



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 3
по курсу «Языки и методы программирования»
«Полиморфизм на основе интерфейсов в языке Java»

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Во время выполнения лабораторной работы требуется разработать на языке Java один из классов, перечисленных в таблице. В классе должен быть реализован интерфейс `Comparable<T>` и переопределён метод `toString`. В методе `main` вспомогательного класса `Test` нужно продемонстрировать работоспособность разработанного класса путём сортировки массива его экземпляров.

Класс полиномов с порядком на основе количества целочисленных корней, принадлежащих интервалу $[0; 10]$.

2 Результаты

Исходный код программы представлен в листинге 1, 2.

Листинг 1 — класс `Test`

```
1 import java.util.Arrays;
2 public class Test {
3     public static void main(String[] args) {
4         Polynomial[] polynomials = new Polynomial[5];
5         polynomials[0] = new Polynomial(new int[]{1, 0, -1});
6         polynomials[1] = new Polynomial(new int[]{1, 0, 0, -1});
7         polynomials[2] = new Polynomial(new int[]{1, 0, 1, 0, -1});
8         polynomials[3] = new Polynomial(new int[]{1, 1, 1});
9         polynomials[4] = new Polynomial(new int[]{1, -1});
10        Arrays.sort(polynomials);
11        for (Polynomial p : polynomials) {
12            System.out.println(p);
13        }
14    }
15 }
```

Листинг 2 — класс Polynomial

```
1 public class Polynomial implements Comparable<Polynomial> {
2     private int[] a_n;
3     public Polynomial(int[] a_n) {
4         this.a_n = a_n;
5     }
6     public int getNumberOfIntegerRoots() {
7         int count = 0;
8         for (int i = 0; i <= 10; i++)
9             if (evaluate(i) < 0.001)
10                count++;
11        return count;
12    }
13    public double evaluate(int x) {
14        double result = 0;
15        for (int i = 0; i < a_n.length; i++)
16            result += a_n[i] * Math.pow(x, i);
17        return result;
18    }
19    @Override
20    public int compareTo(Polynomial other) {
21        return Integer.compare(this.getNumberOfIntegerRoots(), other.
22        getNumberOfIntegerRoots());
23    }
24    @Override
25    public String toString() {
26        String result = "";
27        for (int i = a_n.length - 1; i >= 0; i--)
28            if (i == 0)
29                result += (a_n[i] > 0 && i < a_n.length-1? "+" : "") +
30                a_n[i];
31            else if (a_n[i] != 0)
32                result += (a_n[i] > 0 && i < a_n.length-1? "+" : "") +
33                a_n[i] + "*x^" + i;
34        return result;
35    }
36 }
```

Результат запуска представлен на рисунке 1, 2.

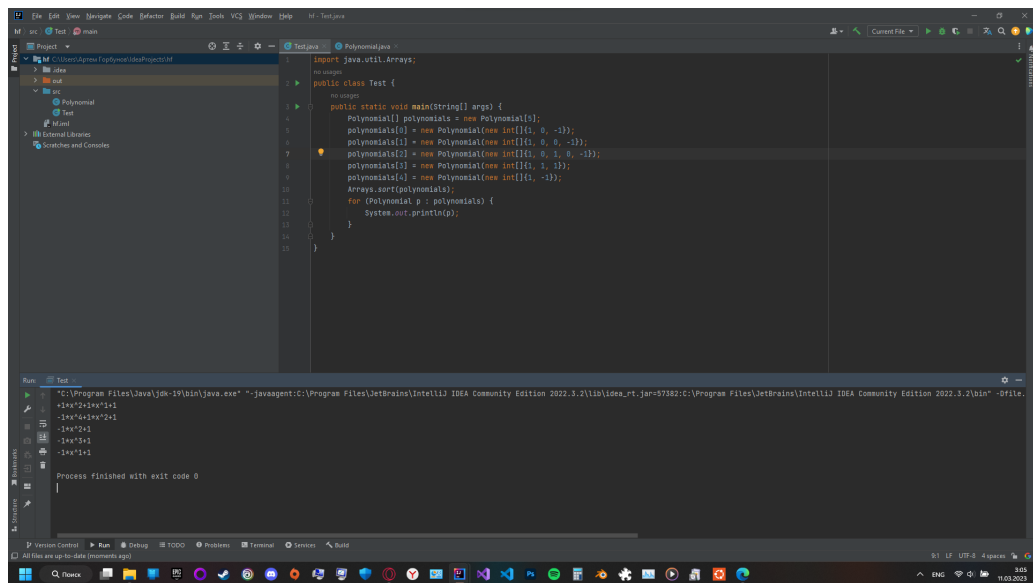


Рис. 1 — Вывод программы

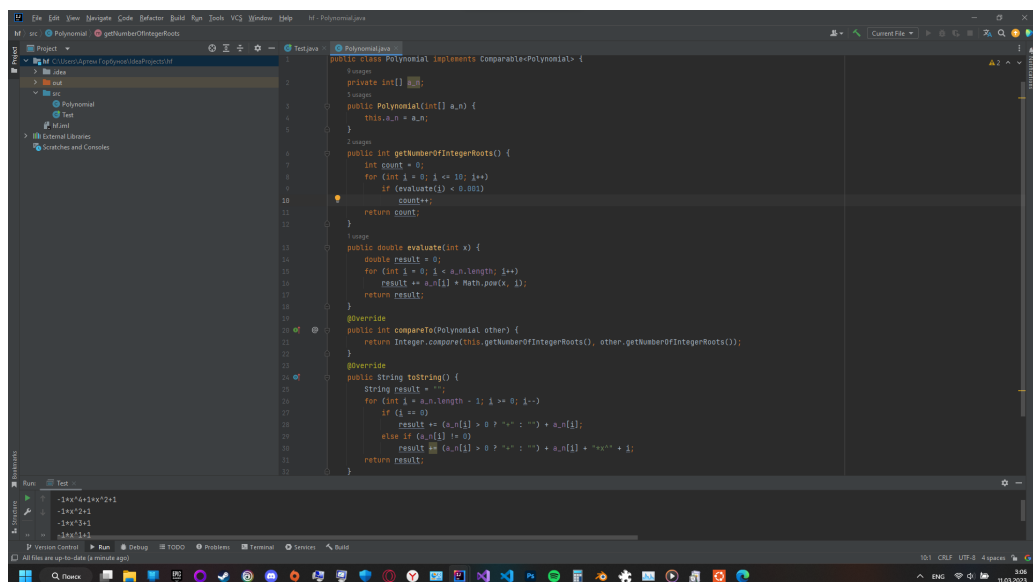


Рис. 2 — Работа программы