Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.01 – «Информатика и вычислительная техника»

**ОТЧЁТ**

**Лабораторная работа №2**

**«Конструктор класса»**

Выполнила работу:

студентка гр. ИВТ-24-2б

Малая Алина Александровна

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

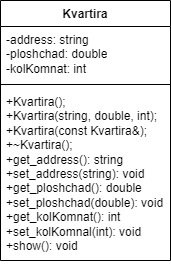
Пользовательский класс КВАРТИРА

Адрес – string

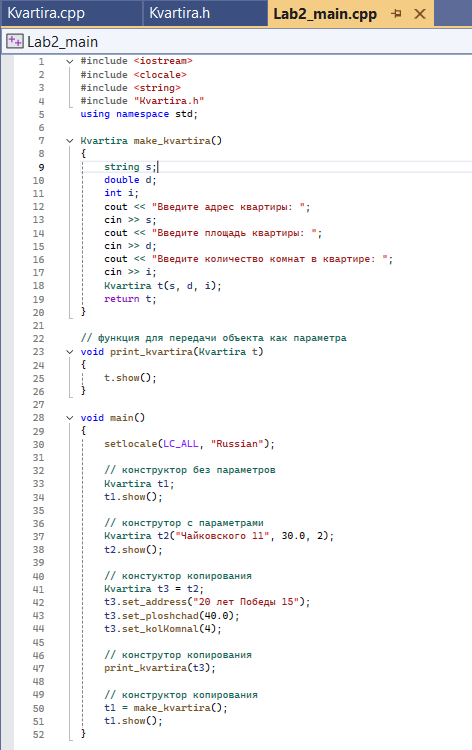
Площадь – double

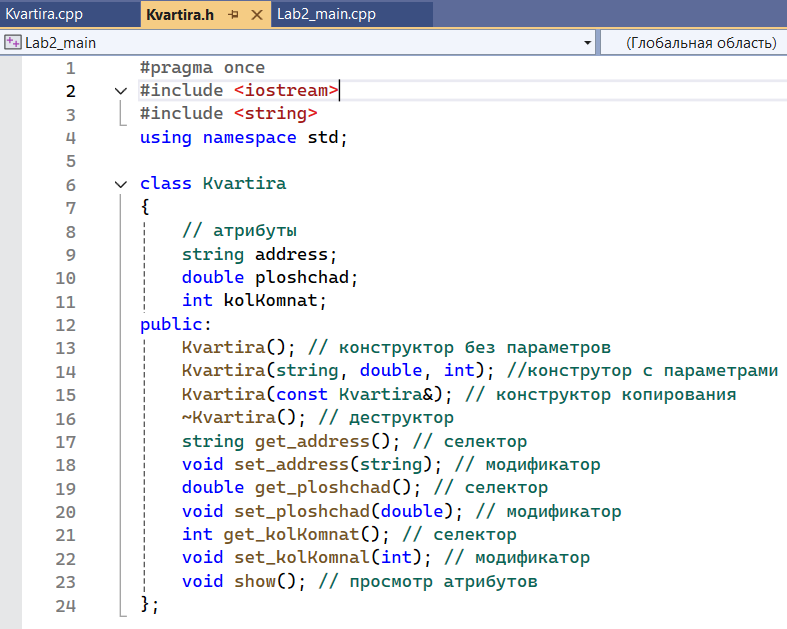
Количество комнат – int

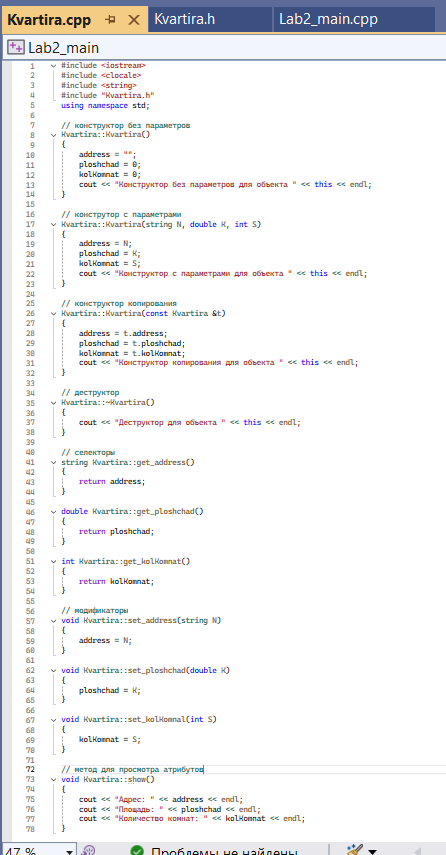
**Блок – схема**

****

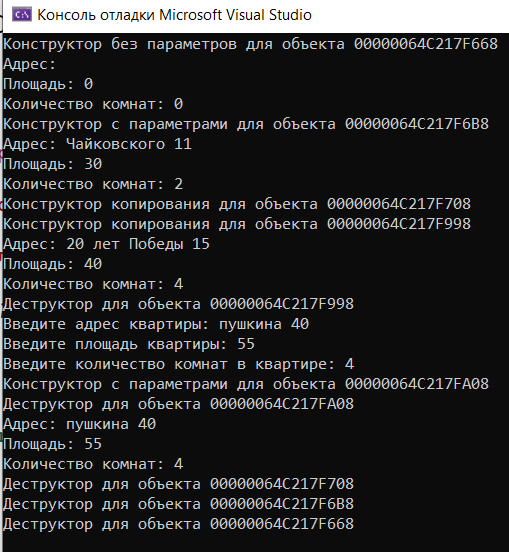
**Код программы**

****

****

****

**Вывод**

****

**Контрольные вопросы**

1. Для чего нужен конструктор?

Конструктор - это специальный метод класса, который вызывается автоматически при создании объекта этого класса. Он используется для:

• Инициализации атрибутов объекта (присвоения начальных значений).

• Выполнения любых действий, необходимых для правильной подготовки объекта к работе.

• Гарантии того, что объект будет создан в допустимом состоянии.

2. Сколько типов конструкторов существует в C++?

Основные типы конструкторов в C++:

• Конструктор по умолчанию (Default constructor): Конструктор без аргументов (или все аргументы имеют значения по умолчанию).

• Конструктор с параметрами (Parameterized constructor): Конструктор, который принимает один или несколько аргументов.

• Конструктор копирования (Copy constructor): Конструктор, который создает новый объект как копию существующего.

• Конструктор перемещения (Move constructor): Конструктор, который "перемещает" ресурсы из одного объекта в другой (для повышения производительности). Этот конструктор появился в C++11.

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

Деструктор - это специальный метод класса, который вызывается автоматически при уничтожении объекта этого класса. Он используется для:

• Освобождения ресурсов, занятых объектом (например, выделенной динамической памяти).

