|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №1  по дисциплине «Языки программирования»  Вариант 5 | | |
|  | | Работу выполнил  студент группы ПМИ-1,2-2023 2 курса  Волегов И.С.  «\_\_\_» \_\_\_\_\_\_\_\_2024 г. |
| Работу проверил  Ракина В.Д.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Задание 1 5](#_Toc181660481)

[Текст задания 5](#_Toc181660482)

[Алгоритм решения 5](#_Toc181660488)

[Тестирование 6](#_Toc181660489)

[Код программы 7](#_Toc181660491)

[Задание 2 8](#_Toc181660492)

[Текст задания 8](#_Toc181660493)

[Алгоритм решения 8](#_Toc181660494)

[Тестирование 9](#_Toc181660495)

[Код программы 9](#_Toc181660497)

[Задание 3 9](#_Toc181660499)

[Текст задания 9](#_Toc181660500)

[Алгоритм решения 10](#_Toc181660501)

[Тестирование 11](#_Toc181660502)

[Код программы 11](#_Toc181660503)

[Задание 4 11](#_Toc181660505)

[Текст задания 11](#_Toc181660506)

[Алгоритм решения 12](#_Toc181660507)

[Тестирование 12](#_Toc181660508)

[Код программы 13](#_Toc181660509)

[Задание 6 13](#_Toc181660510)

[Текст задания 13](#_Toc181660511)

[Алгоритм решения 13](#_Toc181660519)

[Тестирование 14](#_Toc181660520)

[Код программы 15](#_Toc181660521)

# Задание 1

## Текст задания

## Точка координат. Создайте сущность Точка, расположенную на двумерной плоскости, которая описывается:

## • Координата Х: число

## • Координата Y: число

## • Может возвращать текстовое представление вида “{X;Y}”

## Необходимо создать три точки с разными координатами и вывести на экран их текстовое представление.

## Алгоритм решения

**1.Класс Point:**

**Класс Point реализует следующие функциональные возможности:**

**-Поля класса:**

**1.private double x: координата X точки.**

**2.private double y: координата Y точки.**

**-Конструктор:**

**public Point(double x, double y): инициализирует объект Point с заданными координатами.**

**-Методы:**

**1.public double getX(): возвращает значение координаты X.**

**2.public double getY(): возвращает значение координаты Y.**

**3.public String toString(): возвращает строковое представление точки в формате {x;y}.**

**2. Класс Main**

**Класс Main обеспечивает пользовательский интерфейс и реализацию логики программы.**

**- Метод main:**

**1.Вызывает метод createPoint три раза для создания трех объектов типа Point, запрашивая у пользователя координаты каждой точки.**

**2.Выводит информацию о введенных точках.**

**- Метод createPoint:**

**1.Запрашивает у пользователя ввод координат через консоль.**

**2.Проверяет корректность ввода (ввод должен содержать две числовые координаты, разделенные пробелом).**

**3.Если ввод некорректен, программа сообщает об ошибке и запрашивает повторный ввод.**

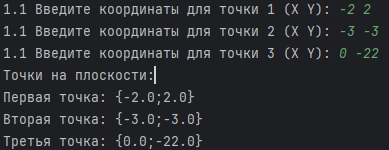
**4.Создает и возвращает объект Point с введенными координатами.**

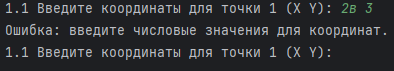
**- Метод isNumeric:**

**1.Проверяет, является ли переданная строка числом, позволяя использовать знаки минус и точку для дробных значений.**

## Тестирование

## 







## Код программы

https://github.com/HintMSI/laba2-Volegov-Ilya

# Задание 2

## Текст задания

Имена. Создайте сущность Имя, которая описывается тремя параметрами: Фамилия, Личное имя, Отчество. Имя может быть приведено к строковому виду, включающему традиционное представление всех трех параметров: Фамилия Имя Отчество (например “Иванов Иван Иванович”). Необходимо предусмотреть возможность того, что какой-либо из параметров может быть не задан, и в этом случае он не учитывается при приведении к текстовому виду. Необходимо создать следующие имена:

• Клеопатра

• Пушкин Александр Сергеевич

• Маяковский Владимир

Обратите внимание, что при выводе на экран, не заданные параметры никак не участвуют в образовании строки.

