Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**

**о работе по информатике**

Семестр: 2

На тему: Лабораторная работа №1: «Классы и объекты. Инкапсуляция».

**Вариант 13**

Выполнил студент ИВТ-22-2б:

Коняев Александр Сергеевич

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

Пермь 2023

**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 обычно 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;

c. вывод на экран Show.

1. Реализовать внешнюю функцию make\_тип(), где тип – тип реализуемой структуры. Функция должна получать значения для полей структуры как параметры функции и возвращать структуру как результат. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

**Анализ задачи**

* Метод Init – инициализация переменных
* Метод Show – вывод значений переменных
* Метод Read – ввод значений переменных с клавиатуры
* Метод make – умножение на вещественное число k

**UML – диаграмма**

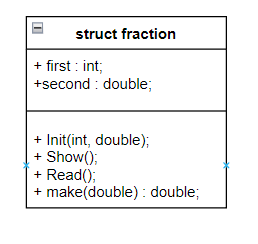


Рисунок 1 – UML-диаграмма структуры «fraction»

**Код программы.**

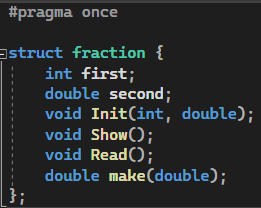


Рисунок 2 – Описание класса «fraction»

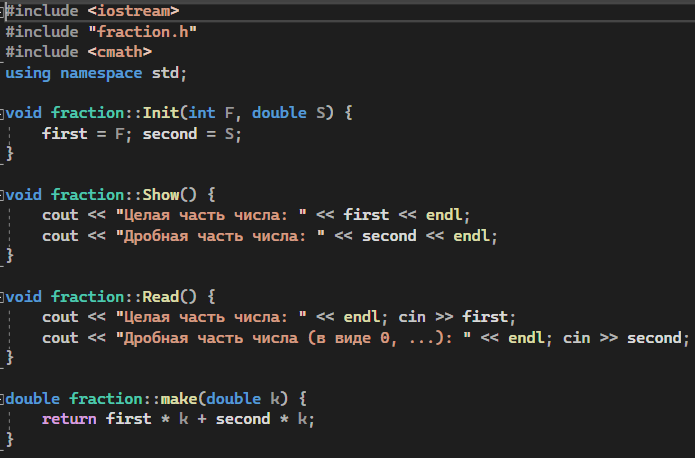


Рисунок 3 – Класс «fraction»

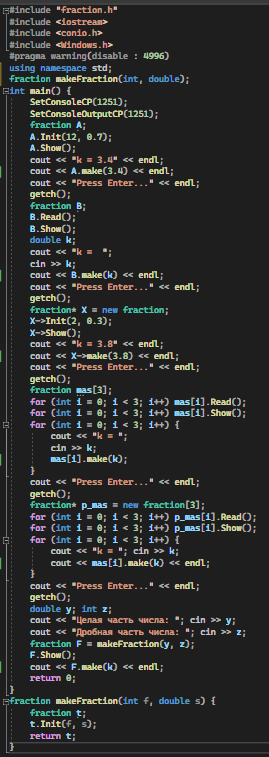


Рисунок 4 – Функция main

**Вывод программы.**

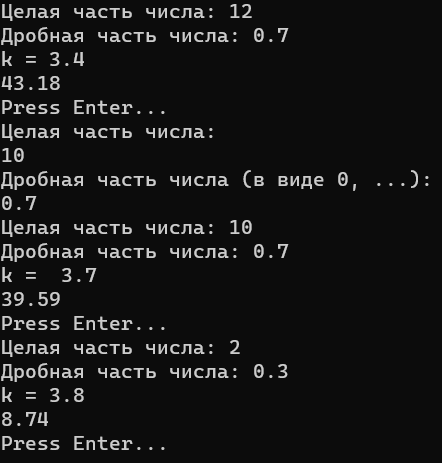
****

Рисунок 5 – Вывод программы

**Ответы на вопросы**

1. Класс – абстрактный тип данных, шаблон, определяющий структуру и поведение объектов, которые создаются на его основе. Класс содержит описание свойств (переменных) и методов (функций), которые могут быть использованы для работы с объектами данного класса.
2. Объект класса - это экземпляр класса, созданный на основе его шаблона. Он содержит конкретные значения свойств (переменных) и может использовать методы (функции), определенные в классе. Объекты класса могут иметь различные значения свойств, но будут иметь общую структуру и поведение, определенные в классе. Каждый объект класса является отдельным экземпляром и может быть обработан индивидуально.
3. Поля – атрибуты класса, составляющие его структуру.
4. Методы – функции класса.
5. Спецификаторы доступа управляют областью памяти.
6. Public: поля и методы доступны в любой части программы.
7. Private: доступ к методам и классам разрешён только внутри этого класса.
8. Private.
9. Public.
10. Protected.
11. Значения атрибутов экземпляра класса можно изменить, используя оператор присваивания. Например, если у нас есть класс Person с атрибутами name и age, мы можем изменить значение атрибута name объекта person следующим образом:

person.name = "John"

Аналогично, мы можем изменить значение атрибута age:

person.age = 30

Также можно изменить значения атрибутов экземпляра класса, вызывая методы класса, которые изменяют значения атрибутов. Например, если у нас есть метод setage(), который устанавливает новое значение для атрибута age, мы можем вызвать его следующим образом:

person.setage(35)

1. Чтобы получить значение атрибутов экземпляра класса, можно использовать оператор доступа к атрибутам "." и указать имя атрибута. Например, чтобы получить значение атрибута name объекта person, мы можем написать:

person.name

Аналогично, чтобы получить значение атрибута age, мы можем написать:

person.age

Также можно определить методы класса, которые возвращают значения атрибутов. Например, если у нас есть метод getage(), который возвращает значение атрибута age, мы можем вызвать его следующим образом:

person.getage()

1. s->name
2. s.name
3. К полю name можно обратиться через указатель (s->name).
4. К полю name можно обратиться только через метод реализованный в этом классе, который работает с этим полем и имеет параметр доступа public.
5. К полю name можно обратиться через указатель (s->name).