**КН-321 Макарчук А.В.**

**Лабораторна робота №1**

**Тема:** Класи та структури

**Мета:**  Навчитись створювати класи засобами мови С++, виконати завдання згідно варіанту

**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.

**КОД ПРОГРАМИ**

#include <iostream>

#include <QCoreApplication>

#include <iostream>

class **IntA**{

private:

int m\_a;

public:

**IntA**(){m\_a = 0;}

**IntA**(int a){ m\_a = a;}

void **setNum**(){

std::cout << "Enter the number a";

std::cin >> m\_a; }

void **toNull**(){

m\_a = 0; }

void **showA**() const{

std::cout << "a is "<< m\_a;

}

int **getA**(){

return m\_a;

}

void **sum**(IntA b, IntA c){

m\_a = b.getA() + c.getA();

}

};

int **main**(int argc, char \*argv[])

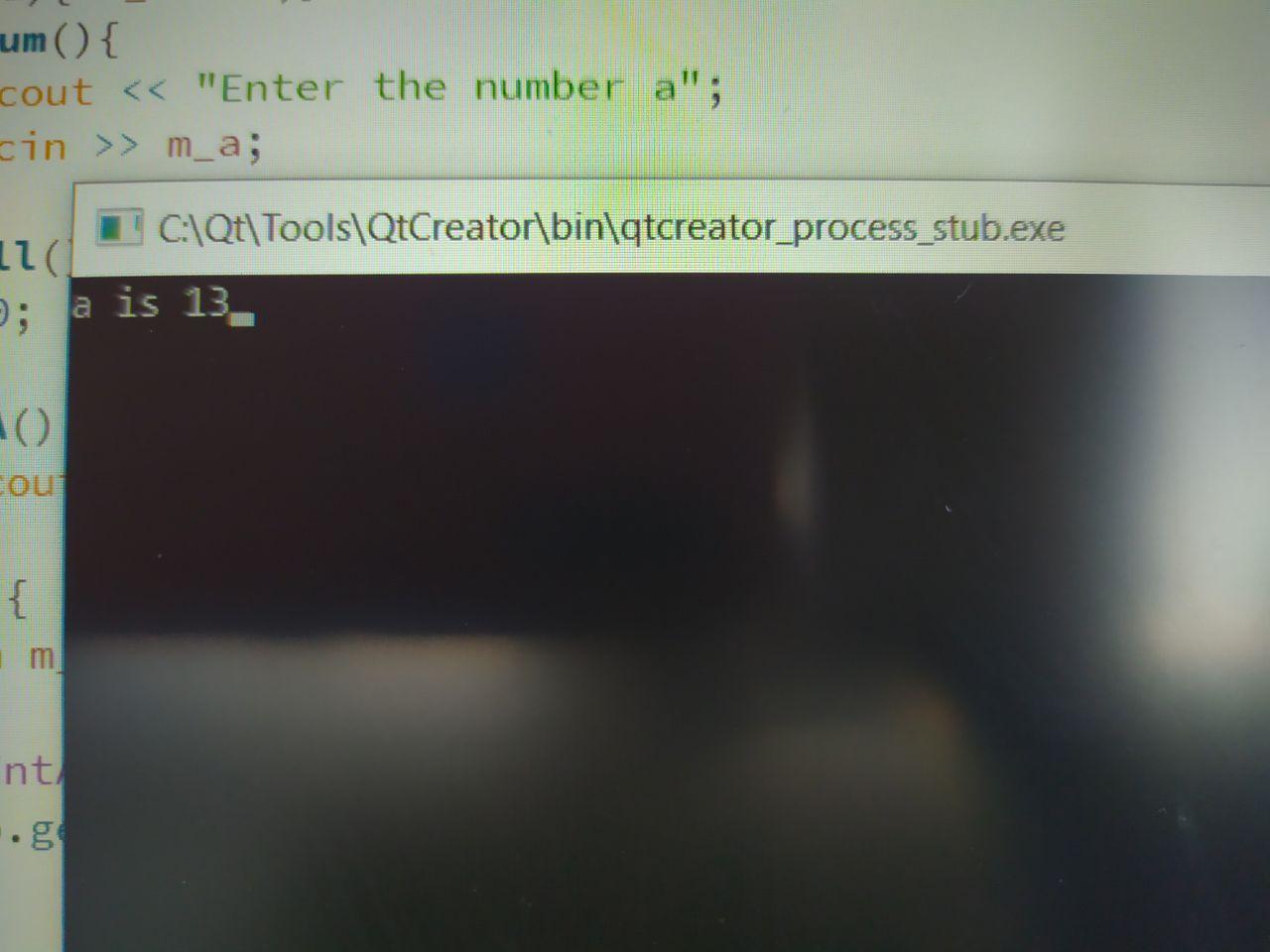
{ QCoreApplication a(*argc*, argv);

