() { cout << **"Initialize D2"** << endl; }  
 ~D2(){ cout << **"Destroying D2"** << endl; }  
};  
  
**class** D3 : **public** D2 {  
 **public**:  
 D3() { cout << **"Initialize D3"** << endl; }  
 ~D3() { cout << **"Destroying D3"** << endl; }  
};  
  
  
  
**int** main() {  
 D3 d3;  
 **return** 0;  
}

**Результат:**



**Задача 2**

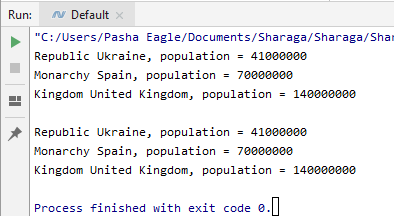
**Варіант 13. республіка, монархія, королівство, держава**

**Реалізувати за допомогою інтерфейсів.**

**Код пограми С#:**

internal class Program  
{  
 interface IDerzhava  
 {  
 uint Population { get; set; }  
   
 string Name { get; set; }  
   
 void Info();  
 }  
  
 class Republic : IDerzhava  
 {  
 public string Name { get; set; }  
  
 public uint Population { get; set; }  
   
 public void Info()  
 {  
 Console.WriteLine("Republic " + Name + ", population = " + Population);  
 }  
  
 public Republic(string name, uint population)  
 {  
 Name = name;  
 Population = population;  
 }  
   
 public Republic(){}  
 }  
   
 class Monarchy : IDerzhava  
 {  
 public string Name { get; set; }  
  
 public uint Population { get; set; }  
   
 public void Info()  
 {  
 Console.WriteLine("Monarchy " + Name + ", population = " + Population);  
 }  
  
 public Monarchy(string name, uint population)  
 {  
 Name = name;  
 Population = population;  
 }  
   
 public Monarchy(){}  
 }  
   
 class Kingdom : IDerzhava  
 {  
 public string Name { get; set; }  
  
 public uint Population { get; set; }  
   
 public void Info()  
 {  
 Console.WriteLine("Kingdom " + Name + ", population = " + Population);  
 }  
  
 public Kingdom(string name, uint population)  
 {  
 Name = name;  
 Population = population;  
 }  
   
 public Kingdom(){}  
 }  
   
   
   
 public static void Main(string[] args)  
 {  
 Republic ukraine = new Republic();  
 Monarchy spain = new Monarchy("Spain", 70000000);  
 Kingdom uk = new Kingdom("United Kingdom", 140000000);  
 ukraine.Name = "Ukraine";  
 ukraine.Population = 41000000;  
   
 ukraine.Info();  
 spain.Info();  
 uk.Info();  
  
 Console.WriteLine();  
   
 IDerzhava[] masiv = new IDerzhava[3];  
  
 masiv[0] = ukraine;  
 masiv[1] = spain;  
 masiv[2] = uk;  
  
 foreach (IDerzhava derzh in masiv)  
 {  
 derzh.Info();  
 }  
 }  
}

**Результат:**



Висновки: отже, на цій лабораторній роботі я навчився організовувати класи та похідні класи з своїми даними та властивостями: визначив структуру класу, типи даних та методів, організував необхідні конструктори та деструктори, використав специфікатори доступу, та описав інтерфейси.