ЛАБОРАТОРНА РОБОТА №1Тема: Вивчення понять класу.  
Мета: Закріпити базові знання про клас. Навчитись створювати класи  
засобами мови С++

1.

/\*Створіть клас Int, що імітує стандартний тип int. Єдине поле цього класу

повинно мати тип int. Створіть методи, які будуть встановлювати значення поля рівним

нулю, ініціалізувати його цілим значенням, виводити значення поля на екран і складати два

значення типу Int\*/

#include "stdafx.h"

#include<iostream>

using namespace std;

class Int {

int a ;

int b ;

int c;

public:

void scan() {

a = b = 0;

cout << "A and B before intialization " << a <<" "<< b<<endl;

cout << "write a and b " << endl;

cin >> a >> b;

}

void show() { cout << "A=" << a<<endl;

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

}

void sum() {

cout << "Sum=" << a + b << endl;

}

};

int main()

{

Int objA, objB;

objA.scan();

objA.show();

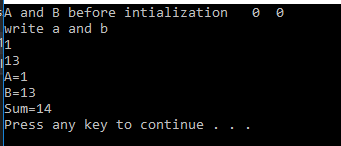
objA.sum();

system("pause");

return 0;

}

Вивід програми



2.

/\*Уявіть пункт для прийому платежів за проїзд по автостраді. Кожна проїжджаюча машина повинна заплатити за проїзд 50 центів, однак частина машин платить за проїзд, а частина проїжджає безкоштовно. У касі ведеться облік числа проїхавших машин і сумарна виручка від плати за проїзд.

Створіть модель такої каси за допомогою класу Kasa. Клас повинен містити два поля. Одне з них, типу unsigned int, призначене для обліку кількості проїхали автомобілів, а друге, що матиме тип double, міститиме сумарну виручку від оплати проїзду. Конструктор повинен ініціалізувати обидва поля нульовими значеннями. Метод payingCar () інкрементує число машин і збільшує на 0,50 сумарну виручку. Інший метод, nорауСаг (), збільшує на одиницю число автомобілів, але залишає без зміни виручку. Метод display () виводить обидвазначення на екран. Там, де це можливо, зробіть методи константними. Створіть програму, яка продемонструє роботу класу. Програма повинна запропонувати користувачеві натиснути одну клавішу для того, щоб зімітувати оплату водієм, і іншу клавішу, щоб зімітувати несумлінного водія. Натискання клавіші Esc повинно привести довидачі поточних значень кількості машин і виручки, і до завершення програми.\*/

#include"stdafx.h"

#include<iostream>

#include<conio.h>

using namespace std;

class Kasa{

private:

unsigned int numberCar;

double money;

public:

Kasa(int \_numberCar, double \_money) {

numberCar = \_numberCar;

money = \_money;

}

void payingCar() {

numberCar++;

money += 0.5;

}

void nopayingCar() {

numberCar++;

}

void display() {

cout<< numberCar << " -" << money << "$" << endl;

}

};

int main()

{

Kasa kasa(0, 0);

int c;

while (1) {

c = \_getch();

if (c == ' ')

{

kasa.nopayingCar();

}

if (c == 'e' || c == 'E')

{

kasa.payingCar();

}

if (c == 'q' || c == 'Q')

break;

kasa.display();

}

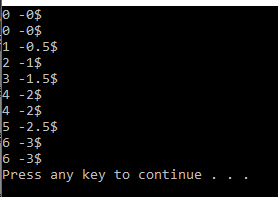
kasa.display();

system("pause");

return 0;

}

Вивід програми.



3. /\*Створіть клас з ім'ям time, що містить три поля типу int, призначених для зберігання годин, хвилин і секунд. Один з конструкторів класу повинен ініціалізувати поля нульовими значеннями, а інший конструктор - заданим набором значень. Створітьметод класу, який буде виводити значення полів на екран у форматі 11:59:59, і метод, складає значення двох об'єктів типу time, переданих в якості аргументів.\*/

#include "stdafx.h"

#include<iostream>

using namespace std;

class time {

int minuts;

int hours;

int seconds;

public:

time() { minuts = seconds = hours = 0; }

time(int \_minuts ,int \_hours , int \_seconds)

{

minuts = \_minuts;

hours = \_hours;

seconds = \_seconds;

}

void get() {

cout << hours <<":"<< minuts <<":"<<seconds<<endl;

}

void Sumtime(time time1,time time2) {

minuts = time1.minuts + time2.minuts;

hours = time1.hours + time2.hours;

seconds = time1.seconds + time2.seconds;

}

};

int main()

{

time time1,time2(12,4,59);

time time3;

time3.Sumtime(time1,time2);

time1.get();

time2.get();

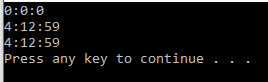
time3.get();

system("pause");

return 0;

}

Вивід програми.



4. /\*Атрибут first – дробове число, координата точки х на площині; атрибут second –дробове число, координата точки у на площині. Реалізуватиметод distance() –відстань точки від початку координат.\*/

#include "stdafx.h"

#include <iostream>

#include "math.h"

using namespace std;

class Distance {

double first;

double second;

public:

Distance(){

}

void read() {

cin >> first;

cin >> second;

}

double distance() {

double d = sqrt(pow(first , 2) + pow(second , 2));

cout << d << endl;

return d;

}

};

int main()

{

Distance point1;

point1.read();

point1.distance();

system("pause");

return 0;

}

Вивід



Висновок: Закріпив базові знання про клас. Навчився створювати класи  
засобами мови С++.