МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа № 2:**

"Классы и объекты. Использование конструкторов."

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Общая часть:

1. Определить пользовательский класс
2. Определить в классе следующие конструкторы: с параметрами, без параметров, копирования
3. Определить в классе деструктор
4. Определить в классе компоненты-функции для просмортра и установки полей данных (селекторы и модификаторы)
5. Написать программу, демонстрирующую вызов всех трех видов конструкторов.

Задание варианта:

Пользовательский класс АБИТУРИЕНТ

ФИО абитуриента – string

Специальность – string

Балл ЕГЭ – int

**Анализ задачи:**

1. Требуется создать класс с именем Abiturient.

2. Определить в классе следующие конструкторы:

* Конструктор по умолчанию (без параметров)
* Конструктор с параметрами (принимает ФИО, специальность и балл ЕГЭ)
* Конструктор копирования (принимает ссылку на другой объект класса Abiturient)

3. Определить в классе деструктор, ктороый вызывается при уничтожении объекта класса.

4. Определить в классе компоненты-функции для просмортра и установки полей данных (селекторы и модификаторы)

* Селекторы используются для получения значений полей данных (ФИО, специальность, балл ЕГЭ).
* Модификаторы используются для установки значений полей данных.

5. Написать программу, демонстрирующую вызов всех трех видов конструкторов.

**Код на С++ с комментариями:**

#include <iostream>

using namespace std;

class Abiturient {

private:

string fio;

string specialnost;

int ball\_ege;

public:

Abiturient() {// Конструктор по умолчанию

fio = "";

specialnost = "";

ball\_ege = 0;

}

Abiturient(string fio, string specialnost, int ball\_ege) {// Конструктор с параметрами

this->fio = fio;

this->specialnost = specialnost;

this->ball\_ege = ball\_ege;

}

Abiturient(const Abiturient& other) {// Конструктор копирования

fio = other.fio;

specialnost = other.specialnost;

ball\_ege = other.ball\_ege;

}

~Abiturient() {// Деструктор

cout << "Деструктор вызван для объекта: " << fio << endl;

}

string getFio() const { return fio; }// Селекторы

string getSpecialnost() const { return specialnost; }

int getBallEge() const { return ball\_ege; }

void setFio(string fio) { this->fio = fio; }// Модификаторы

void setSpecialnost(string specialnost) { this->specialnost = specialnost; }

void setBallEge(int ball\_ege) { this->ball\_ege = ball\_ege; }

};

int main() {

setlocale(LC\_ALL, "ru");

Abiturient a1;// Вызов конструктора по умолчанию

Abiturient a2("Иванов Иван Иванович", "Информатик", 240); // Вызов конструктора с параметрами

Abiturient a3(a2);// Вызов конструктора копирования

cout << "Конструктор по уполчанию: " << endl;

cout << "ФИО: " << a1.getFio() << endl;// Демонстрация работы селекторов

cout << "Специальность: " << a1.getSpecialnost() << endl;

cout << "Балл ЕГЭ: " << a1.getBallEge() << endl;

cout << "Конструктор с параметрами: " << endl;

cout << "ФИО: " << a2.getFio() << endl;// Демонстрация работы селекторов

cout << "Специальность: " << a2.getSpecialnost() << endl;

cout << "Балл ЕГЭ: " << a2.getBallEge() << endl;

cout << "Конструктор копирования: " << endl;

cout << "ФИО: " << a3.getFio() << endl;// Демонстрация работы селекторов

cout << "Специальность: " << a3.getSpecialnost() << endl;

cout << "Балл ЕГЭ: " << a3.getBallEge() << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

a1.setFio("Петров Петр Петрович");// Демонстрация работы модификаторов

a1.setSpecialnost("Математик");

a1.setBallEge(190);

a2.setFio("Кириллов Кирилл Кирилович");// Демонстрация работы модификаторов

a2.setSpecialnost("Биолог");

a2.setBallEge(225);