IntA n1(5),n2(8),n3; n3.sum(n1, n2);

n3.showA();

return a.exec(); }

**РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ**



**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.

**КОД ПРОГРАМИ**

#include <QCoreApplication>

#include <iostream>

#include <conio.h>

#define ESC 27

class **Kasa**

{

private:

unsigned int countOfCars;

double allMoney;

public:

**Kasa**(){

countOfCars = 0;

allMoney = 0;}

void **nopaingCar**()

{ countOfCars += 1; }

void **paingCar**()

{ countOfCars++; allMoney += 0.5; }

void **display**() const {

std::cout << "All cars:" << countOfCars << std::endl; std::cout << "All money:" << allMoney << std::endl; } };

int **main**(int argc, char \*argv[])

{ QCoreApplication a(*argc*, argv);

Kasa car;

std::cout << "let's get started!!To show statistic click ESC,\n to add payed car click z,\n to add not payed car click x\n";

while(true)

{ int c = 0; switch (c = \_getch())

{ case 122: // its z

std::cout << "z is clicked\n";

car.paingCar();

break;

case 120: //its x

std::cout << "x is clicked\n";

car.nopaingCar();

break;

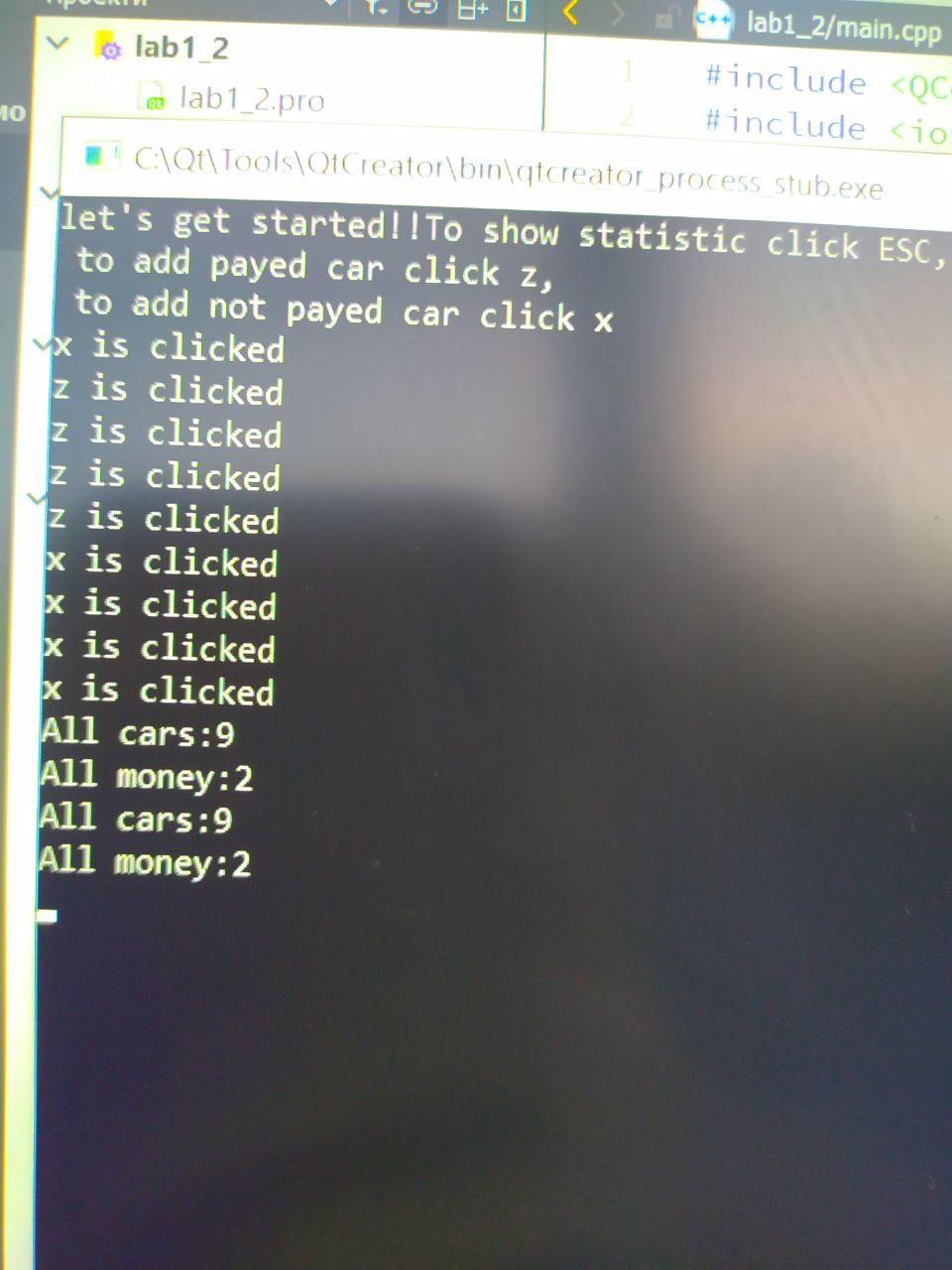
case ESC:

