Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 - «Информатика и вычислительная техника»

**О Т Ч Е Т**

**по лабораторной работе “Классы” №2**

**по дисциплине**

**«Основы алгоритмизации и программирования» семестр 2**

Выполнил студент гр. ИВТ-21-1б

Ипатов Дмитрий Сергеевич

Проверил:

Ст. Преподаватель кафедры ИТАС

Яруллин Д.В.

(оценка) (подпись)

г. Пермь-2022

**Постановка задачи:**

1. Определить пользовательский класс.
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
3. Определить в классе деструктор.
4. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).
5. Написать демонстрационную программу, в которой продемонстрировать все три случая вызова конструктора-копирования, вызов конструктора с параметрами и конструктора без параметров.

**Анализ задачи:**

В каждом классе должен быть хотя бы один метод, который предназначен для инициализации объекта. Его имя совпадает с именем класса, и он вызывается автоматически при инициализации объекта. Автоматический вызов конструктора позволяет избежать ошибок, связанных с использованием неинициализированных переменных.

Существует три типа конструкторов:

* конструктор с параметрами, используется для инициализации объекта требуемыми значениями;
* конструктор без параметров, используется для создания «пустого» объекта;
* конструктор копирования, используется для создания объекта, аналогичного тому, который уже существует.

Конструктор без параметров и конструктор копирования создаются по умолчанию.

Деструктор – это особый вид метода, применяющийся для освобождения ресурсов, выделенных конструктором объекту. Деструктор вызывается автоматически, когда объект удаляется из памяти:

* + - для локальных объектов это происходит при выходе из блока, в котором они объявлены;
    - для глобальных *—* как часть процедуры выхода из main;
    - для объектов, заданных через указатели*,* деструктор вызывается неявно при использовании операции delete.

**Код программы:**

#include <iostream>

using namespace std;

class entrant

{

private:

string FIO;

string profession;

int scores;

public:

entrant(string FIO, string profession, int scores) //конструктор с параметрами

{

this->FIO = FIO;

this->profession = profession;

this->scores = scores;

}

entrant() // конструктор без параметров

{

this->FIO = "-";

this->profession = "-";

this->scores = 0;

}

entrant(const entrant& ent) //конструктор копирования

{

this->FIO = ent.FIO;

this->profession = ent.profession;

this->scores = ent.scores;

}

void SetFIO(string FIO)

{

this->FIO = FIO;

}

void SetProfession(string profession)

{

this->profession = profession;

}

void SetScores(int scores)

{

this->scores = scores;

}

void Print()

{

cout << "ФИО: " << FIO << endl;

cout << "Профессия: " << profession << endl;

cout << "Баллы ЕГЭ: " << scores << endl << endl;

}

~entrant()

{

cout << "Деструктор" << endl;

}

};

int main()

{

setlocale(LC\_ALL, "rus");

entrant student1;

student1.Print();

entrant student2("Ипатов Дмитрий Сергеевич", "Программист", 238);

student2.Print();

entrant student3 = student2;

student3.Print();

student3.SetFIO("Титов Никита Викторович");

student3.SetProfession("Репер");

student3.SetScores(203);

student3.Print();

return 0;

}

**Скриншоты результатов:**

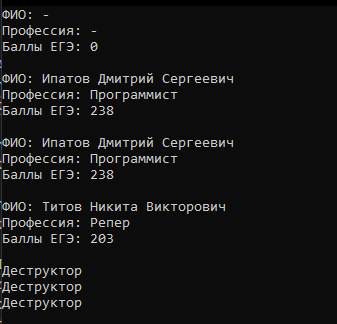


Рис. 1 Результаты программы

**Контрольные вопросы:**

1. Для чего нужен конструктор?

Для создания экземпляров класса.

1. Сколько типов конструкторов существует в C++?

три: копирования, с параметрами и без параметров.

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

Для удаления объектов. Если нам нужны какие-то действия при удалении объекта (например вывод сообщения) или если есть поля с динамически выделяемой памятью.

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

Конструктор без параметров: для создания объекта без каких либо исходных данных.

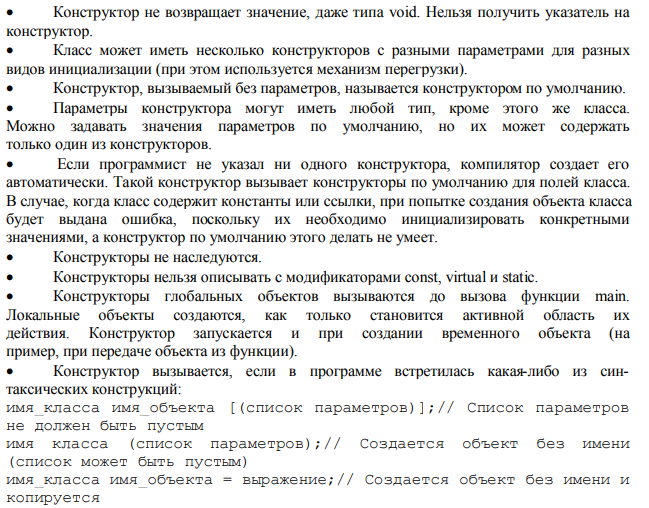
Конструктор с параметрами: для инициализации полей экземпляра переданными параметрами.

Конструктор копирования: для создания копии уже существующего объекта.

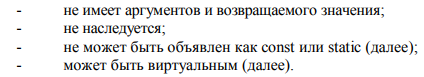
1. В каких случаях вызывается конструктор копирования?

Когда в программе необходимо создать копию уже существующего объекта, когда мы передаём объект класса в качестве параметра для функции или когда возвращаем объект класса из функции.

1. Перечислить свойства конструкторов



1. Перечислить свойства деструкторов



1. К каким атрибутам имеют доступ методы класса?

К любым.

1. Что представляет собой указатель this?

Указатель на объект вызывающий метод.

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

Ничем, кроме того, что метод вне класса должен иметь свой прототип внутри класса и определяться вот так:https://lh5.googleusercontent.com/EdGRHdeVOloCxACWLdED479FQheO2e9ONcnSFmGn3gLymuEeyEYn7NUWX4e4OmEr13CfbVhLMTlejpUl4doC-GWnxV90LrrRoURfbdDQU5h0pv8nw2FJuOkc1pxWlEGMj7Lrwu6SKpoUhtpv2rg

1. Какое значение возвращает конструктор?

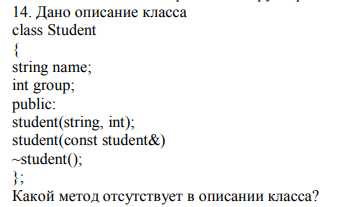
Никакое, даже void.

1. Какие методы создаются по умолчанию?

Конструктор без параметров, конструктор копирования и деструктор.

1. Какое значение возвращает деструктор?

Никакое, даже void.



Конструктор копирования ( в коде нет “;” значит это не прототип, но при этом нет логики этого метода), не хватает так же конструктора без параметров (его описание желательно, но не обязательно), так же не хватает как минимум селекторов, так как поля по умолчанию приватные и к ним никак не получить доступ.

https://lh3.googleusercontent.com/HtNjKSY1tmksp_AM2f2sOLHJS8k-6rRTlECUR9Iles8QrlUhnXYwFXKlZOOCuopARl9lnni-SZt1k-4xqRFL9d6BqkOvQru7ZH3obYiFbeA9CP66j5mJcZ2vQlv-gj60b2sJTkZSFC7GELFdn90

Конструктор без параметров.