a3.setFio("Антонов Антон Антонович");// Демонстрация работы модификаторов

a3.setSpecialnost("Физик");

a3.setBallEge(183);

cout << "Конструктор по уполчанию после демонстрации работы модификаторов: " << endl;

cout << "ФИО: " << a1.getFio() << endl;// Демонстрация работы селекторов после модификации

cout << "Специальность: " << a1.getSpecialnost() << endl;

cout << "Балл ЕГЭ: " << a1.getBallEge() << endl;

cout << "Конструктор с параметрами после демонстрации работы модификаторов: " << endl;

cout << "ФИО: " << a2.getFio() << endl;// Демонстрация работы селекторов после модификации

cout << "Специальность: " << a2.getSpecialnost() << endl;

cout << "Балл ЕГЭ: " << a2.getBallEge() << endl;

cout << "Конструктор копирования после демонстрации работы модификаторов: " << endl;

cout << "ФИО: " << a3.getFio() << endl;// Демонстрация работы селекторов после модификации

cout << "Специальность: " << a3.getSpecialnost() << endl;

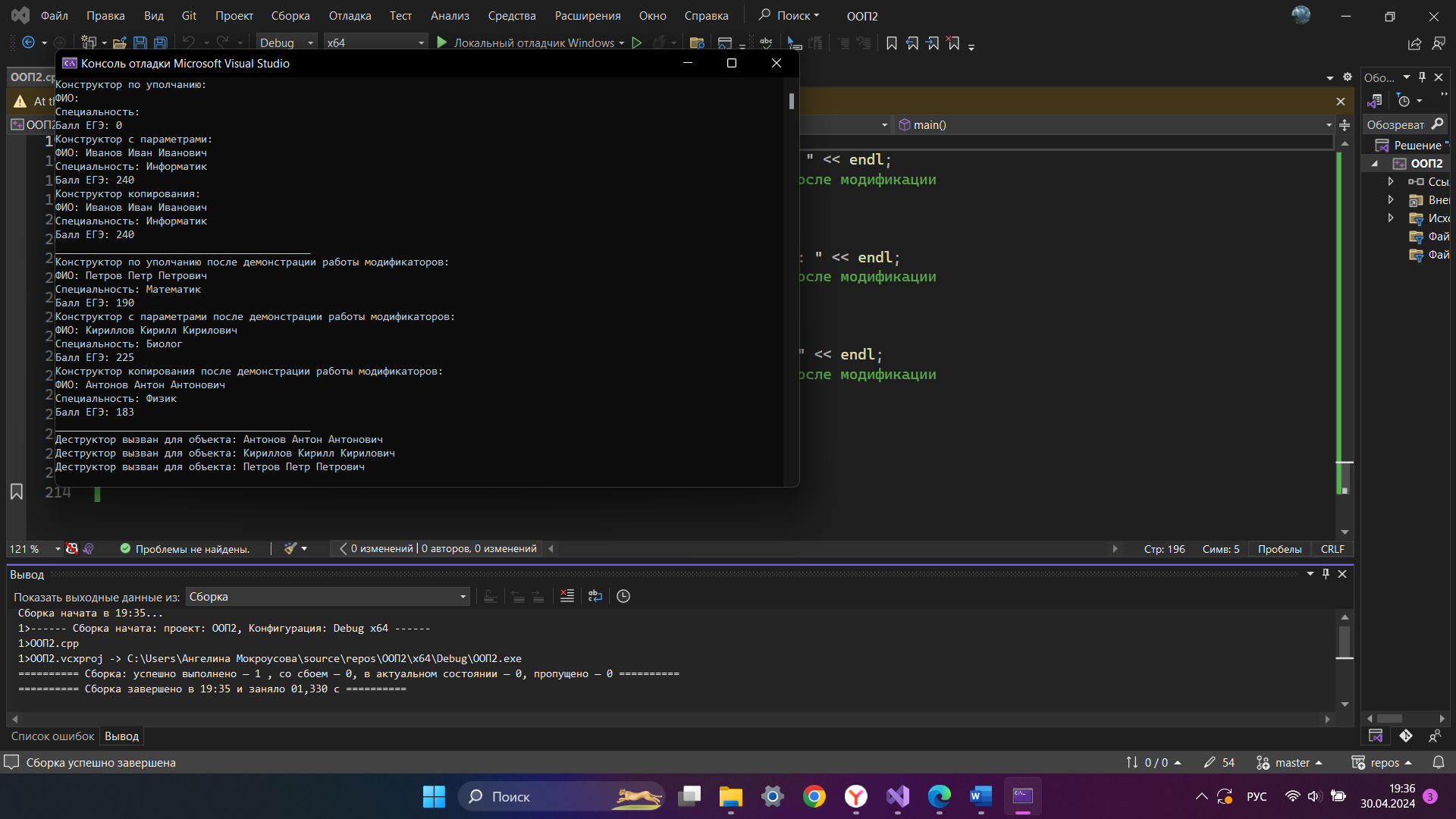
cout << "Балл ЕГЭ: " << a3.getBallEge() << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