## Алгоритм решения

**Программа состоит из двух классов: Main и Name.**

**- Класс Name:**

**1.Представляет объект с фамилией, именем и отчеством.**

**2.Содержит конструктор для инициализации этих полей.**

**3.Реализует метод toString(), который возвращает строковое представление объекта в формате "Фамилия Имя Отчество", исключая лишние пробелы.**

**- Класс Main:**

**1.Отвечает за взаимодействие с пользователем.**

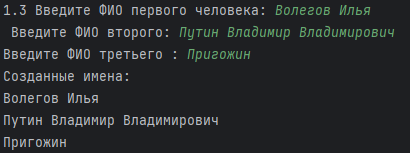
**2.Запрашивает у пользователя ввод ФИО трех человек.**

**3.Создает объекты класса Name на основании введенных данных.**

**4.Выводит созданные имена на экран.**

## Тестирование

## 



## Код программы

# <https://github.com/HintMSI/laba2-Volegov-Ilya>

# Задание 3

## Текст задания

Прямая. Создайте сущность Линия, расположенную на двумерной плоскости, которая описывается:

• Координата начала: Точка

• Координата конца: Точка

• Может возвращать текстовое представление вида “Линия от {X1;Y1} до {X2;Y2}”

Для указания координат нужно использовать сущность Точка, разработанную в задании 1.1. Создайте линии со следующими характеристиками:

1. Линия 1 с началом в т. {1;3} и концом в т.{23;8}.

2. Линия 2, горизонтальная, на высоте 10, от точки 5 до точки 25.

3. Линия 3, которая начинается всегда там же, где начинается линия 1, и заканчивается всегда там же, где заканчивается линия 2. Таким образом, если положение первой или второй линии меняется, то меняется и третья линия.

4. После создания всех трех объектов измените координаты первой и второй линий, причем сделайте это таким образом, чтобы положение третьей линии соответствовало требованиям пункта 3.

5. Измените координаты первой линии так, чтобы при этом не изменились, координаты третьей линии.

## Алгоритм решения

**- Класс Point:**

**1.Представляет точку на плоскости с координатами (x, y).**

**2.Содержит конструктор для инициализации полей.**

**3.Предоставляет методы для получения значений координат (getX() и getY()).**

**4.Реализует метод toString(), который возвращает строковое представление точки в формате {x;y}.**

**- Класс Line:**

**1.Представляет линию, заданную двумя точками: начальной и конечной.**

**2.Содержит конструктор для инициализации начальной и конечной точек.**

**3.Предоставляет методы для установки новых начальной и конечной точек (setStart() и setEnd()), а также для получения текущих значений (getStart() и getEnd()).**

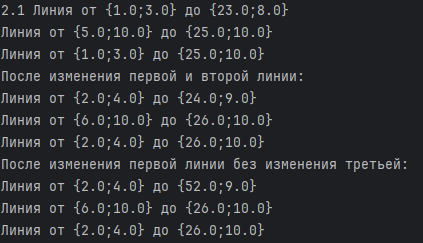
**4.Реализует метод toString(), который возвращает строковое представление линии в формате "Линия от {x1;y1} до {x2;y2}".**

**- Класс Main:**

**1.Создает несколько объектов типа Point и Line, выводит их на экран.**

**2.Изменяет свойства линий и демонстрирует результат изменений**.

## Тестирование



## Код программы

# <https://github.com/HintMSI/laba2-Volegov-Ilya>

# Задание 4

## Текст задания

Города. Создайте сущность Город, которая будет представлять собой точку на карте со следующими характеристиками:

• Название города

• Набор путей к следующим городам, где путь представляет собой сочетание Города и стоимости поездки в него.

Кроме того, Город может возвращать текстовое представление, в виде названия города и списка связанных с ним городов (в виде пары: “название:стоимость”). Используя разработанную сущность реализуйте схему, представленную на рисунке 2.

## Алгоритм решения

**- Класс City:**

**1.Представляет город и содержит информацию о его названиях и путях к другим городам.**

**2.Имеет массив объектов типа Path, который хранит пути, ведущие к другим городам.**

**3.Содержит метод addPath(City destination, int cost), который позволяет добавлять путь к другому городу с заданной стоимостью.**

**4.Реализует метод toString(), возвращающий строковое представление города и всех его путей в формате "Город: Город A (стоимость), Город B (стоимость), ...".**

**- Класс Path:**

**1.Представляет путь от одного города к другому и содержит информацию о городе назначения и стоимости пути.**

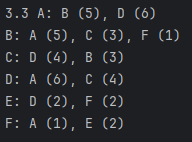
**2.Содержит методы getDestination() и getCost(), которые позволяют получить значения города назначения и стоимости пути соответственно.**

