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: 09.03.04 – «Программная инженерия»

**Лабораторная работа 2.**

**" Классы и объекты. Использование конструкторов "**

Выполнила студентка гр. РИС-24-3б

Черепанова Диана Романовна

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

(оценка) (подпись)

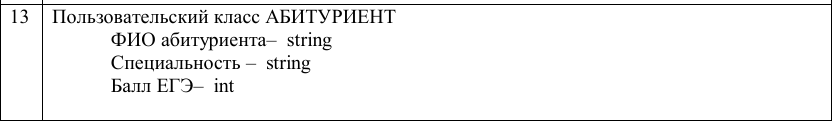
(дата)

г. Пермь, 2025

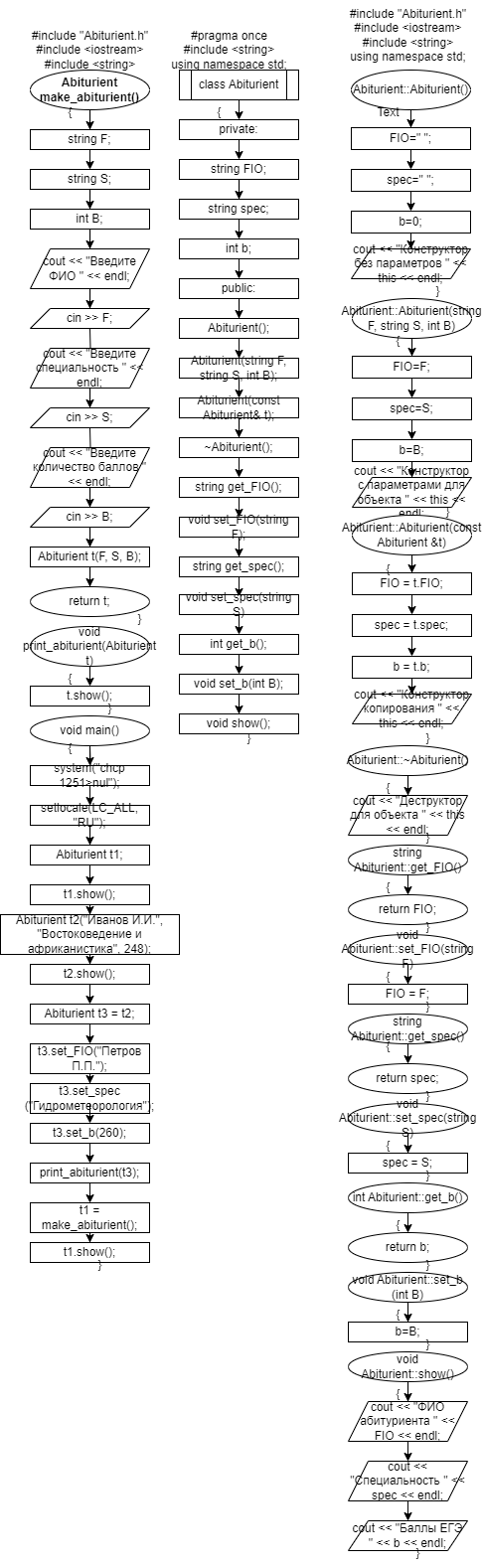
**Постановка задачи**

Определить пользовательский класс

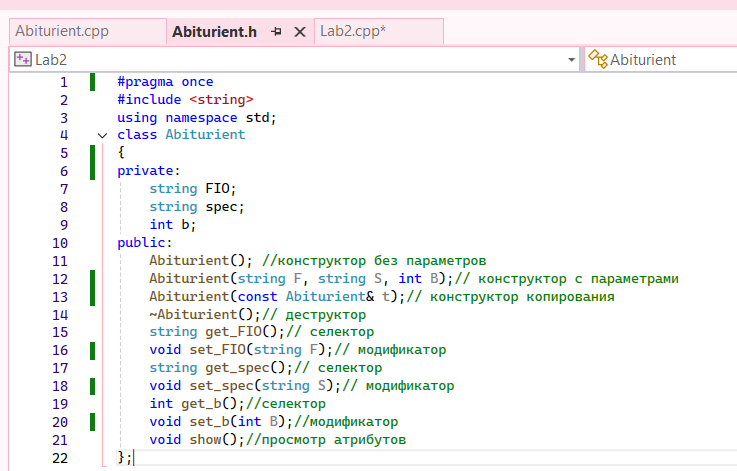
1. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.
2. Определить в классе деструктор.
3. Определить в классе компоненты-функции для просмотра и установки полей данных (селекторы и модификаторы).
4. Написать демонстрационную программу, в которой продемонстрировать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