car.display(); }

}

return a.exec();}

**РЕЗУЛЬТАТ ВИКОНАННЯ 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.

**КОД ПРОГРАМИ**

#include <QCoreApplication>

#include <iostream>   
class **Time**{

private:

int m\_seconds;

int m\_minute;

int m\_hour;

public:

**Time**(int a, int b, int c) : m\_seconds(a), m\_minute(b),m\_hour(c) {}

**Time**() : m\_seconds(0), m\_minute(0),m\_hour(0) {}

friend Time operator+(const Time &t1, const Time &t2);   
 void **showtime**() const{

std::cout << "time:: "<< m\_seconds << "::" << m\_minute << "::" << m\_hour;

}   
};

Time operator+(const Time &t1, const Time &t2){

return Time(t1.m\_seconds+ t2.m\_seconds, t1.m\_minute+t2.m\_minute, t1.m\_hour + t2.m\_hour); }

int **main**(int argc, char \*argv[]) {

Time const n1(10,20,15),n2(30,10,5);

Time n3;

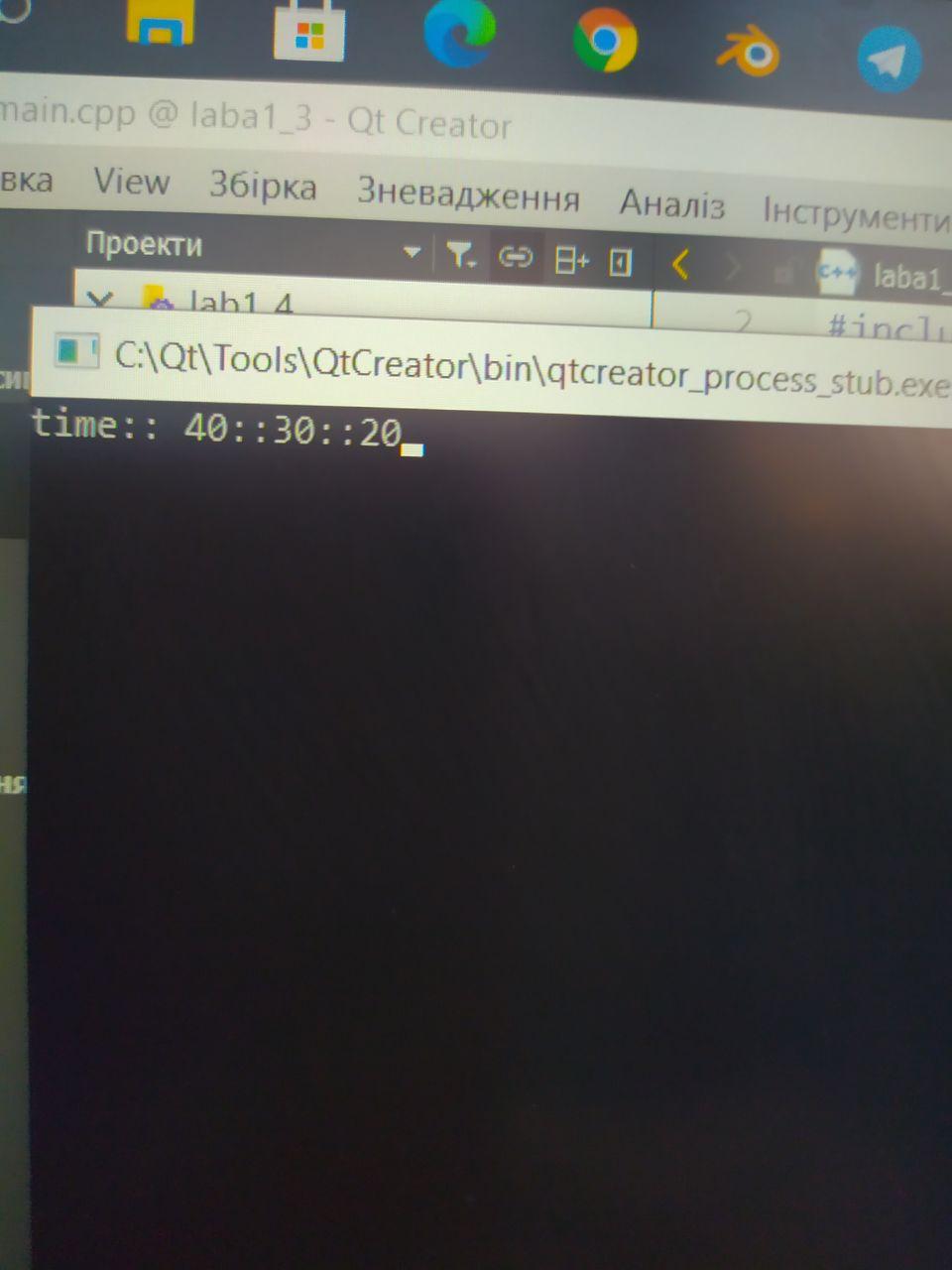
QCoreApplication a(*argc*, argv);

n3 = n1 + n2; n3.showtime();

return a.exec();

}

**РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ**



**Завдання4:** У даній лабораторній роботі в 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 №7:**

Атрибут first – дробове число , ліва границя діапазону ; атрибут second – дробове число, права границя діапазону. Реалізувати метод rangecheck()- перевірку заданого числа на приналежність діапазону.

