Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №18.2**

Дисциплина: «Информатика»

Тема: Объектно-ориентированное программирование. Использование конструкторов

Вариант 4

Выполнил:

Студент группы РИС-20-1б

Еске Вячеслав Сергеевич

Проверила:

Доцент кафедры ИТАС

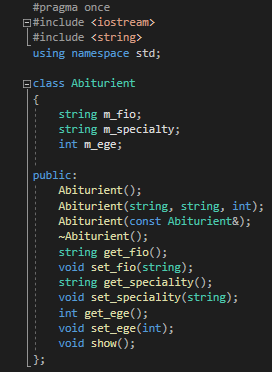
Полякова О. А.

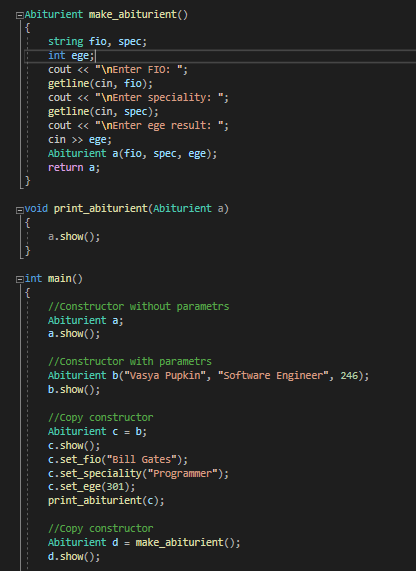
**Постановка задачи**

1. Определить пользовательский класс. Определить в класс следующие конструкторы: без параметров, с параметрами, копирования. Определить в классе деструктор.
2. Пользовательский класс ЧИСЛО   
   Мантисса – foat  
   Порядок – int  
   Строковое представление – string

**Анализ задачи**

1. Для решения задачи необходимо…
   1. Организовать класс Abiturient, который хранит имя, специальность и результат за экзамены.
   2. Организовать методы для ввода данных, гетторы, сетторы, конструкторы и деструктор.



* 1. В функции main произвести тестирование. 

1. В программе были использованы следующие типы данных:
   1. Тип данных int для хранения результата за экзамен;



* 1. Тип данных string для хранения ФИО и специальности;



**Код**

Main.cpp

#include "Number.h"

#include <iostream>

#include <string>

using namespace std;

Number make\_number()

{

float m;

int p;

string s;

cout << "Введите мантиссу: ";

cin >> m;

cout << "Введите порядок: ";

cin >> p;

cout << "Введите строку: ";

cin >> s;

Number n(m, p, s);

return n;

}

void print\_number(Number n)

{

n.show();

}

int main()

{

setlocale(LC\_ALL, "Rus");

Number n1;

n1.show();

Number n2(0.118625, 3, "118.625");

n2.show();

Number n3 = n2;

n3.set\_mantissa(0.523674);

n3.set\_poryadok(1);

n3.set\_stroka("5.23674");

print\_number(n3);

n1=make\_number();

n1.show();

return 0;

system("pause");

}

Number.h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Number

{

float mantissa;

int poryadok;

string stroka;

public:

Number();

Number(float, int, string);

Number(const Number&);

~Number();

float get\_mantissa();

void set\_mantissa(float);

int get\_poryadok();

void set\_poryadok(int);

string get\_stroka();

void set\_stroka(string);

void show();

};

Number.cpp

#include "Number.h"

#include <iostream>

#include <string>

using namespace std;

Number::Number()

{

mantissa = 0;

poryadok = 0;

stroka = "";

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

}

Number::Number(float M, int P, string S)

{

mantissa = M;

poryadok = P;

stroka = S;

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

}

Number::Number(const Number &n)

{

mantissa = n.mantissa;

poryadok = n.poryadok;

stroka = n.stroka;

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

}

Number::~Number()

{

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

}

float Number::get\_mantissa()

{

return mantissa;

}

int Number::get\_poryadok()

{

return poryadok;

}

string Number::get\_stroka()

{

return stroka;

}

void Number::set\_mantissa(float M)

{

mantissa = M;

}

void Number::set\_poryadok(int P)

{

poryadok = P;

}

void Number::set\_stroka(string S)

{

stroka = S;

}

void Number::show()

{

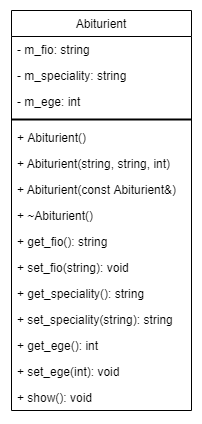
cout << "Мантисса: " << mantissa << endl;

cout << "Порядок: " << poryadok << endl;

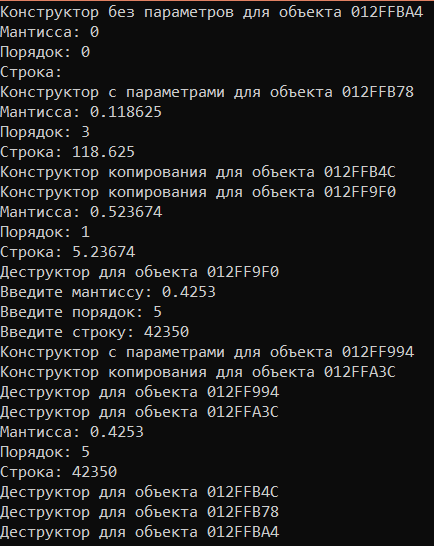
cout << "Строка: " << stroka << endl;

}

**UML диаграмма**



**Скриншоты тестов**

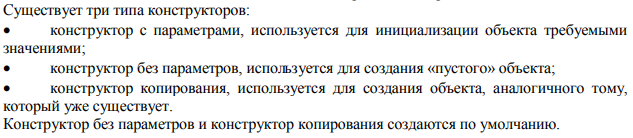


**Контрольные вопросы**



1. Конструктор необходим для инициализации объекта.



