**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

### **Федеральное государственное автономное образовательное учреждение высшего образования**

**«Санкт-Петербургский государственный университет  
аэрокосмического приборостроения»**

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «Классы, конструкторы, деструкторы» |
| по дисциплине: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4131 |  |  |  | В.А. Алексеев |
|  |  |  | подпись, дата |  | инициалы, фамилия |

1. Задание

В работе необходимо реализовать класс stopwatch для имитации работы секундомера. Используйте конструктор для начальной установки секундомера в нуль. Образуйте две функции-члена класса start() и stop() соответственно для запуска и остановки секундомера. Включите в класс и функцию-член show() для вывода на экран величины истекшего промежутка времени (время привести в секундах). Также используйте деструктор для автоматического вывода на экран времени, прошедшего с момента создания объекта класса stopwatch до его удаления. В классе должны быть предусмотрены:

-конструктор для установки начальных значений полей,

-конструктор по умолчанию,

-конструктор копирования,

-деструктор.

Поля класса должны иметь спецификатор доступа private. Доступ к полям осуществляется через public методы. В функции main() создается не менее 3 объектов класса (с использованием всех конструкторов) и осуществляется вызов методов класса.

1. Листинг

Header.h

#pragma once

#include <ctime>

class stopwatch {

private:

time\_t seccreate;

time\_t secondstart;

time\_t secondstop;

time\_t additionalsec;

char name[10];

public:

stopwatch(time\_t, char\*);

stopwatch(const stopwatch&);

stopwatch();

void start();

void stop();

void show();

~stopwatch();

}

Source.cpp

#include "Header.h"

#include <iostream>

#include <ctime>

#include <windows.h>

#include <conio.h>

using namespace std;

stopwatch::stopwatch(time\_t additionalsec, char\* ame) {

seccreate = time(NULL);

strcpy\_s(name, ame);

this->additionalsec = additionalsec;

}

stopwatch::stopwatch() {

seccreate = time(NULL);

strcpy\_s(name, "first");

additionalsec = 10;

}

stopwatch::stopwatch(const stopwatch& ob) {

this->additionalsec = ob.additionalsec;

this->seccreate = ob.seccreate;

this->secondstart = ob.secondstart;

this->secondstop = ob.secondstop;

strcpy\_s(this->name, ob.name);

cout << "Секундомер скопирован." << endl;

}

void stopwatch::start() {

cout << "Запустился секундомер " << name << ". Отсчет идет с " << additionalsec << " секунды." << endl;

secondstart = time(NULL);

}

void stopwatch::stop() {

secondstop = time(NULL);

cout << "Время работы секундомера " << name << " : " << secondstop - secondstart + additionalsec << endl;

}

void stopwatch::show() {

cout << "Прошло секунд : " << time(NULL) - secondstart << endl;

}

stopwatch::~stopwatch() {

cout << "Отработал деструктор. Время жизни объекта " << name << " : " << time(NULL) - seccreate << endl;

cin.get();

}

int main() {

setlocale(LC\_ALL, "rus");

stopwatch a;

a.start();

Sleep(3000);

a.show();

Sleep(2000);

a.stop();

char nname[10];

cout << "Введите имя секундомера : "; cin >> nname;

stopwatch b(0, nname);

b.start();

Sleep(3000);

b.show();

Sleep(3000);

b.stop();

stopwatch c(b);

c.start();

Sleep(2000);

c.show();

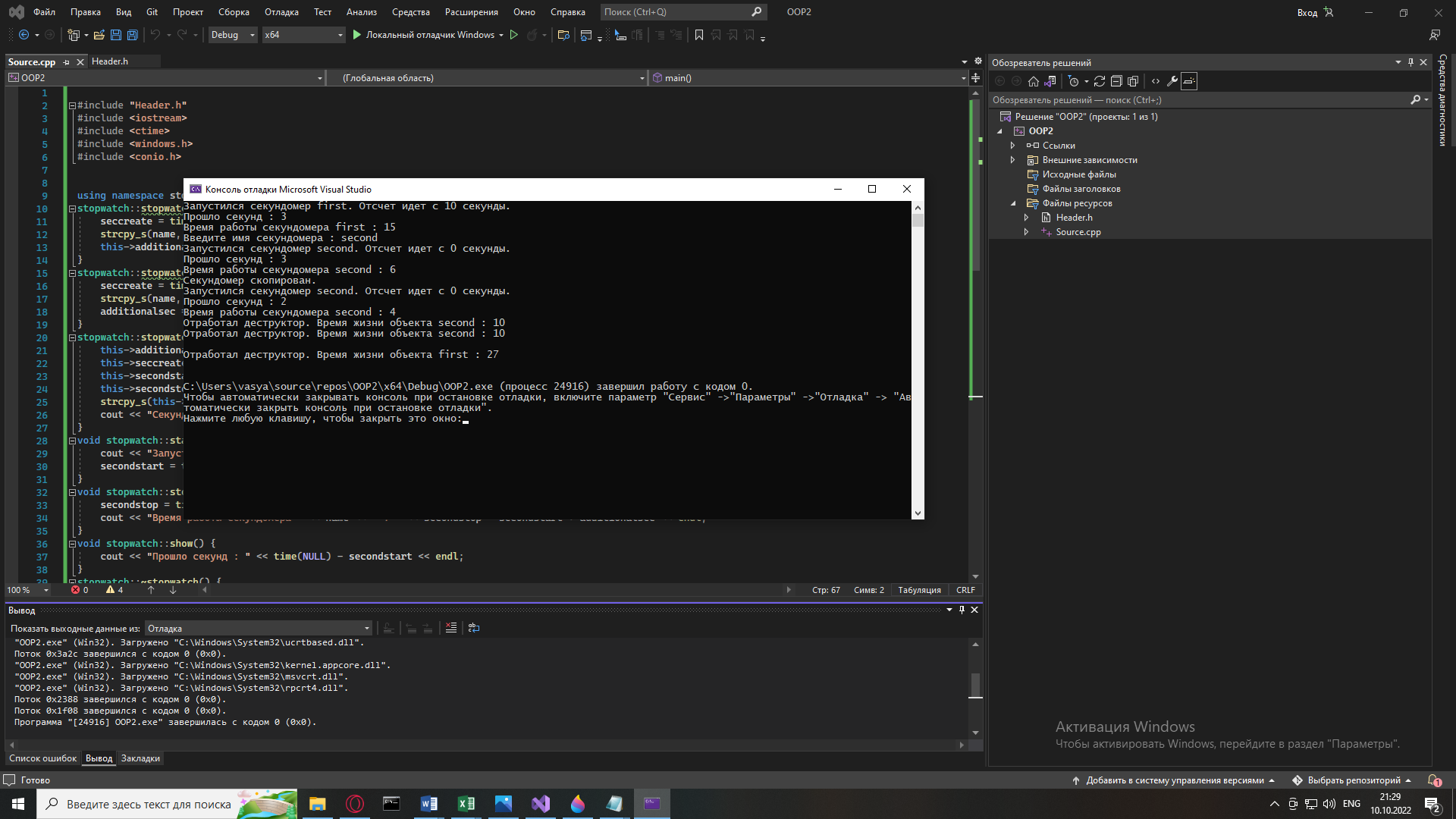
Sleep(2000);

c.stop();

return 0;

}

1. Результат работы



1. Вывод

Я изучил принципы создания классов с конструкторами и применил на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования, а также закрепил знания по теме: Классы, конструкторы, деструкторы.