**КОД ПРОГРАМИ**

#include <QCoreApplication>

#include <iostream>

class **DiapasonN**{

private:

double m\_first;

double m\_second;

public:

**DiapasonN**(): m\_first(0),m\_second(0){}

**DiapasonN**(double a, double b): m\_first(a),m\_second(b){}

void **rangeCheck**(double a){

if (a>= m\_first && a<= m\_second)

std::cout << "A is in diapason\n";

else

std::cout << "A isn't in diapason\n";

}

void **init**() const{

std::cout << "Checking if your diapason is correct\n";

if (m\_first < m\_second)

std::cout << "It's correct!\n";

else {

std::cout << "Oops your diapason isn't correct.Try again\n";

read();

}

}

void **display**() const

{

std::cout << "The diapason is from " << m\_first << " to " << m\_second;

}

void **read**(){

std::cout << "Enter your Diapason from ";

std::cin >> m\_first;

std::cout <<" to ";

std::cin >> m\_second;

init();

}

};

double **getNum**(){

double a; std::cout << "Enter the number, what you want to check";

std::cin >> a; return a;

}

int **main**(int argc, char \*argv[])

{

DiapasonN k, n(5.0,8.0);

const DiapasonN s(5.0,8.0);

k.read();

k.display();

double b = getNum();

