Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе 18.2**

Тема: Классы и объекты. Использование конструкторов.

Выполнил работу

студент группы РИС-20-1б

Томилов В.А.

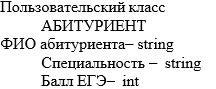
Проверила

Доцент кафедры ИТАС

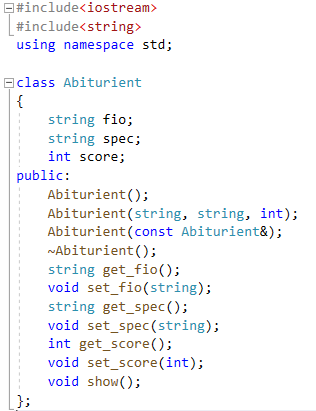
к.т.н ПоляковаО.А.

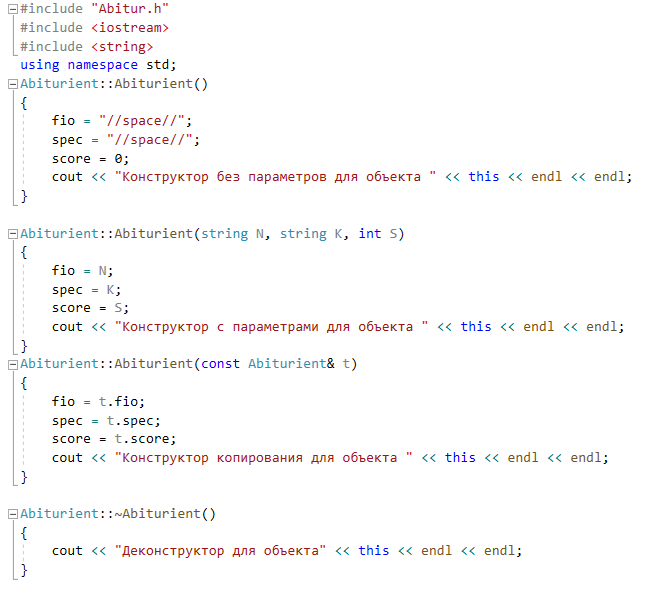
Пермь, 2021

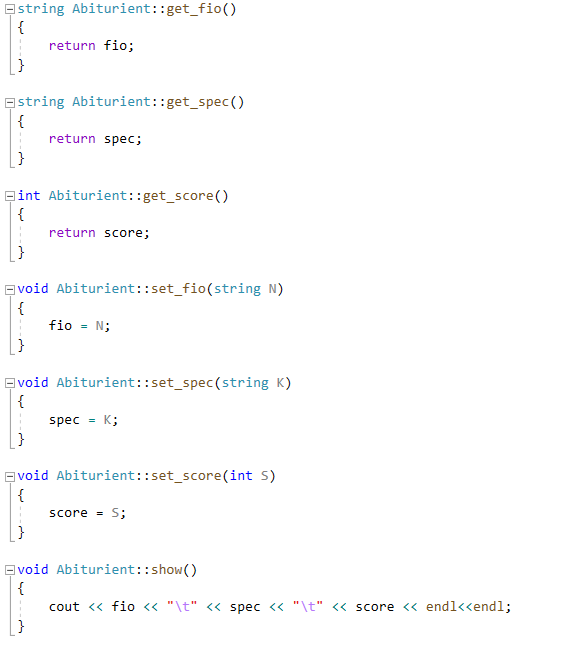
Постановка задачи

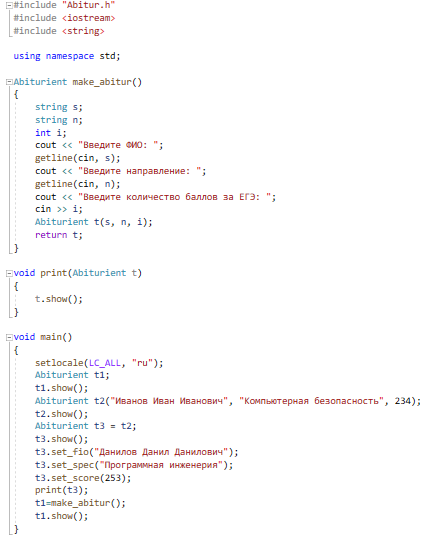


Анализ задачи

* 1. Для решения задачи необходимо
  2. Добавить в проект файл Abiturient.h, содержащий описание класса.
  3. Добавим в проект файл Abiturient.cpp, содержащий описание методов класса Abiturient.