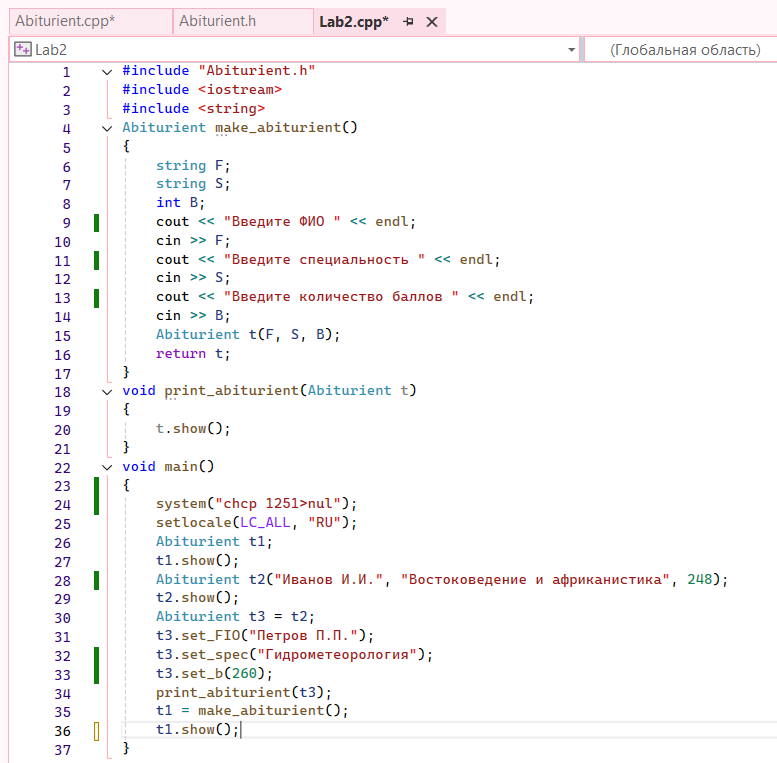
Блок-схема



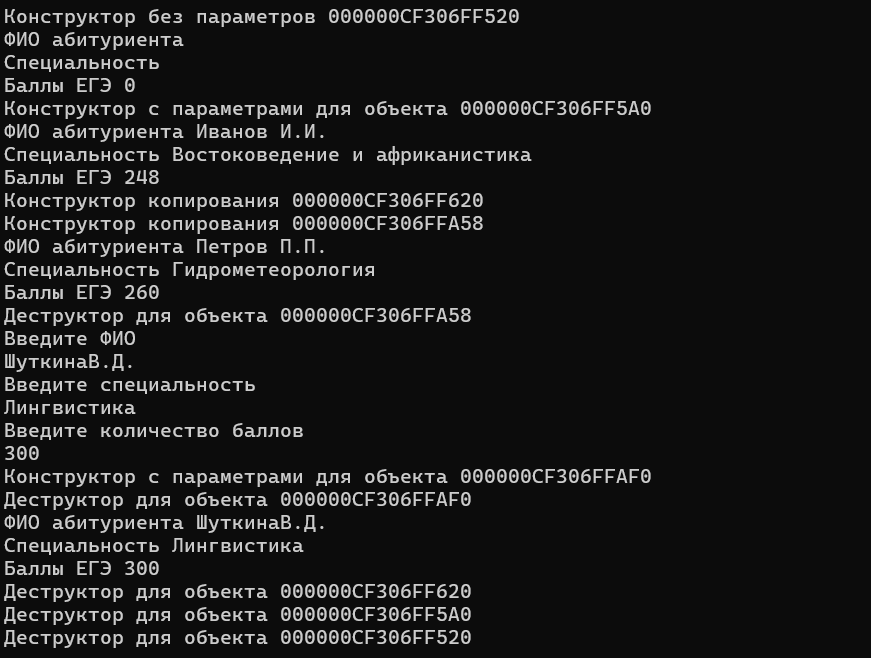
Код







Результат программы



Ответы на вопросы

1. Для чего нужен конструктор?