return 0;

}

**Скриншот работы программы:**



**Вывод:**

Мною была аписана программа, которая успешно выполняет поставленную перед ней задачу. В процессе написания был повторен и закреплен на практике пройденный ранее материал.

**Ответы на вопросы:**

1. Для чего нужен конструктор?

Конструктор нужен для определения всех полей объекта класса сразу же по его объявлении

1. Сколько типов конструкторов существует в C++?

В C++ существует 3 типа конструкторов: без параметров, с параметрами и копирования;

1. Для чего используется деструктор? В каких случаях деструктор описывается явно?

Деструктор используется для выполнения некоторых действий перед тем, как объект класса будет удален из памяти компьютера. Деструктор следует описывать явно в тех случаях, когда одно или несколько полей класса инициализируются динамически, чтобы избежать утечки памяти;

1. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?

Конструктор без параметров используется для создания “пустого” объекта. Конструктор с параметрами используется для инициализации полей класса заданными значениями. Конструктор копирования используется для копирования значений полей одного объекта класса в другой;

1. В каких случаях вызывается конструктор копирования?

Он вызвается в тех случаях, когда новый объект инициализируется путем копирования полей другого;

1. Перечислить свойства конструкторов.

Конструкторы должны иметь публичный доступ, не передаются через наследование, не возвразают значение (всегда типа void), не могут быть виртуальными, статическими или константными, любой класс по умолчанию имеют конструкторы без параметров и копирования.

1. Перечислить свойства деструкторов.

Деструкторы должны иметь публичный доступ, в каждом классе автоматически объявляется деструктор, если программист не сделал свой, не возвращают значения (всегда тип void), не принимают параметров, не могут быть статическими или константными, но могут быть виртуальтными, не наследуются.

1. К каким атрибутам имеют доступ методы класса?

Методы класса имеют доступ ко всем атрибутам класса;

1. Что представляет собой указатель this?

Указатель this представляет из себя указатель на объект класса, который может использоваться в методах самого класса;

1. Какая разница между методами определенными внутри класса и вне класса?

Разницы нет, кроме синтаксиса определения;

1. Какое значение возвращает конструктор?

Конструкторы не возвращают значений (то есть void);

1. Какие методы создаются по умолчанию?

Конструкторы без параметров и копирования, а также деструктор;

1. Какое значение возвращает деструктор?

Деструкторы не возвращаеют значений (то есть void);

1. В описании класса отсутствует конструктор без параметров;
2. Будет вызван конструктор без параметров;
3. Будет вызван конструктор с параметрами;
4. Будут вызваны конструктор с параметрами для s1 и конструктор копирования для s2;
5. Будут вызваны конструкторы с параметрами для s1 и без параметров для s2;
6. Будет использоваться конструктор коопирования;
7. Каким образом можно присвоить новое значение атрибуту name объекта р? p.set\_name(<аргументы>).

**Скриншот с GitHab:**

