ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

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

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Иванов И.В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы**

Изучить принципы создания классов с конструкторами, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования.

**Вариант** 1. Создайте 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.

**Ход выполнения**

**Листинг**

#include <iostream>

#include <ctime>

#include "windows.h"

#include <ctype.h>

#include <conio.h>

using namespace std;

class StopWatch {

int sec;

bool start;

clock\_t start\_time;

int value;

public:

StopWatch();

StopWatch(int sec, bool start, clock\_t start\_time, int value);

StopWatch(const StopWatch& other);

~StopWatch();

void Start();

void Stop();

void Show();

};

StopWatch::StopWatch() {

sec = 0;

start = false;

value = 0;

start\_time = false;

}

StopWatch::StopWatch(int sec, bool start, clock\_t start\_time, int value) {

this-> sec = sec;

this-> start = start;

this->value = value;

this->start\_time = start\_time;

}

StopWatch::StopWatch(const StopWatch& other) {//конструктор копирования

sec = other.sec;

start = other.start;

value = other.value;

start\_time = other.start\_time;

}

StopWatch::~StopWatch() {// деструктор

cout << "Время выполнения программы: " << (clock() - start\_time) / CLOCKS\_PER\_SEC << " сек" << endl;

}

void StopWatch::Start() {

if (start) {

return;

}

start\_time = clock();

start = true;

while (!\_kbhit()) {// проверка на нажатие клавиши для остановки

Sleep(1000);// задержка выполнения программы на 1 сек

sec++;

cout << sec << endl;

}

}

void StopWatch:: Show() {

if (start) {

cout << "Секунд прошло: " << sec << endl;

}

}

void StopWatch::Stop() {

if (!start) {

return;

}

value += (clock() - start\_time) / CLOCKS\_PER\_SEC;

start = false;

}

int main() {

setlocale(LC\_ALL, "ru");

/\* cout << "1.по умолчанию" << endl<< "2.копирование"<<endl<< "3.с параметром" << endl;

cin >> i;\*/

StopWatch countdown;

cout << "Вызов конструктора по умолчанию" << endl;

cout << "Старт: " << endl;

countdown.Start();

countdown.Stop();

cout << "Стоп: " << endl;

countdown.Show();

int i;

cin >> i;

if (i == 0) {

StopWatch countdown2(countdown);

cout << "Вызов конструктора копирования" << endl;

cout << "Старт: " << endl;

countdown2.Start();

countdown2.Stop();

cout << "Стоп: " << endl;

countdown2.Show();

}

cin >> i;

if (i==0) {

StopWatch countdown3(5, false, 0, false);

cout << "Вызов конструктора c параметром" << endl;

cout << "Старт: " << endl;

countdown3.Start();

countdown3.Stop();

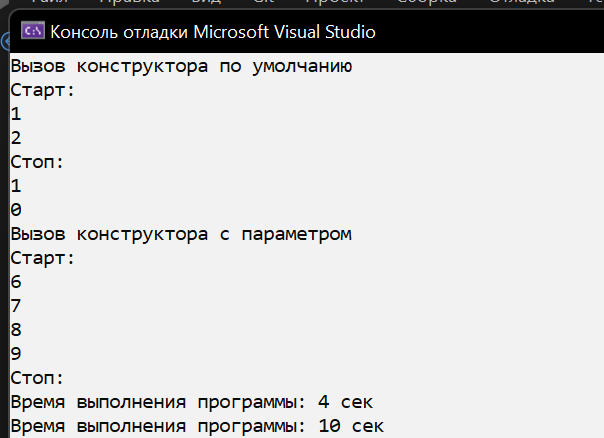
cout << "Стоп: " << endl;

countdown3.Show();

}

return 0;

}**Результат работы программы при исходных данных**

****

**Вывод**

Изучил принципы создания классов с конструкторами, применив на практике знания базовых синтаксических конструкций языка C++ и объектно-ориентированного программирования.