Конструктор – один из методов инициализации объекта созданного класса. Конструктор имеет одинаковое имя с классом.

1. Сколько типов конструкторов существует в С++?

Существует три типа конструкторов:

1. Конструктор без параметров, для создания “пустого” объекта
2. Конструктор с параметрами, для создания объекта с заданными значениями
3. Конструктор копирования, для создания объекта с атрибутами ранее созданного.
4. Для чего используется деструктор? В каких случаях деструктор описывается явно?

Деструктор вызывается, когда нужно удалить объект из памяти. Описывать деструктор явно необходимо в случае, когда объект содержит указатели на память, выделяемую автоматически (иначе поля-указатели не буду свободными).

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 объекта требуемыми значениями.

Конструктор копирования используется для создания нового объекта путём копирования существующего.

1. В каких случаях вызывается конструктор копирования?

Конструктор копирования вызывается:

* При описании нового объекта с инициализацией другим объектом
* При передаче объекта в функцию по значению
* При возврате объекта из функции

1. Перечислить свойства конструкторов.

* Не возвращает значение, нельзя получить указатель на конструктор
* В классе может быть несколько конструкторов (нужна перегрузка)
* Тип аргументов в конструкторе любой, кроме класса
* Конструктор без параметров – конструктор по умолчанию
* Не указан какой-либо конструктор – компилятор создает конструктор по умолчанию автоматически
* Не наследуются
* Нельзя описывать с модификаторами const, virtual, static

1. Перечислить свойства деструкторов.

* Нет аргументов и не возвращает значение
* Не наследуется
* Нельзя описывать с модификаторами const, static
* Можно описывать с модификатором virtual

1. К каким атрибутам имеют доступ методы класса?

Методы класса имеют доступ ко всем элементам класса, независимо от спецификаторов доступа и порядка объявления.

1. Что представляет собой указатель this?

Указатель this – указатель на объект, используемый в каком-либо из методов класса. Используется в качестве возвращающегося значения.

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

Разница между методами, определенными внутри класса и вне класса состоит лишь в их описании.

1. Какое значение возвращает конструктор?

Любой тип конструктора не возвращает значения.

1. Какие методы создаются по умолчанию?

По умолчанию создаются конструкторы и деструктор.

1. Какое значение возвращает деструктор?

Деструктор не возвращает значение.

1. Дано описание класса

class Student

{

string name;

int group;

public:

Student(string, int);

Student(const Student&);

~Student();

};

Какой метод отсутствует в описании класса?

В данном случае отсутствует конструктор без параметров.

1. Какой метод будет вызван при выполнении следующих операторов:

Student\* s;

s = new Student;

В данном случае будет вызван конструктор без параметров.

1. Какой метод будет вызван при выполнении следующих операторов:

Student s(“Ivanov”, 20);

В данном случае будет вызван конструктор с параметрами.

1. Какие методы будут вызваны при выполнении следующих операторов:

Student s1(“Ivanov”, 20);

Student s2 = s1;

В данном случае будут вызваны конструктор с параметрами и конструктор копирования соответственно.

1. Какие методы будут вызваны при выполнении следующих операторов:

Student s1(“Ivanov”, 20);

Student s2;

s2 = s1;

В данном случае будут вызваны конструктор с параметрами, конструктор без параметров и конструктор копирования соответственно.

1. Какой конструктор будет использоваться при передаче параметра в функцию print():

void print(student a)

{

a.show();

}

В данном случае будет вызван конструктор копирования.

1. Класс описан следующим образом:

class Student

{

string name;

int age;

public:

void set\_name(string);

void set\_age(int);

…….

};

Student p;

Каким образом можно присвоить новое значение атрибуту name объекта p?

Присвоение нового значения атрибуту name объекта p осуществляется путём обращения к модификатору, которое аналогично обращению к полям структуры:

String row = “Ivanov”; // строка, хранит данные для атрибута name

p.set\_name(string row);

GitHub: https://github.com/d1frd