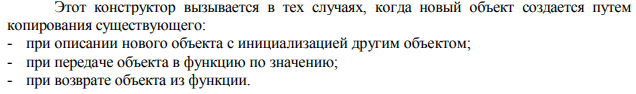




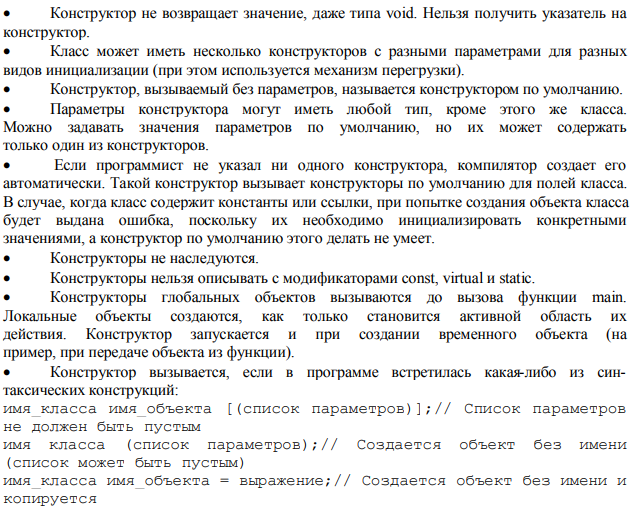
3. Деструктор – это особый вид метода, применяющийся для освобождения ресурсов, выделенных конструктором объекту. Описывать деструктор явным образом требуется в случае, когда объект содержит указатели на память, выделяемую динамически.



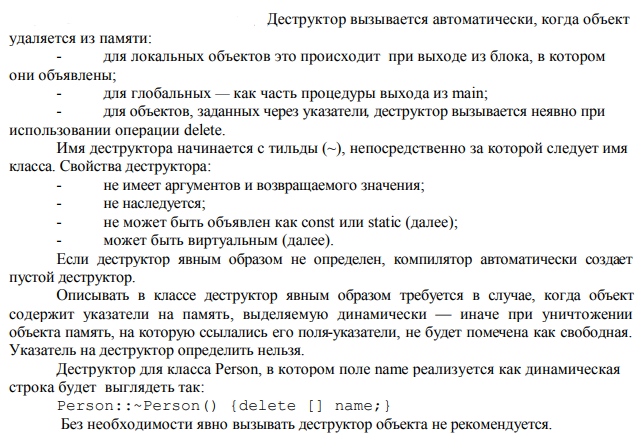
4. Конструктор с параметрами используется для инициализации объекта требуемыми значениями. Конструктор без параметров используется для создания «пустого» объекта. Конструктор копирования – это специальный вид конструктора, получающий в качестве единственного параметра указатель на объект этого же класса.





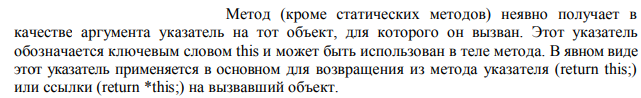


7. 

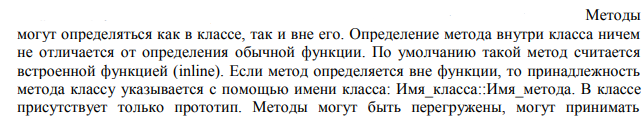


8.



9.



10.



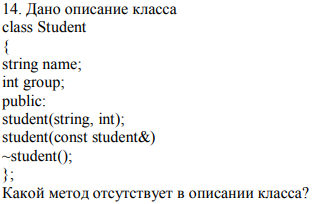
11. Конструктор не возвращает значения.



12. Конструкторы и деструкторы без параметров, если таковые не были прописаны явно.



13. Деструктор не возвращает значения.



14. Отсутствует конструктор без параметров в описании класса.



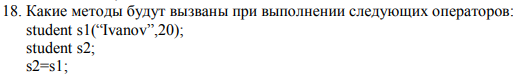
15. Будет вызван конструктор без параметров.



16. Будет вызван конструктор с параметрами.



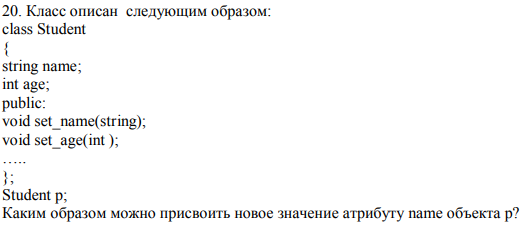
17. Будут вызваны конструктор с параметрами, а затем конструктор копирования.



18. Будут вызваны конструктор с параметрами, затем конструктор без параметров, а затем конструктор копирования.



19. Конструктор без параметров.



20. C помощью метода set\_name().