Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)

Факультет: Информатика и системы управления
Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа №3-2

по курсу: «Языки и методы программирования»

«Полиморфизм на основе интерфейсов в языке Java»

Выполнил:
Студент группы ИУ9-21Б

Проверил:
Посевин Д. П.

Москва, 2024

1. Цель

Приобретение навыков реализации интерфейсов для обеспечения возможности полиморфной обработки объектов класса.

2. Персональный вариант

Класс, представляющий множество арифметических прогрессий, с порядком на основе количества чисел из интервала (0; 100), принадлежащих прогрессиям множества.

3. Решение

3.1. Код

```
import java.util.*;

public class Test {
    static ArithmeticProgression
     ↪  genNewArithemticProgression(Random rand) {
        int length = rand.nextInt(1, 10);
        double first = rand.nextInt(-10, 100),
               step = rand.nextInt(1, 20);
        return new ArithmeticProgression(first, step, length);
    }

    public static void main(String[] args) {
        Random rand = new Random();
        ArrayList<ArithmeticProgression> a = new
         ↪  ArrayList<ArithmeticProgression>();
        int m = rand.nextInt(3, 10);
        for (int i = 0; i < m; i++)
            a.add(genNewArithemticProgression(rand));

        Collections.sort(a);

        for (ArithmeticProgression ap : a)
            ap.print();
    }
}
```

Код 1: Test.java

```java
public class ArithmeticProgression implements
 ↪  Comparable<ArithmeticProgression> {
    private double first, step;
    private int length;

    public ArithmeticProgression(double first, double step, int
     ↪  length) {
        this.first = first;
        this.length = length;
        this.step = step;
    }

    public void print() {
        System.out.printf("First: %f; Step: %f; Length: %d\n",
         ↪  first, step, length);
        System.out.printf("Points in (0, 100): %d\n",
         ↪  this.countPoints());
    }

    public int countPoints() {
        int count = 0;
        double value = this.first;

        for (int k = 0; k < this.length && value < 100; k++,
         ↪  value += step)
            if (0 < value)
                count++;

        return count;
    }

    public int compareTo(ArithmeticProgression other) {
        return this.countPoints() - other.countPoints();
    }
}
```

