Министерство образования и науки Российской Федерации

Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.2**

Дисциплина: «Информатика»

Тема: Объектно-ориентированное программирование. Использование конструкторов

Вариант 8

Выполнил:

Студент группы РИС-20-1б

Тараканов Д. М.

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

Пермь

2021 год

# **Цель работы**

1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.
2. Создание объектов с использованием конструкторов.

# **Постановка задачи**

1. Определитель пользовательский класс.
2. Определить в классе следующие конструкторы:
3. Без параметров;
4. С параметрами;
5. Копирования;
6. Определить в классе деструктор.
7. Определить в классе компоненты-функции для просмотра и установки полей.
8. Написать демонстрационную программу, в которой продемонстрировать все три случая вызова конструктора - копирования, вызов конструктора с параметрами и конструктора без параметров.

(8 вариант)

1. Пользовательский класс КВАРТИРА.
2. Адрес - string.
3. Площадь - double.
4. Количество комнат – int.

# **Анализ задачи**

1. Определить какие действия предстоит выполнить:

* Разработать класс flat который хранит в себе адрес, площадь и кол-во комнат в квартире.
* Организовать методы для ввода данных, селекторы и модификатор, конструкторы и деструкторы.

class Flat {

string adress;

double square;

int rooms;

public:

Flat();//Конструкторо без параметров

Flat(string, double, int);//Конструктор с параметрами

Flat(const Flat&);//Конструктор копирования

~Flat();//Деструктор

string get\_adress();//Селектор

void set\_adress(string);//Модификатор

double get\_square();//Селектор

void set\_square(double);//Модификатор

int get\_rooms();//Селектор

void set\_rooms(int);//Модификатор

void show();//Просмотр атрибутов

};

* Разработать функцию make\_flat для создания объекта типа flat.

Flat make\_flat() {

string s;

cout << "\nВведите адресс квартиры: "; getline(cin,s);

double d;

cout << "\nВведите площадь квартиры: "; cin >> d;

int i;

cout << "\nВведи кол-во комнат в квартире: "; cin >> i;

Flat temp(s, d, i);

return temp;

}

* Протестировать работу конструкторов в функции main.

void main() {

system("chcp 1251");

//Конструктор без параметров

cout << "\n1.\n";

Flat apartment1;

apartment1.show();

//Конструктор с параметрами

cout << "\n2.\n";

Flat apartment2 = {"Ленина 48", 20.0, 3};

apartment2.show();

//Конструктор копирования

cout << "\n3.\n";

Flat apartment3 = apartment2;

apartment3.set\_adress("Екатериниская 28а");

apartment3.set\_square(14.23);

apartment3.set\_rooms(1);

print\_flat(apartment3);

//Через функцию make\_flat

cout << "\n4.\n";

apartment1 = make\_flat();

apartment1.show();

//Через селекторы

cout << "\n5.\n";

string s = apartment2.get\_adress();

double d = apartment2.get\_square();

int i = apartment2.get\_rooms();

cout << "\nКвартира\n\nАдресс: " << s << endl;

cout << "\nПлощадь: " << d << endl;

cout << "\nКол-во комнат: " << i << endl;

system("pause");

}

1. C какими типами данных предстояло работать:

* Для хранения адреса квартира реализована переменная address типа string.

string adress;

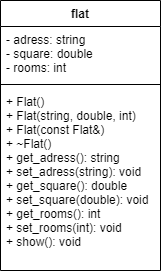
* Для хранения площади квартиры реализована переменная square типа double.

double square;

* Для хранения кол-ва комнат квартиры реализована переменная rooms типа int.

int rooms;

# **UML диаграмма**



# **Код программы на языке C++**

Заголовочный файл flat.h

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Flat {

string adress;

double square;

int rooms;

public:

Flat();//Конструкторо без параметров

Flat(string, double, int);//Конструктор с параметрами

Flat(const Flat&);//Конструктор копирования

~Flat();//Деструктор

string get\_adress();//Селектор

void set\_adress(string);//Модификатор

double get\_square();//Селектор

void set\_square(double);//Модификатор

int get\_rooms();//Селектор

void set\_rooms(int);//Модификатор

void show();//Просмотр атрибутов

};

Обьявление класса в flat.cpp

#include <iostream>

#include <string>

#include "flat.h"

using namespace std;

Flat::Flat() {

adress = " ";

square = 0.0;

rooms = 0;

cout << "\nКонструктор без параметров для обьекта " << this << endl;

}

Flat::Flat(string A, double S, int R) {

adress = A;

square = S;

rooms = R;

cout << "\nКонструктор с параметрами для обьекта " << this << endl;

}

Flat::Flat(const Flat &temp) {

adress = temp.adress;

square = temp.square;

rooms = temp.rooms;

cout << "\nКонструктор копирования для обьекта " << this << endl;

