Міністерство науки і освіти України

коломийський політехнічний коледж

національного університету «Львівська політехніка»

**Циклова комісія: Інженерія програмного забезпечення**

**Практична робота №3**

**З дисципліни**

**«Об’єктно-орієнтоване програмування»**

Виконав студент групи П-31

Мотрунич Владислав Іванович

Коломия 2018

Теорія

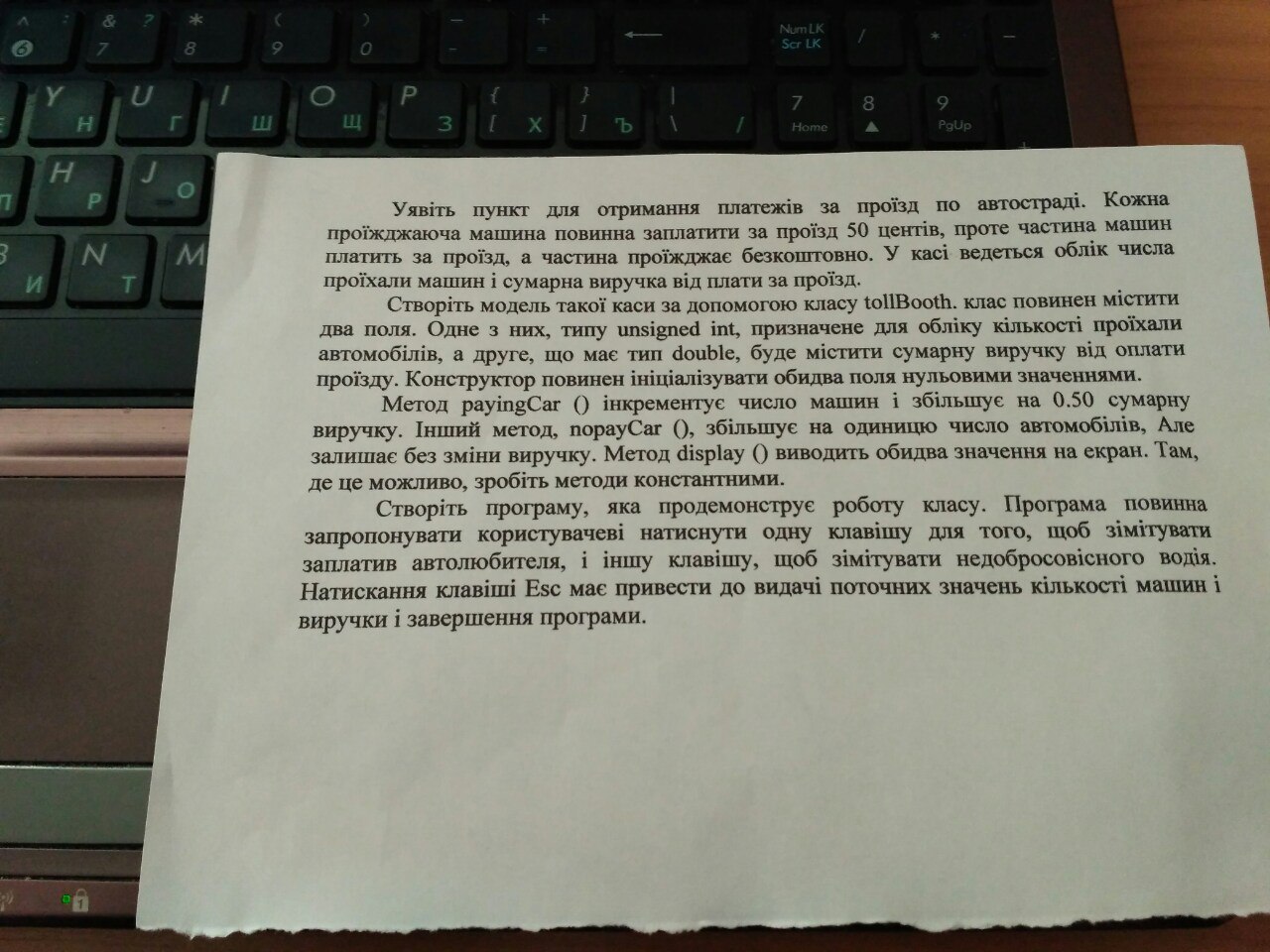
Константні методи не змінюють поля свого класу.

Константні поля не змінюються, як і константні класи.

Слово const треба використовувати коли потрібно захистити поле від змін.