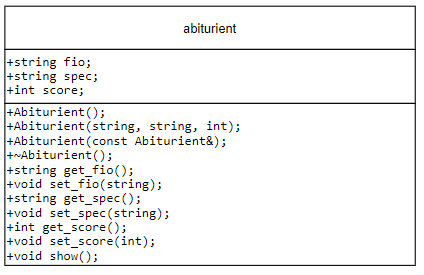


* 1. Добавить в проект файл Lab2\_main.cpp, содержащий основную программу.
  2. В ходе работы были использованы следующие типы данных:

2.1 (int) score – балл ЕГЭ

2.2 (string) fio, spec – ФИО абитуриента и Специальность

UML – диаграмма



Код программы

main.cpp

#include "Abitur.h"

#include <iostream>

#include <string>

using namespace std;

Abiturient make\_abitur()

{

string s;

string n;

int i;

cout << "Введите ФИО: ";

getline(cin, s);

cout << "Введите направление: ";

getline(cin, n);

cout << "Введите количество баллов за ЕГЭ: ";

cin >> i;

Abiturient t(s, n, i);

return t;

}

void print(Abiturient t)

{

t.show();

}

void main()

{

setlocale(LC\_ALL, "ru");

Abiturient t1;

t1.show();

Abiturient t2("Иванов Иван Иванович", "Компьютерная безопасность", 234);

t2.show();

Abiturient t3 = t2;

t3.show();

t3.set\_fio("Данилов Данил Данилович");

t3.set\_spec("Программная инженерия");

t3.set\_score(253);

print(t3);

t1=make\_abitur();

t1.show();

}

Abiturient.h

#pragma once

#include<iostream>

#include<string>

using namespace std;

class Abiturient

{

string fio;

string spec;

int score;

public:

Abiturient();

Abiturient(string, string, int);

Abiturient(const Abiturient&);

~Abiturient();

string get\_fio();

void set\_fio(string);

string get\_spec();

void set\_spec(string);

int get\_score();

void set\_score(int);

void show();

};

Abiturient.cpp

#include "Abitur.h"

#include <iostream>

#include <string>

using namespace std;

Abiturient::Abiturient()

{

fio = "//space//";

spec = "//space//";

score = 0;

cout << "Конструктор без параметров для объекта " << this << endl << endl;

}

Abiturient::Abiturient(string N, string K, int S)

{

fio = N;

spec = K;

score = S;

cout << "Конструктор с параметрами для объекта " << this << endl << endl;

}

Abiturient::Abiturient(const Abiturient& t)

{

fio = t.fio;

spec = t.spec;

score = t.score;

cout << "Конструктор копирования для объекта " << this << endl << endl;

}

Abiturient::~Abiturient()

{

cout << "Деконструктор для объекта" << this << endl << endl;

}

string Abiturient::get\_fio()

{

return fio;

}

string Abiturient::get\_spec()

{

return spec;

}

int Abiturient::get\_score()

{

return score;

}

void Abiturient::set\_fio(string N)

{

fio = N;

}

void Abiturient::set\_spec(string K)

{

spec = K;

}

void Abiturient::set\_score(int S)

{

score = S;

}

void Abiturient::show()

{

cout << fio << "\t" << spec << "\t" << score << endl<<endl;

}

Ответы на вопросы

1. Конструкторы предназначены только для выполнения инициализации.
2. 4: конструктор по умолчанию (default constructor), внешний конструктор (external constructor), конструктор копирования (copy constructor), виртуальный конструктор (virtual constructor)
3. Деструктор — это специальный тип метода класса, который выполняется при удалении объекта класса.
4. Конструктор с параметрами, используется для инициализации объекта требуемыми значениями;

Конструктор без параметров используется для создания «пустого» объекта; Конструктор копирования вызывается в тех случаях, когда новый объект создается путем копирования существующего.

1. –
2. Его имя совпадает с именем класса, и он вызывается автоматически при инициализации объекта.

* конструктор не возвращает значение, даже типа void.
* конструктор нельзя описывать с модификаторами const, virtual и static.
* конструкторы глобальных объектов вызываются до вызова функции main

7. Деструктор вызывается автоматически, когда объект удаляется из памяти. Имя деструктора начинается с тильды (~).

* не имеет аргументов и возвращаемого значения;
* не наследуется;
* не может быть объявлен как const или static (далее);
* может быть виртуальным (далее).

1. Методы класса имеют неограниченный доступ ко всем элементам класса
2. Указатель на тот объект, для которого он вызван.
3. Определение метода внутри класса ничем не отличается от определения обычной функции. По умолчанию такой метод считается встроенной функцией (inline). Если метод определяется вне функции, то принадлежность метода классу указывается с помощью имени класса: Имя\_класса::Имя\_метода.
4. Не возвращает.
5. Конструктор, деструктор
6. Не возвращает.
7. Конструктор
8. Конструктор без параметров
9. Конструктор с параметрами
10. Конструктор с параметрами

Конструктор копирования

1. Конструктор с параметрами

Конструктор без параметров

Присваивание

1. Конструктор без параметров
2. p.set\_name(string)