**- Класс Main:**

**1.Создает несколько экземпляров класса City и устанавливает пути между ними с помощью метода addPath().**

**2.Выводит на экран информацию о каждом городе и его путях.**

## Тестирование



## Код программы

https://github.com/HintMSI/laba2-Volegov-Ilya

# Задание 6

## Текст задания

## Дроби. Создайте сущность Дробь со следующими особенностями:

## • Имеет числитель: целое число

## • Имеет знаменатель: целое число

## • Дробь может быть создана с указанием числителя и знаменателя

## • Может вернуть строковое представление вида “числитель/знаменатель”

## • Может выполнять операции сложения, вычитания, умножения и деления с другой Дробью или целым числом.

## Результатом операции должна быть новая Дробь (таким образом,обе исходные дроби не изменяются) Затем необходимо выполнить следующие задачи: 1. Создать несколько экземпляров дробей. 2. Написать по одному примеру использования каждого метода. 3. Вывести на экран примеры и результаты их выполнения в формате «1/3 \* 2/3 = 2/9» 4. Посчитать f1.sum(f2).div(f3).minus(5)

## Алгоритм решения

- Создание класса **Fraction** с приватными полями **numerator** (числитель) и **denominator** (знаменатель).

- Реализация конструктора класса **Fraction**, который принимает числитель и знаменатель и инициализирует объект дроби.

- Реализация методов для выполнения операций над дробями:

1.Метод **sum(Fraction other)** для сложения двух дробей.

2.Метод **minus(int number)** для вычитания целого числа из дроби.

3.Метод **div(Fraction other)** для деления одной дроби на другую.

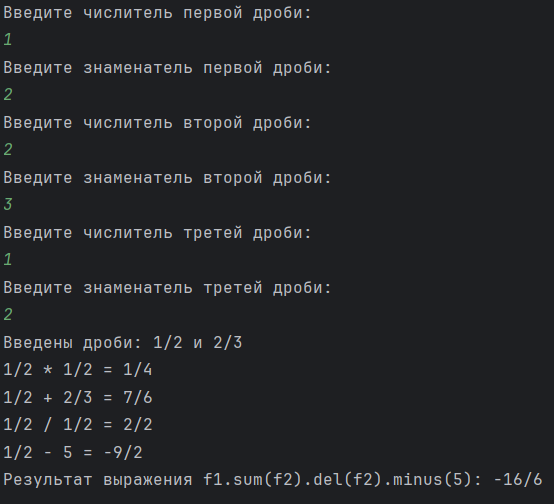
4.Метод **multiply(Fraction other)** для умножения двух дробей.

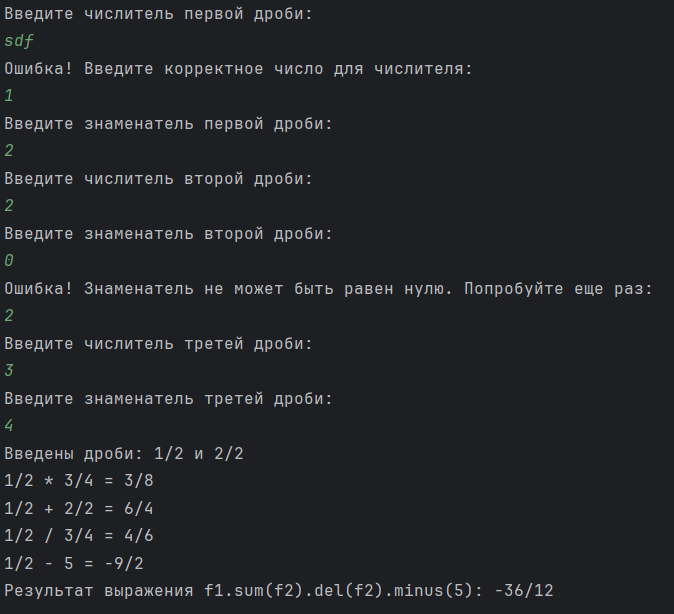
- Переопределение метода **toString(),** чтобы корректно выводить дробь в виде строки.

- В методе **main** осуществляется ввод пользователем числителя и знаменателя двух дробей, создание объектов **Fraction** для каждой дроби и выполнение операций с ними.

- Вывод результатов операций на экран, включая умножение, сложение, деление и вычитание дробей, а также комбинированный пример, используя несколько операций подряд.

## Тестирование





## Код программы

https://github.com/HintMSI/laba2-Volegov-Ilya