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 1.**

**" Классы и объекты. Инкапсуляция"**

Выполнила студентка гр. РИС-24-3б

Черепанова Диана Романовна

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

1. Реализовать определение нового класса. Для демонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.

2. Структура-пара - структура с двумя полями, которые  имеют имена first и second. Требуется реализовать тип данных с помощью такой структуры. Во всех заданиях должны присутствовать:

a. метод инициализации Init (метод должен контролировать значения

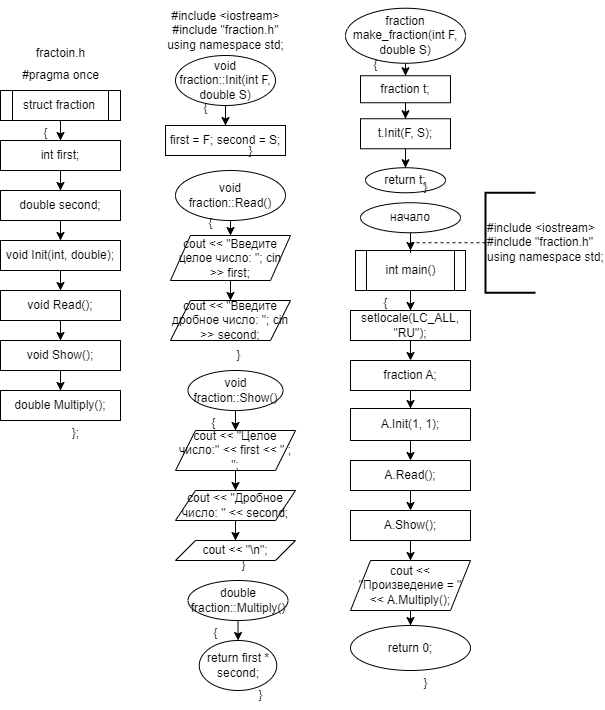
аргументов на корректность);

b. Ввод с клавиатуры Read;

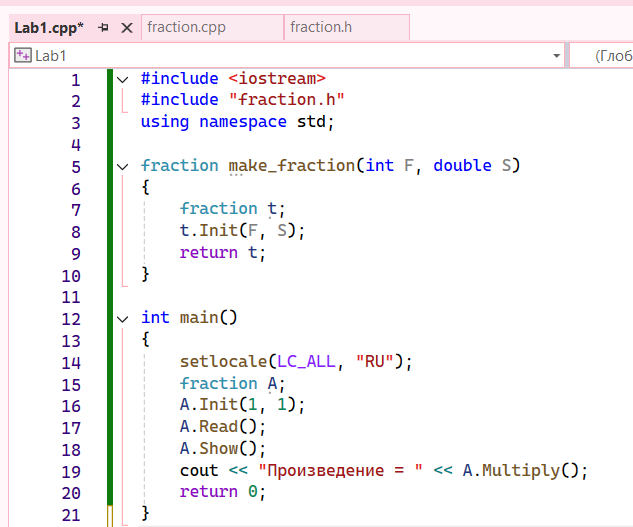
с. Вывод на экран Show.

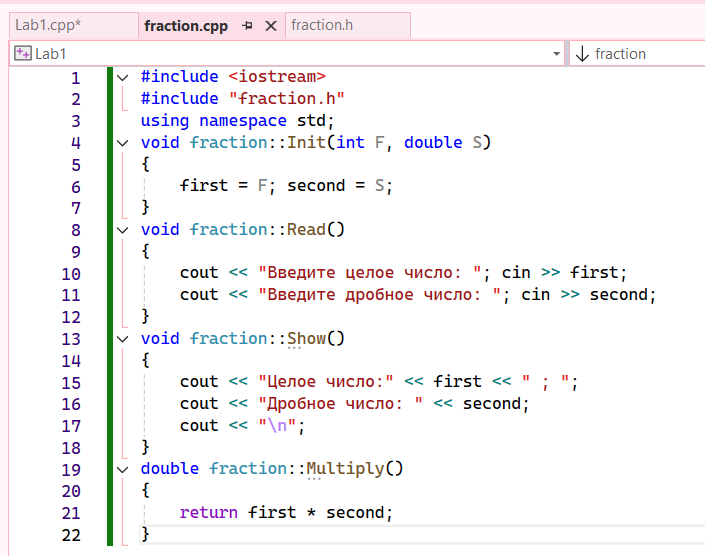
3. Поле first – положительное целое число, целая часть числа, поле second – положительное дробное число, дробная часть числа. Реализовать метод multiply (double k) – умножение на вещественное число k.

