**Лабораторна робота №10**

**Тема**: Успадкування.

**Мета**: Навчитися створювати похідні класи, ієрархію класів.

**Засвоїти поняття** клас; клас базовий; клас похідний; ієрархія класів; об’єкт; динамічний об’єкт;

конструктор; деструктор; конструктор за замовчуванням; конструктор порожній; конструктор копіювання; конструтор перетворення;

ініціалізація; надання значення; успадкування; модель успадкування; спосіб успадкування; агрегація класів.

**Вміти:** створювати класи; описувати потрібний конструктор; створювати похідні класи; створювати ієрархію класів з обранням відповідної моделі та способу успадкування.

**Виконання роботи**

**Індивідуальні завдання** 4 Створити функцію, котрій у якості параметрів передаються імена двох текстових файлів. У тілі функції мають бути описані команди для читання кожного символу з одного файлу та записування його у кінець іншого. Використати цю функцію у головній програмі.

**Лістинг:**

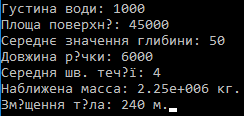
**Індивідуальні завдання** 4 Створити клас, що моделює деяку водойму і містить єдину властивість – густину рідини у ній. Від цього класу успадкувати два класи – клас моря та клас річки. Усі класи повинні містити конструктори для ініціалізації їх властивостей. У класі моря мають бути описані дві властивості, що характеризують площу його поверхні та середнє значення глибини. Також у цьому класі описати функцію наближеного обчислення маси рідини моря (вважати, що море займає об’єм прямого циліндра з відомою площею основи та висотою).

Клас річки повинен містити її довжину, середню швидкість течії та функцію обчислення зміщення тіла, що плаває у ній за деякий проміжок часу. У головній програмі створити два об’єкти моря та три об’єкти річок. Властивості усіх об’єктів вказати при їх створенні.

**Лістинг:**

#include <iostream>

#include <conio.h>

using namespace std;

class Water //опис базового класу

{

public:

int x, y,Ro\_;

public:

Water(int Ro=5) //конструктор базового класу

{

Ro\_=Ro;

}

~Water(){}

};

class Sea : public Water //опис першого похідного класу

{

protected:

float M;

public:

Sea(int S=5, int h=10)

{

x=S;

y=h;

}

void Mass() //опис методу першого похідного класу

{ float V;

V = x\*y;

M=Ro\_\*V;

cout<<"Наближена масса: "<<V<<" кг."<<endl;

}

~Sea() {} //деструктор першого похідного класу

};

class River : public Water //опис другого похідного

//класу

{

protected:

float s;

public:

River(int l=5, int v=10) //конструктор другого

//похідного класу

{

x=l;

y=v;

}

void dislocation() //опис методу другого похідного класу

{ int t=60;

s=y\*t;

cout<<"Зміщення тіла: "<<s<<" м.";

}

~River() {} //деструктор другого похідного класу

};

int main() //основна програма

{

setlocale(0,"");

int Ro,S,h,l,v; //змінні програми

//введення змінних

cout<<"Густина води: ";

cin >> Ro;

cout<<"Площа поверхні: ";

cin >> S;

cout<<"Середнє значення глибини: ";

cin >> h;

cout<<"Довжина річки: ";

cin >> l;

cout<<"Середня шв. течії: ";

cin >> v;

Water Z(Ro); //створення об'єкта базового класу

Z.~Water(); //виклик деструктора для об'єкта базового класу

Sea W(S,h); //створення об'єкта першого похідного класу

W.Mass();

W.~Sea(); //виклик деструктора для об'єкта першого

River R(l, v); //створення об'єкта другого похідного

R.dislocation();

R.~River(); //виклик деструктора для об'єкта другого

getch(); //очікування натискування клавіші

return 0;

}

**Висновок:** на цій лабораторній роботі я навчився створювати похідні класи, ієрархію класів.