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.»**

Выполнил студент гр. РИС-24-1б

Чижов Денис Николаевич

Проверил:

Доц. Каф. ИТАС 

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Задача:**

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.

Пользовательский класс КВАРТИРА  
Адрес – string  
Площадь – double   
Количество комнат – int

**Анализ:**

**Структура программы**:

* Программа состоит из трёх файлов: заголовочного файла class.h, основного файла main.cpp и файла с реализацией методов methods.cpp.
* Класс point инкапсулирует координаты точки (x, y) и предоставляет методы для работы с ними.

**Функционал класса point**:

* **Init(double x, double y)**: Инициализирует координаты точки значениями x и y.
* **Read()**: Позволяет пользователю ввести координаты точки с клавиатуры.
* **Show()**: Выводит координаты точки на экран.
* **distance()**: Возвращает расстояние от точки до начала координат (0, 0), вычисляемое по формуле: sqrt(x^2 + y^2).

**Работа программы**:

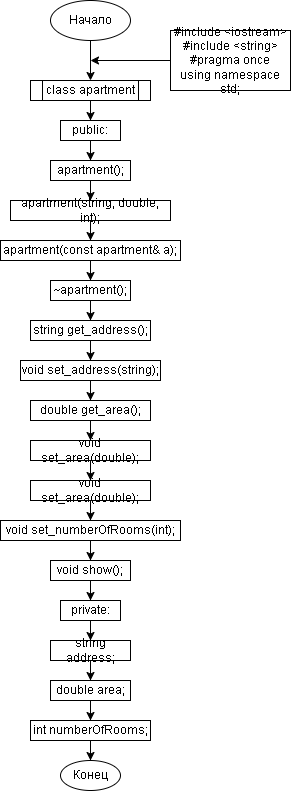
* В main.cpp создаются два объекта класса point: A и B.
* Для объекта A вызывается метод Read(), чтобы пользователь мог ввести координаты, а затем Show() для их отображения.
* Для объекта B координаты инициализируются методом Init(4, 5), а затем выводятся на экран методом Show().
* Для обоих объектов вызывается метод distance(), и результаты выводятся на экран.

**Замечания**:

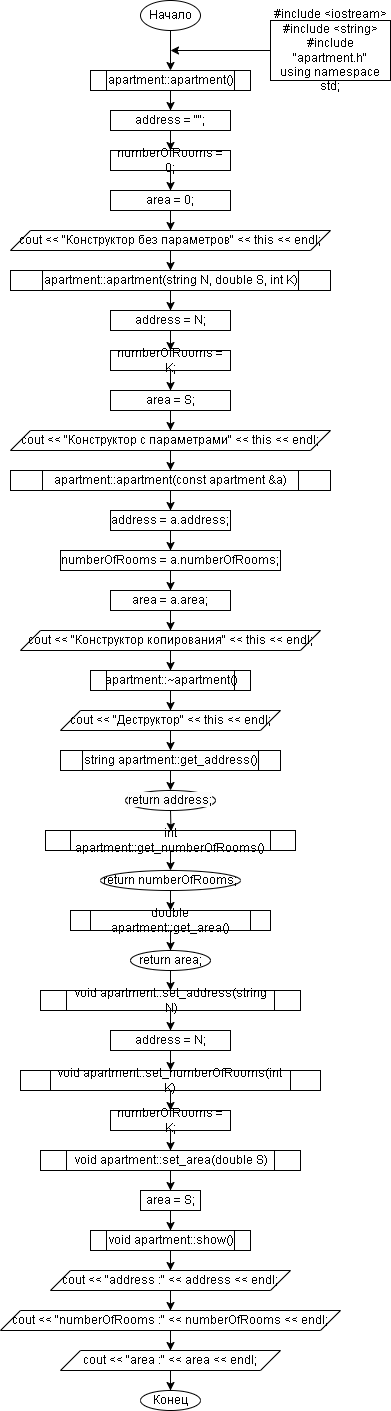
* Метод distance() возвращает расстояние до начала координат, но его название может вводить в заблуждение. Лучше назвать его, например, distanceFromOrigin().
* В методе Show() отсутствуют пробелы между выводимыми значениями, что может ухудшить читаемость вывода.
* Для улучшения пользовательского опыта в методе Read() можно добавить подсказки, например: cout << "Введите первую координату: ";.