• Выполнения любых завершающих действий, необходимых перед уничтожением объекта.

Деструктор описывается явно в тех случаях, когда:

• Класс управляет динамической памятью (например, выделяет память с помощью new и нужно освободить с помощью delete).

• Класс содержит ресурсы, которые требуют явного освобождения (например, открытые файлы, сетевые соединения).

• Логика уничтожения объекта требует особых действий.

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

• Конструктор без параметров (Default constructor): Для создания объектов без необходимости передачи начальных значений. Он создает объект в некотором "стандартном" состоянии.

• Конструктор копирования (Copy constructor): Для создания нового объекта, который является копией существующего. Это важно при передаче объектов по значению, возврате объектов из функций и других операциях, где требуется создать копию.

• Конструктор с параметрами (Parameterized constructor): Для создания объектов с заданными начальными значениями, определяемыми параметрами. Позволяет инициализировать объект нужными данными при создании.

5. В каких случаях вызывается конструктор копирования?

Конструктор копирования вызывается в следующих случаях:

• Когда объект передается в функцию по значению (не по ссылке и не по указателю).

• Когда функция возвращает объект по значению.

• Когда объект создается и инициализируется другим существующим объектом того же класса (например, Student s2 = s1;).

• Когда компилятор генерирует временный объект, который является копией.

6. Перечислите свойства конструкторов.

• Имя конструктора совпадает с именем класса.

• Конструктор не возвращает значение (даже void не указывается).

• Конструктор может иметь любое количество параметров (включая отсутствие параметров - конструктор по умолчанию).

• Класс может иметь несколько конструкторов (перегрузка конструкторов).

• Конструктор вызывается автоматически при создании объекта.

• Конструктор может быть public, protected или private.