}

Flat::~Flat() {

cout << "\nДеструктор " << this << endl;

}

string Flat::get\_adress() {

return adress;

}

double Flat::get\_square() {

return square;

}

int Flat::get\_rooms() {

return rooms;

}

void Flat::set\_adress(string A) {

adress = A;

}

void Flat::set\_square(double S) {

square = S;

}

void Flat::set\_rooms(int R) {

rooms = R;

}

void Flat::show() {

cout << "\nКвартира\n\nАдресс: " << adress << endl;

cout << "\nПлощадь: " << square << endl;

cout << "\nКол-во комнат: " << rooms << endl;

}

Main файл

#include <iostream>

#include <string>

#include "flat.h"

using namespace std;

Flat make\_flat() {

string s;

cout << "\nВведите адресс квартиры: "; getline(cin,s);

double d;

cout << "\nВведите площадь квартиры: "; cin >> d;

int i;

cout << "\nВведи кол-во комнат в квартире: "; cin >> i;

Flat temp(s, d, i);

return temp;

}

void print\_flat(Flat temp) {

temp.show();

}

void main() {

system("chcp 1251");

//Конструктор без параметров

cout << "\n1.\n";

Flat apartment1;

apartment1.show();

//Конструктор с параметрами

cout << "\n2.\n";

Flat apartment2 = {"Ленина 48", 20.0, 3};

apartment2.show();

//Конструктор копирования

cout << "\n3.\n";

Flat apartment3 = apartment2;

apartment3.set\_adress("Екатериниская 28а");

apartment3.set\_square(14.23);

apartment3.set\_rooms(1);

print\_flat(apartment3);

//Через функцию make\_flat

cout << "\n4.\n";

apartment1 = make\_flat();

apartment1.show();

//Через селекторы

cout << "\n5.\n";

string s = apartment2.get\_adress();

double d = apartment2.get\_square();

int i = apartment2.get\_rooms();

cout << "\nКвартира\n\nАдресс: " << s << endl;

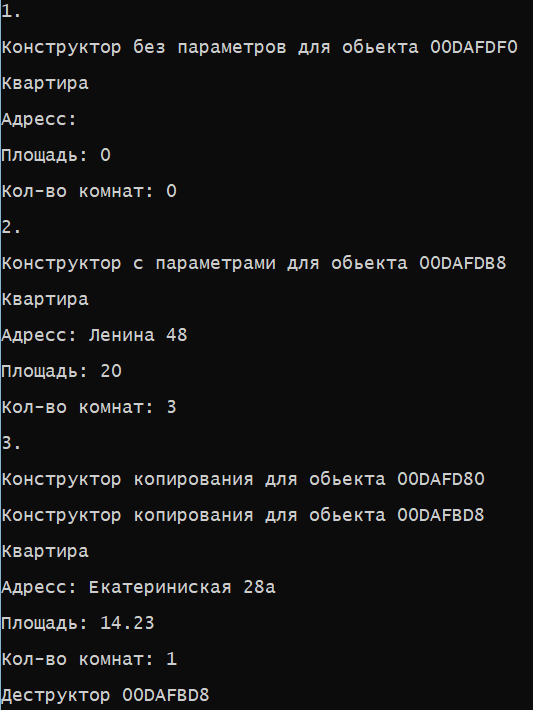
cout << "\nПлощадь: " << d << endl;

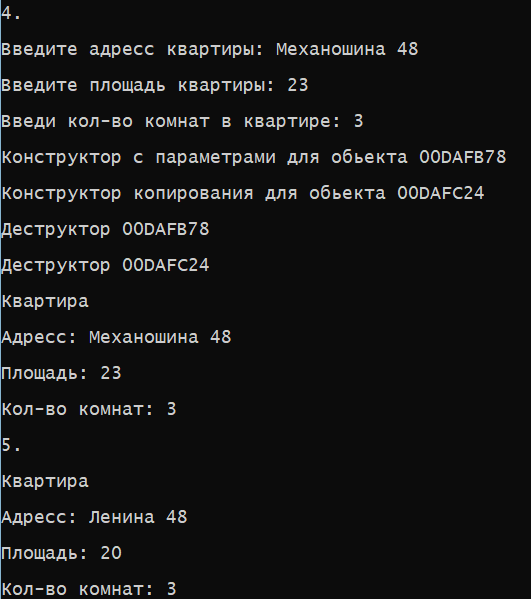
cout << "\nКол-во комнат: " << i << endl;

system("pause");

}

# **Скриншоты тестов**



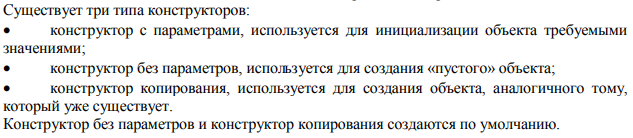


**Контрольные вопросы**

1. Для чего нужен конструктор?