**Блок-схема:**

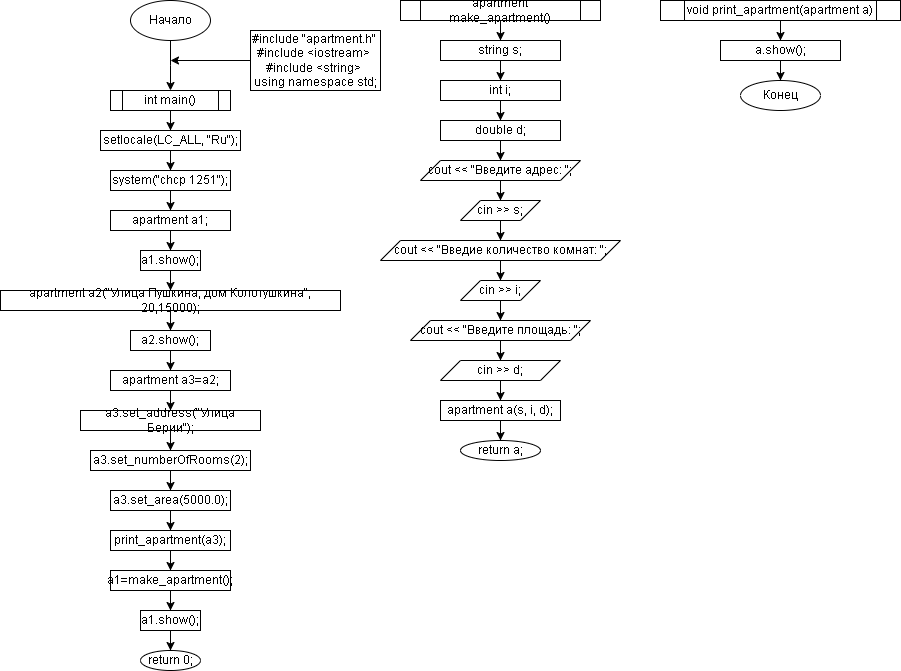
class.h:



methods.cpp:



main.cpp:



**Код:**

apartment.h:

#include <iostream>

#include <string>

#pragma once

using namespace std;

class apartment {

public:

apartment(); //конструктор без параметров

apartment(string, double, int);

apartment(const apartment& a);

~apartment();

string get\_address();

void set\_address(string);

double get\_area();

void set\_area(double);

int get\_numberOfRooms();

void set\_numberOfRooms(int);

void show();

private:

string address;

double area;

int numberOfRooms;

};

apartment.cpp:

#include <iostream>

#include <string>

#include "apartment.h"

using namespace std;

//конструктор без параметров

apartment::apartment()

{

address = "";

numberOfRooms = 0;

area = 0;

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

}

//конструктор с параметрами

apartment::apartment(string N, double S, int K)

{

address = N;

numberOfRooms = K;

area = S;

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

}

//конструктор копирования

apartment::apartment(const apartment &a)

{

address = a.address;

numberOfRooms = a.numberOfRooms;

area = a.area;

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

}

//деструктор

apartment::~apartment()

{

cout << "Деструктор" << this << endl;

}

//селекторы

string apartment::get\_address()

{

return address;

}

int apartment::get\_numberOfRooms()

{

return numberOfRooms;

}

double apartment::get\_area()

{

return area;

}

//модификаторы

void apartment::set\_address(string N)

{

address = N;

}

void apartment::set\_numberOfRooms(int K)

{

numberOfRooms = K;

}

void apartment::set\_area(double S)

{

area = S;

}

//метод для просмотра атрибутов

void apartment::show()

{

cout << "address :" << address << endl;

cout << "numberOfRooms :" << numberOfRooms << endl;

cout << "area :" << area << endl;

}

main.cpp:

#include "apartment.h"

#include <iostream>

#include <string>

using namespace std;

//функция для возврата объекта как результата

apartment make\_apartment()

{

string s;

int i;

double d;

cout << "Введите адрес: ";

cin >> s;

cout << "Введие количество комнат: ";

cin >> i;

cout << "Введите площадь: ";

cin >> d;

apartment a(s, i, d);

return a;

}

//функция для передачи объекта как параметра

void print\_apartment(apartment a)

{

a.show();

}

int main()

{

setlocale(LC\_ALL, "Ru");

system("chcp 1251");

//конструктор без параметров

apartment a1;

a1.show();

//коструктор с параметрами

apartment a2("Улица Пушкина, дом Колотушкина", 20,15000);

a2.show();

//конструктор копирования

apartment a3=a2;

a3.set\_address("Улица Берии");

a3.set\_numberOfRooms(2);

a3.set\_area(5000.0);

//конструктор копирования

print\_apartment(a3);

//конструктор копирования

a1=make\_apartment();

a1.show();

return 0;

}

**Вывод:**

Программа работает. Задача решена.