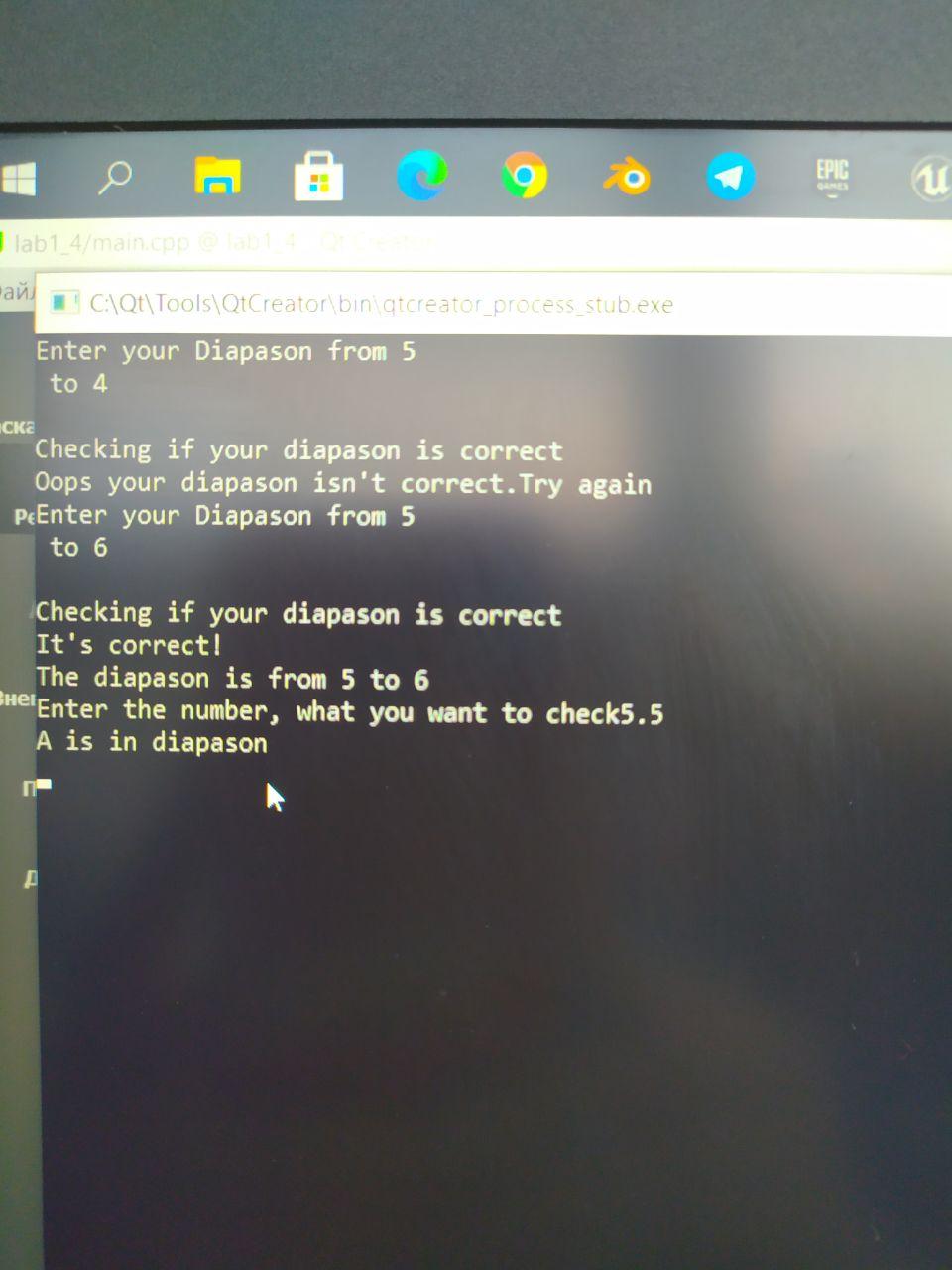
k.rangeCheck(b);

QCoreApplication a(*argc*, argv);

return a.exec();

}

**РЕЗУЛЬТАТ ВИКОНАННЯ ПРОГРАМИ**



**ВИСНОВОК:** На даній лабораторній роботі я навчилася створювати класи засобами мови С++, виконати завдання згідно варіанту