• Если в классе не определен ни один конструктор, компилятор автоматически сгенерирует конструктор по умолчанию (только если в классе нет константных полей без инициализации).

7. Перечислите свойства деструкторов.

• Имя деструктора совпадает с именем класса, но начинается с символа ~ (тильда).

• Деструктор не принимает аргументов.

• Деструктор не возвращает значение (даже void не указывается).

• В классе может быть только один деструктор.

• Деструктор вызывается автоматически при уничтожении объекта.

• Деструктор может быть только public или protected. Private деструктор запрещает создание объектов класса в принципе, кроме как внутри самого класса (обычно используют в синглтонах).

• Если в классе не определен деструктор, компилятор автоматически сгенерирует деструктор (если это необходимо).

8. К каким атрибутам имеют доступ методы класса?

Методы класса имеют доступ ко всем атрибутам класса (и private, и protected, и public). Это одно из основных свойств инкапсуляции.

9. Что представляет собой указатель this?

this - это неявный указатель, который доступен внутри любого нестатического метода класса. Он указывает на объект, для которого был вызван данный метод. this позволяет:

• Обращаться к атрибутам текущего объекта, особенно если имя аргумента метода совпадает с именем атрибута класса.

• Возвращать текущий объект из метода (например, для реализации цепочки вызовов).

• Передавать текущий объект в другие функции.

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

В основном, разница заключается в следующем:

• Метод, определенный внутри класса, автоматически считается встроенным (inline). Компилятор может встроить его код непосредственно в место вызова, что может повысить производительность (но это не всегда так).

• Метод, определенный вне класса, требует явного указания области видимости класса с использованием оператора :: (двойное двоеточие).

• Оба типа методов имеют одинаковый доступ ко всем членам класса (если объявлены как public, protected или private в соответствии с потребностями класса).

11. Какое значение возвращает конструктор?

Конструктор не возвращает никаких значений (даже void не указывается). Конструктор создаёт объект, а не возвращает его.

12. Какое значение возвращает деструктор?

Деструктор не возвращает никаких значений (даже void не указывается`).

13. Какие методы создаются по умолчанию?

По умолчанию компилятор может создать следующие методы:

• Конструктор по умолчанию (если не определен ни один конструктор).

• Конструктор копирования (если не определен).

• Оператор присваивания копированием (copy assignment operator) (если не определен).

• Деструктор (если не определен).

• Конструктор перемещения (move constructor) (если не определен, и не определен конструктор копирования, деструктор или оператор присваивания копированием).

• Оператор присваивания перемещением (move assignment operator) (если не определен, и не определен конструктор копирования, деструктор или оператор присваивания копированием).

14. Какой метод отсутствует в описании класса?

В описании класса отсутствует оператор присваивания копированием (copy assignment operator), который выглядит примерно так:

```c++

Student& operator=(const Student& other);

```

И оператор присваивания перемещением (move assignment operator), которого не будет без реализации первых 4 методов, описанных в ответе на вопрос 13

15. Какой метод будет вызван при выполнении следующих операторов: Student\* s; s=new Student;?

Будет вызван конструктор по умолчанию: Student().

16. Какой метод будет вызван при выполнении следующих операторов: Student s("Ivanov", 20);?

Будет вызван конструктор с параметрами: Student(string, int).

17. Какие методы будут вызваны при выполнении следующих операторов: Student s1("Ivanov", 20); Student s2=s1;?

1. Student(string, int) (для s1).

2. Student(const Student&) (конструктор копирования для s2).

18. Какие методы будут вызваны при выполнении следующих операторов: Student s1("Ivanov", 20); Student s2; s2=s1;?

1. Student(string, int) (для s1).

2. Student() (конструктор по умолчанию для s2).

3. Оператор присваивания копированием (copy assignment operator) для присваивания s1 к s2. Так как оператор не задан явно, будет использоваться сгенерированный компилятором.

19. Какой конструктор будет использоваться при передаче параметра в функцию print(): void print(student a){a.show();}?

Будет использоваться конструктор копирования: Student(const Student& a). При передаче объекта по значению в функцию создаётся его копия, и для этого вызывается конструктор копирования.

20. Каким образом можно присвоить новое значение атрибуту name объекта p

```c++

p.set\_name("Новое Имя");

```

**GitHub**

<https://github.com/amalayaa>