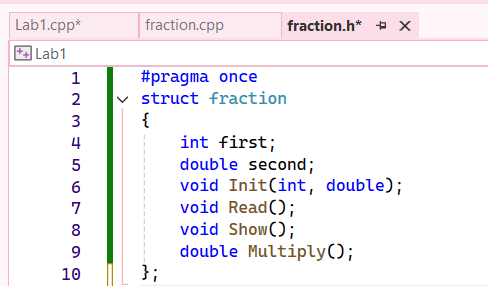
**Блок-схема**

****

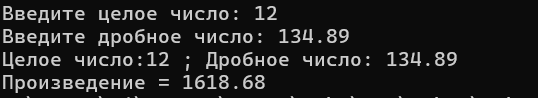
**Код**

****

****

****

**Вывод**

****

**Вопросы:**

1. Что такое класс?

Класс – это абстрактный тип данных, определяемый пользователем. Представляет собой модель реального объекта в виде данных и функций для работы с ними.

1. Что такое объект (экземпляр) класса?

Объект (экземпляр) класса - это переменная типа класса. Их может быть бесконечное количество.

1. Как называются поля класса?

Поля - переменные внутри класса, также называются атрибутами класса.

1. Как называются функции класса?

Классы имеют свои функции, которые называются методами класса.

1. Для чего используются спецификаторы доступа?

Спецификаторы доступа используются для изменения видимость компонентов. Всего есть 3 спецификатора public, private, protected. В структуре (struct) все поля и методы по умолчанию имеют модификатор доступа public, а в классе (class) все поля и методы по умолчанию имеют модификатор доступа private.

6. Для чего используется спецификатор public?

При спецификаторе доступа public общедоступные компоненты доступны в любой части программы. Они могут быть использованы любой функцией как внутри класса, так и вне его. Доступ из вне осуществляется через имя объекта.

7.Для чего используется спецификатор private?

При спецификаторе доступа private собственные компоненты локализованы в классе и не доступны извне. Они могут использоваться функциями - членами данного класса и функциями - "друзьями" того класса, в котором они описаны.

8.Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?

По умолчанию будет исполльзоваться private.

9. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?

По умолчанию будет использоваться public.

10. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?

Спецификатор public - для доступа к методам интерфейса.

11.Каким образом можно изменить значения атрибутов экземпляра класса?

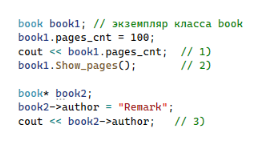


12. Каким образом можно получить значения атрибутов экземпляра класса?

1. Напрямую

2. Через public методы классса

3. Через указатель.



13.Класс описан следующим образом struct Student { string name; int group; ……… }; Объект класса определен следующим образом Student \*s=new Student; Как можно обратиться к полю name объекта s?

s.name = "name"

14.Класс описан следующим образом class Student

{ string name; int group; ……. }; Объект класса определен следующим образом Student \*s=new Student; Как можно обратиться к полю name объекта s?

s->name

15.Класс описан следующим образом class Student { string name; int group; public: ….. }; Объект класса определен следующим образом Student s; Как можно обратиться к полю name объекта s?

s.name

16.Класс описан следующим образом class Student { public: char\* name; int group; ……… }; Объект класса определен следующим образом Student \*s=new Student; Как можно обратиться к полю name объекта s?

s->name