***ЛАБОРАТОРНА РОБОТА №4***

**Тема:  *Успадковування класів***

**Мета: ознайомитись зі способами та механізмами успадкування класів та навчитись використовувати їх для побудови об’єктно-орієнтованих програм.**

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Ознайомитись з теоретичним матеріалом про наслідування класів.

2. Напишіть програму згідно наступного завдання 1.

Завдання 1. Уявіть собі видавничу компанію, яка торгує книгами і аудіо-записами цих книг. Створіть клас publication, в якому зберігаються назва (рядок) і ціна (тип float) книги. Від цього класу успадковуються ще два класи: book, який містить інформацію про кількість сторінок у книзі (типу int), і type, який містить час запису книги у хвилинах (тип float). У кожному з цих трьох класів повинен бути метод getdata(), через який можна отримувати дані від користувача з клавіатури, і putdata(), призначений для виведення цих даних. Напишіть функцію main() програми для перевірки класів book і type . Створіть їх об'єкти в програмі і запросіть користувача ввести і вивести дані з використанням методів getdataQ і putdata().

3. Напишіть програму згідно завдання 2.

Завдання 2. До класів з попереднього завдання (попередньо зберігши окремо код) додайте базовий клас sales, в якому міститься масив, що складається з трьох значень типу float, куди можна записати загальну вартість проданих книг за останні три місяці. Включіть в клас методи getdata() для отримання значень вартості від користувача і putdata() для виведення цих цифр. Змініть класи book і type так, щобвони стали похідними обох класів: publication і sales. Об'єкти класів book і type повинні вводити і виводити дані про продажі разом з іншими своїми даними. Напишіть функцію main() для створення об'єктів класів book і type, щоб протестувати можливості введення/виведення даних.

Код програми:

#include "stdafx.h"

#include<iostream>

#include<string>

using namespace std;

class publication

{

string name;

float price;

public:

publication() {}

~publication() {}

void getDate()

{

cout << "Name ";

cin >> name;

cout << "Price ";

cin >> price;

}

void putDate()

{

cout << "Name " << name << endl;

cout << "price " << price << endl;

}

};

class sales

{

float mas[3];

public:

sales() {}

~sales() {}

void getDate()

{

for (int i = 0; i<3; i++)

{

cout << "month " << 1+i << " = ";

cin >> mas[i];

}

}

void putDate()

{

for (int i = 0; i < 3; i++)

{

cout << "month" << 1+i << " = " << mas[i] << " grn" << endl;

}

}

};

class book :public publication, public sales

{

int n;

public:

book() {}

~book() {}

void getDate()

{

publication::getDate();

cout << "n = ";

cin >> n;

sales::getDate();

}

void putDate()

{

publication::putDate();

cout << "n " << n << endl;

sales::putDate();

}

};

class type :public publication ,public sales

{

string time;

public:

type() {}

~type() {}

void getDate()

{

publication::getDate();

cout << "time = ";

cin >> time;

sales::getDate();

}

void putDate()

{

publication::putDate();

cout << "time " << time << endl;

sales::putDate();

}

};

int main()

{

book obj1;

obj1.getDate();

obj1.putDate();

type obj2;

obj2.getDate();

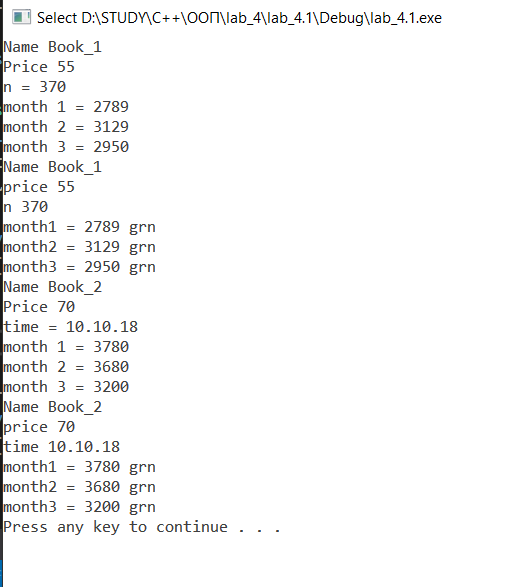
obj2.putDate();

system("pause");

return 0;

}

Результат програми:



## ****ІНДИВІДУАЛЬНЕ ЗАВДАННЯ****

## Створити клас ЧОТИРИКУТНИК з полями – координатами вершин. Визначити конструктори, деструктор, функції доступу до полів, введення-виведення та обчислення добутку чисел. Створити похідний клас ТРАПЕЦІЯ. Визначити конструктори за замовчуванням і з різним числом параметрів, деструктор, функції доступу до полів, введення-виведення, обчислення площі трапеції.

Код програми:

#include "stdafx.h"

#include<iostream>

#include<string>

using namespace std;

class quadrangle

{protected:

double a, b;

public:

quadrangle(){}

~quadrangle() {}

void getDate()

{

cout << "Write a ";

cin >> a;

cout << "Write b ";

cin >> b;

}

double operation()

{

double ab;

ab = a \* b;

return ab;

}

void putDate()

{

cout << "A = " << a << endl;

cout << "B = " << b << endl;

cout << "Sum = " << operation() << endl;

}

};

class trapeze:public quadrangle

{

double h;

public:

trapeze(){}

trapeze(double \_a, double \_b, double \_h) { a = \_a; b = \_b; h = \_h; }

~trapeze() {}

void getDate()

{

quadrangle::getDate();

cout << "Write h ";

cin >> h;

}

double S()

{

double S;

S = 0.5\*(a + b)\*h;

return S;

}

};

int main()

{

trapeze obj1;

trapeze obj2(3, 4, 7);

obj1.getDate();

cout << "S1 = " << obj1.S() << endl;

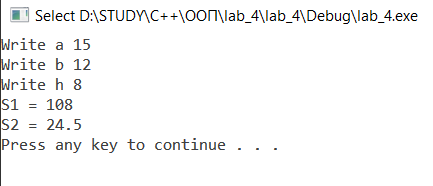
cout << "S2 = " << obj2.S() << endl;

system("pause");

return 0;

}

Результат програми:



**Висновок**: **:**ознайомився зі способами та механізмами успадкування класів та навчився використовувати їх для побудови об’єктно-орієнтованих програм.