

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

Классы, конструкторы, деструкторы

по курсу: Объектно-ориентированное программирование

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2022

Условие

6 вариант (16 mod 10)

Определите класс Student (Студент). Этот класс должен иметь следующие поля: name (имя), surname (фамилия), year (год поступления в вуз). Класс должен иметь

метод getFullName(), с помощью которого можно вывести одновременно имя и фамилию

студента. Также класс должен иметь метод getCourse(), который будет выводить текущий

курс студента (от 1 до 5). Курс вычисляется так: нужно от текущего года отнять год

поступления в вуз. Текущий год получите самостоятельно.

Листинг программы

main.cpp

```
#include <iostream>
using namespace std;

#include <string.h>

#include "student.h"

int main() {
    // смена кодировки
    setlocale(LC_CTYPE, "Russian");

    Student student_default;
    cout << "По умолчанию" << endl;
    student_default.getFullName();
    student_default.getCourse();

    Student student_args("Дмитрий", "Сутак", 2018);
    cout << "С аргументами" << endl;
    student_args.getFullName();
    student_args.getCourse();

    Student student_copy = student_args;
    cout << "Копирование" << endl;
    student_copy.getFullName();
    student_copy.getCourse();

    return 0;
}
```

student.h

```
#include <iostream>
using namespace std;

#include <time.h>
#include <string.h>

class Student {
public:
    Student(); // конструктор по умолчанию
    Student(string Name, string Surname, int Year); // конструктор с аргументами
    Student(const Student & ref_Student); // конструктор копирования
    ~Student();

    void getFullName();
    void getCourse();

private:
    string name;
    string surname;
    int year;
};

// конструктор по умолчанию
Student::Student() {
    name = "Иван";
    surname = "Иванов";
    year = 2007;
}

// конструктор с аргументами
Student::Student(string Name, string Surname, int Year) {
    name = Name;
    surname = Surname;
    year = Year;
}

// конструктор копирования
Student::Student(const Student & ref_Student) {
    name = ref_Student.name;
    surname = ref_Student.surname;
    year = ref_Student.year;
    cout << "Класс скопирован." << endl;
```

```

}

// деструктор
Student::~Student() {
    cout << "Стработал деструктор" << endl;
}

void Student::getFullName() {
    cout << "ФИО студента: " << name << " " << surname << endl;
}

// вывод времени
void Student::getCourse() {
    const time_t timer = time(NULL);
    struct tm *saved_time = localtime(&timer);

    cout << "Текущий курс студента: " << (1900 + saved_time->tm_year) - year <<
endl;
}

```

Скриншоты

```

По умолчанию
ФИО студента: Иван Иванов
Текущий курс студента: 15
С аргументами
ФИО студента: Дмитрий Сугак
Текущий курс студента: 4
Класс скопирован.
Копирование
ФИО студента: Дмитрий Сугак
Текущий курс студента: 4
Стработал деструктор
Стработал деструктор
Стработал деструктор
Program ended with exit code: 0

```

Вывод

Я узнал, что такое конструктор и деструктор, а так-же ещё и научился ими пользоваться.