Конструктор необходим для инициализации объекта.

1. Сколько типов конструкторов существует в C++?



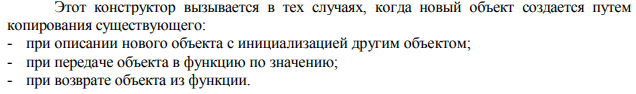
1. Для чего используется деструктор? В каких случаях деструктор описывается явно?

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

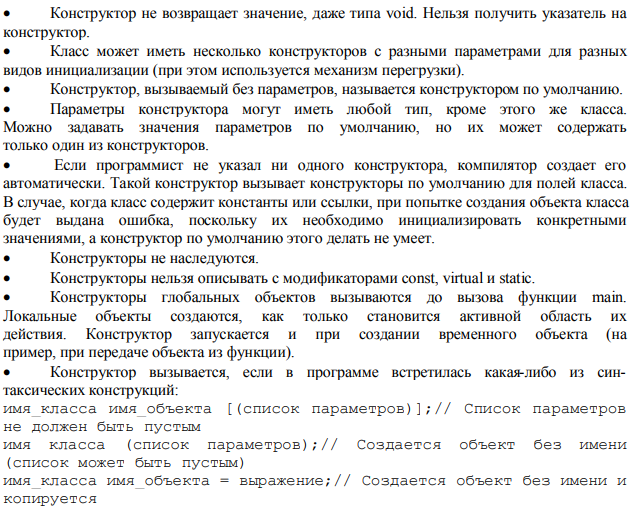
1. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?

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 того же класса.

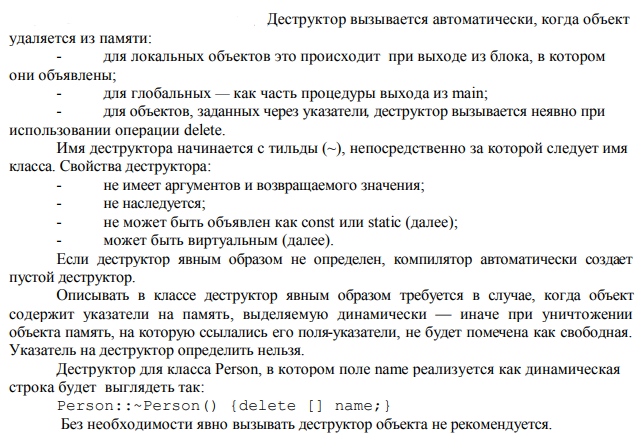
1. В каких случаях вызывается конструктор копирования?



1. Перечислить свойство конструкторов.



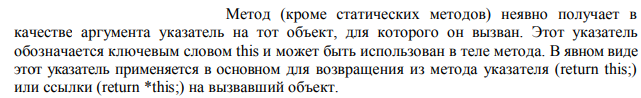
1. Перечислить свойства деструкторов.



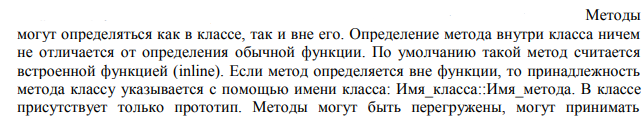
1. К каким атрибутам имеют доступ методы класса?



1. Что представляет собой указатель this?



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



1. Какое значение возвращает конструктор?

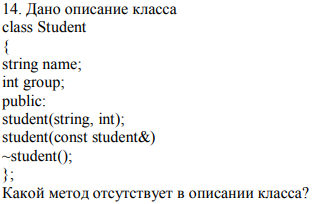
Конструктор не возвращает значения.

1. Какие методы создаются по умолчанию?

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

1. Какое значение возвращает деструктор?

Деструктор не возвращает значения.



Отсутствует конструктор без параметров в описании класса.



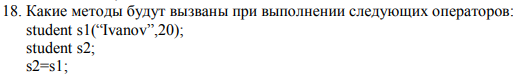
Будет вызван конструктор без параметров.



Будет вызван конструктор с параметрами.



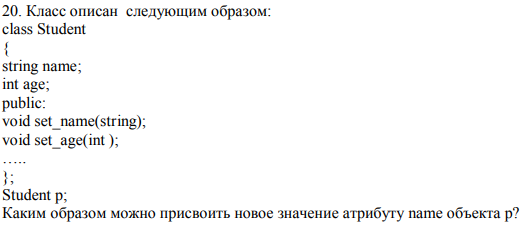
Будут вызваны конструктор с параметрами, а затем конструктор копирования.



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



Конструктор без параметров.



С помощью метода set\_name().