Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**Звіт з лабораторної роботи №6**

на тему:

**“ Спадкування”**

**Виконала:**

ст. гр. КІ-15

Аркавенко Л.С.

**Перевірив:**

асист. каф. ЕОМ

Козак Н. Б.

Львів – 2020

**Мета роботи:** познайомитися із спадкуванням класів.

**Теоретична частина:**

Спадкування– це механізм, за допомогою якого один клас може одержувати атрибути та функціональність іншого. Спадкування дозволяє створювати ієрархію класів. При створенні нового класу, що повністю дублює існуючий клас і дещо розширяє його новими властивостями і функціональністю програміст може не дублювати існуючий клас і дописувати в нього розширену функціональність, а вказати, що новий клас є спадкоємцем елементів попередньо визначеного класу і визначити у ньому лише необхідну нову функціональність. В цьому випадку існуючий клас, функціональність якого розширюється у новому класі, називається базовим класом(base class). Новостворений клас називається похідним класом(derived class), або спадкоємцем. Кожен похідний клас може бути використаним у ролі базового класу для майбутніх похідних класів створюючи при цьому дерево спадкування, яке ще називають ієрархією спадкування класів (class hierarchy).Спадкування прийнято відображати у вигляді графу (дерева)у напрямку зверху-вниз. При цьому клас, що є у самому верху є самим першим базовим класомі називається кореневим класом або коренем дерева спадкування класів. Похідний клас, через проміжний, може наслідувати характеристики базового класу. У цьому випадку говорять, що базовий клас є непрямим базовим класом (indirect base class) для похідного. Зокрема, корінь дерева наслідувань є непрямим базовим класом для усіх класів, які знаходяться нижче першого рівня ієрархії.

Клас, який При одиночному спадкуванні(single inheritance) клас породжується одним базовим класом. При множинному спадкуванні(multiple inheritance) похідний клас успадковує властивості декількох базових класів, причому можлива ситуація коли один базовий клас буде успадкований кілька разів по кількох гілках. При створенні об'єкта похідного класу в пам'яті зберігаються копії усіх класів, які становлять вітку, що породила даний клас.

**Індивідуальне завдання:**

Створити абстрактний базовий клас і похійдний від нього клас, які реалізують модель предметної області згідно варіанту.Кожен клас має мати мінімум 3 власні елементи даних один з яких створюється динамічно, методи встановлення і читання характеристик елементів-даних класу (Setі Get), та мінімум 2 абстрактні методи обробки данихі мінімум2 методи обробки даних у похідному класі. Крім цього клас має містити перевантаження оператора присвоєння, конструкторів по замовчуванню і копіювання та віртуальний деструктор. Для розроблених класів реалізуватипрограму-драйвер, яка демонструє роботу класів.

Магазин

**Хід роботи:**

**Код програми:**

#include <iostream>

using namespace std;

class Shop {

protected: char \*name = new char[30];

double worktime;

double opentime,closetime;

public:

Shop() {

opentime = 8;

closetime = 22;

worktime = closetime - opentime;

}

Shop(double opentime, double closetime) {

this->closetime = closetime;

this->opentime = opentime;

}

virtual ~Shop() {

delete[] name;

}

void set() {

cout << "Enter name shop: ";

cin >> name;

}

virtual void get() {

cout << "Name shop: " << name << endl;

cout << "Time open: " << opentime << endl;

cout << "Time close: " << closetime << endl;

cout << "Work time: " << worktime << endl;

}

void Worktime() {

worktime = closetime - opentime;

}

Shop operator = (const Shop &a) {

this->name = a.name;

}

};

class Supermarket :

public Shop {

private:

double vuruchka, prubutok;

int cass;

public: Supermarket() {

vuruchka = 100;

prubutok = vuruchka / 10;

cass = 1;

}

Supermarket(int cass) {

this->cass = cass;

}

void set()

{

cout << "Enter number cass: ";

cin >> cass;

}

void get()

override {

cout << "Profit today: " << prubutok << endl;

cout << "Revenue for today: " << vuruchka << endl;

}

void Prubutoc() {

prubutok = vuruchka / 10;

}

void salary() {

double salary;

salary = prubutok / cass;

}

};

int main() {

Shop a(9.45, 18.00);

Supermarket s;

a.set();

a.get();

s.set();

s.get();

a.Worktime();

s.Prubutoc();

s.salary();

s.Worktime();

system("pause");

}

****

Рис 1. Програма та консольний вивід до завдання.