***ЛАБОРАТОРНА РОБОТА №1***

**Тема:** Вивчення базових понять класу.

**Мета:** Закріпити базові знання про клас. Навчитись створювати класи засобами мови С++

***ПОРЯДОК ВИКОНАННЯ РОБОТИ***

**Завдання 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.

Напишіть програму, в якій будуть створені три об'єкти класу Int, два з яких будуть ініціалізованими. Додайте два ініціалізованних об'єкта, надайте результат третьому, а потім відобразіть результат на екрані.

**Завдання 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 обидва значення 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 до видачі поточних значень кількості машин і виручки, і до завершення програми.

**Завдання 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. Створіть метод 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.

У функції main () слід створити два ініціалізованих об'єкта (подумайте, чи повинні вони бути константними) і один неініціалізованний об'єкт. Потім складіть два ініціалізованих значення, а результат надайте третьому об'єкту і виведіть його значення на екран. Де можливо, зробіть методи константними.

**Завдання 4.** \* У кожному завданні потрібно реалізувати клас. У програмі обов'язково повинні бути продемонстровані різні способи створення об'єктів і масивів об'єктів. Програма повинна демонструвати використання всіх функцій і методів.

У всіх завданнях обов’язково повинні бути присутні:

 метод ініціалізації Init( ), метод повинен контролювати значення аргументів на коректність;

 введення з клавіатури Read( );

 виведення на екран Display( ).

Варіант 9: Атрибут first – ціле додатнє число , години; Атрибут second – ціле додатнє число, хвилини. Реалізувати метод minutes() – приведення часу в хвилини.

Код програми

**#include "stdafx.h"**

**#include <iostream>**

**#include <conio.h>**

**using namespace std;**

**class Int{**

**private:**

**int value;**

**public:**

**Int(int defaultVal) { value = defaultVal; }**

**Int():Int(0){}**

**void Set(int val) { value = val; }**

**int Get() { return value; }**

**void Show() { cout << value << endl; }**

**void ResetInt() { value = 0; }**

**static int Add(Int n1, Int n2) { return n1.Get() + n2.Get(); }**

**};**

**class Kasa**

**{**

**unsigned int carsAmount;**

**double suma;**

**public:**

**const Kasa() { carsAmount = 0; suma = 0; }**

**const void payingCar() { carsAmount++; suma += 0.5; }**

**const void nopayCar() { carsAmount++; }**

**const void display() { cout << "Cars amount = " << carsAmount << ", suma = " << suma << endl; }**

**};**

**class Time**

**{**

**int hours;**

**int minutes;**

**int seconds;**

**void SetHours(int val) { hours = val % 12; }**

**void SetMinutes(int val) { minutes = val % 60; if(val > 59) SetHours(hours + val/60); }**

**void SetSeconds(int val) { seconds = val % 60; if (val > 59) SetMinutes(minutes + val / 60); }**

**public:**

**const Time(int \_hours, int \_minutes, int \_seconds)**

**{**

**SetHours(\_hours);**

**SetMinutes(\_minutes);**

**SetSeconds(\_seconds);**

**}**

**const Time() :Time(0, 0, 0) {}**

**const void Print() { cout << hours << ":" << minutes << ":" << seconds << endl; }**

**const void Add(Time t1, Time t2)**

**{**

**SetHours(t1.hours + t2.hours);**

**SetMinutes(t1.minutes + t2.minutes);**

**SetSeconds(t1.seconds + t2.seconds);**

**}**

**};**

**class NewTime**

**{**

**int hours;**

**int minutes;**

**public:**

**NewTime(int hours, int minutes) { InitHours(hours); InitMinutes(minutes); }**

**NewTime():NewTime(0,0) {}**

**void InitHours(int val) { if (val >= 0 && val < 24) hours = val; else hours = 0; }**

**void InitMinutes(int val) { if (val >= 0 && val < 60) minutes = val; else minutes = 0; }**

**void Display() { cout << "Time = " << hours << ":" << minutes << endl; }**

**void Read()**

**{**

**cout << "Write hours: ";**

**int val = 0;**

**cin >> val;**

**InitHours(val);**

**cout << "Write minutes: ";**

**val = 0;**

**cin >> val;**

**InitMinutes(val);**

**}**

**int Minutes() { return hours \* 60 + minutes; }**

**};**

**int main()**

**{**

**//------------------Z1----------------**

**Int n1(5), n2(7), n3;**

**n3.Set(Int::Add(n1, n2));**

**n3.Show();**

**//------------------Z2----------------**

**char symb;**

**cout << endl << "Try kasa: " << endl;**

**Kasa kasa;**

**while ((symb = getch()) != 27)**

**{**

**if (symb == 'p') kasa.payingCar();**

**else if (symb == 'n') kasa.nopayCar();**

**}**

**kasa.display();**

**//-------------------Z3----------------**

**cout << endl;**

**Time t1(2, 8, 64);**

**Time t2(22, 61, 2);**

**t1.Print();**

**t2.Print();**

**Time t3;**

**t3.Add(t1, t2);**

**t3.Print();**

**//--------------------Z4---------------**

**cout << endl;**

**NewTime nt1;**

**NewTime \*nt2 = new NewTime(3, 34);**

**NewTime mas1[2];**

**NewTime mas2[] = {NewTime(4,5), NewTime(15,20) };**

**nt1.Read();**

**mas1[0].InitHours(8);**

**mas1[1].InitMinutes(20);**

**mas2[1].Display();**

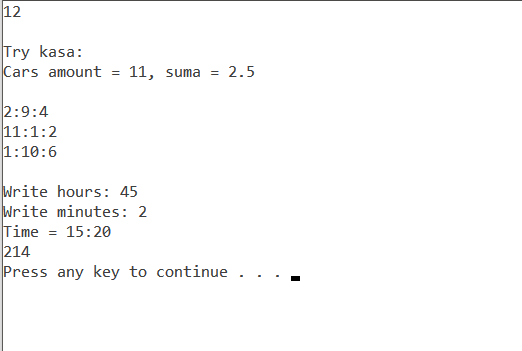
**cout << nt2->Minutes() << endl;**

**system("pause");**

**return 0;**

**}**

Вивід програми



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