Так в аргументах треба вказати const перед потрібним полем щоб захистити його від змін у даному методі

Задача



#include "stdafx.h"

#include <iostream>

#include <conio.h>

#include <stdlib.h>

using namespace std;

const char ESC = 27;

class tollBooth

{

public:

tollBooth() :passed(0), payed(0) {}

void AddPassed()

{

passed++;

}

void AddPayed()

{

payed+=0.5;

}

int GetPassed()

{

return passed;

}

double GetPayed()

{

return payed;

}

private:

unsigned int passed;

double payed;

};

static class Controller

{

public:

const static void PayingCar(tollBooth &Palka)

{

Palka.AddPassed();

Palka.AddPayed();

}

const static void NoPayingCar(tollBooth &Palka)

{

Palka.AddPassed();

}

};

static class View

{

public:

const static void Display(tollBooth Palka)

{

cout << "\n passed = " << Palka.GetPassed() << endl;

cout << "\n payed = " << Palka.GetPayed() << endl;

}

};

int main()

{

tollBooth toll;

char a;

cout << "enter 1 when payed or 0 when not payed "<<endl;

do

{

a = \_getche();

if (a == '1')

{

Controller::PayingCar(toll);

}

else if (a == '0')

{

Controller::NoPayingCar(toll);

}

} while (a != ESC);

cout << endl;

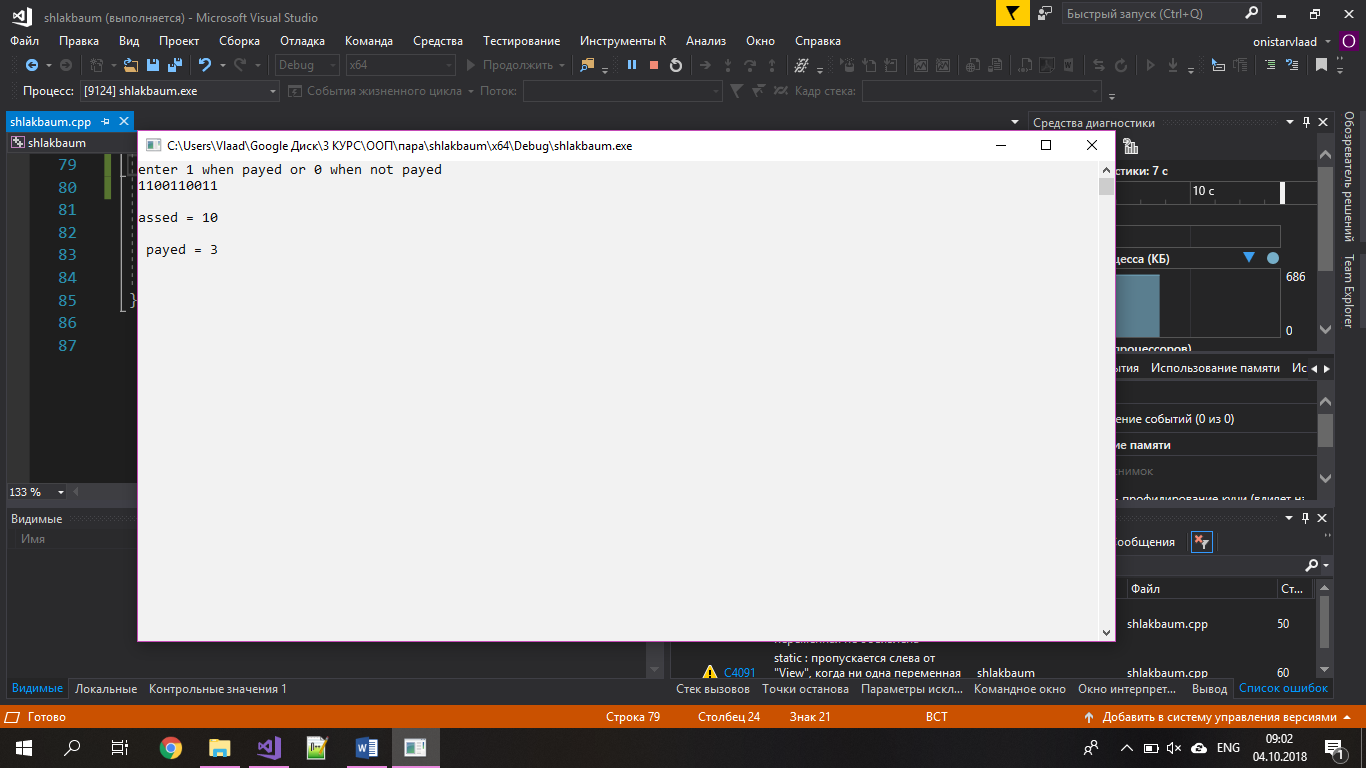
View::Display(toll);

\_getch();

\_getch();

return 0;

}



Висновок

Дана практична робота навчила мене працювати з словом const.

Його можна додавати до методів які не повинні змінити нічого, або до полів, які не повинні бути змінені.