Код 2: ArithmeticProgression.java

3.2. Скриншоты

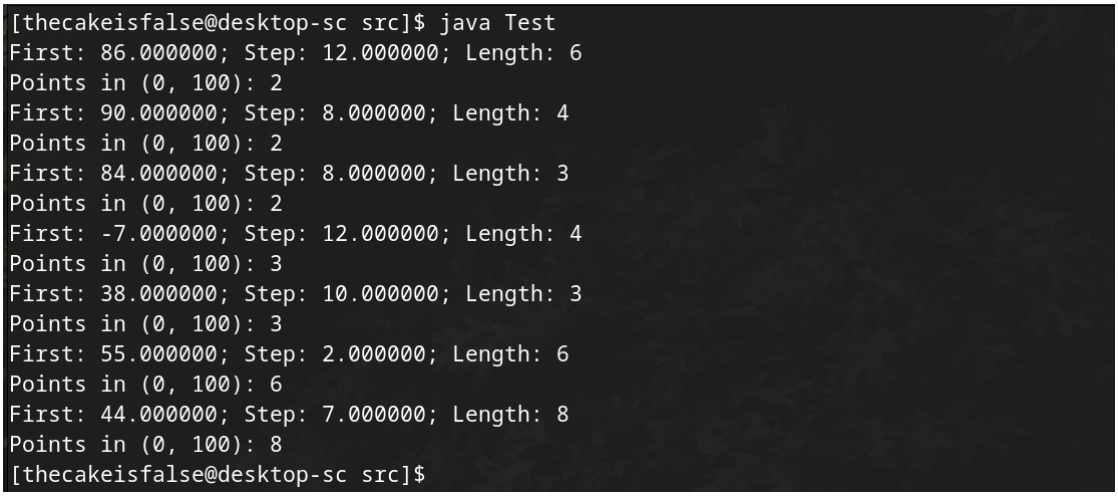

Рис. 1: Пример работы

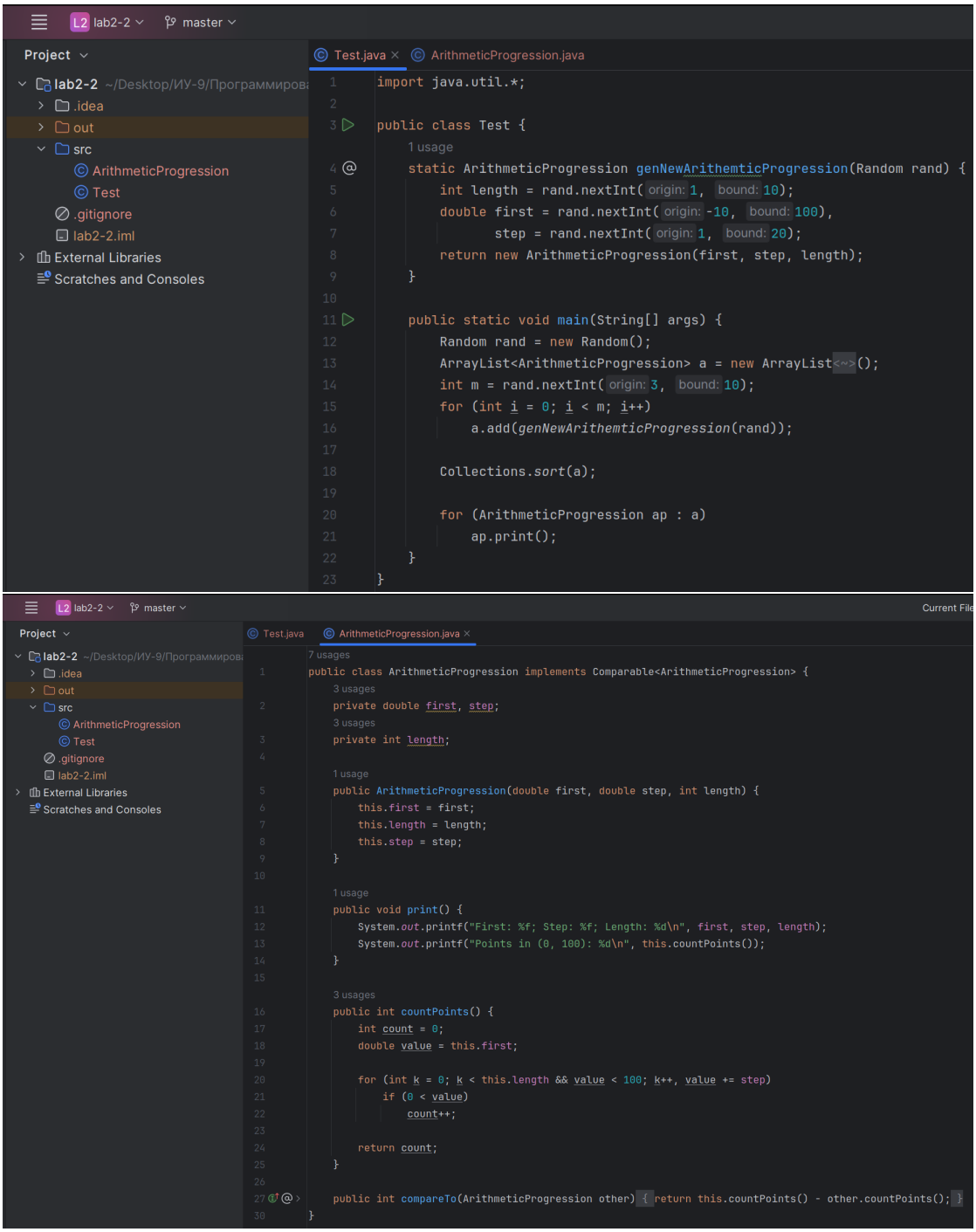

Рис. 2: Исходный код