



**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, искусственный интеллект 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 Яннаев А. С.

Преподаватель Посевин Д. П.

*Москва 2025*

# 1 Цель работы

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

# 2 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.

Вариант №26 | Класс дробей, числитель и знаменатель которых взаимно просты, с естественным порядком на множестве рациональных чисел.

# 3 Результаты

Исходный код программы представлен в 1, 2.

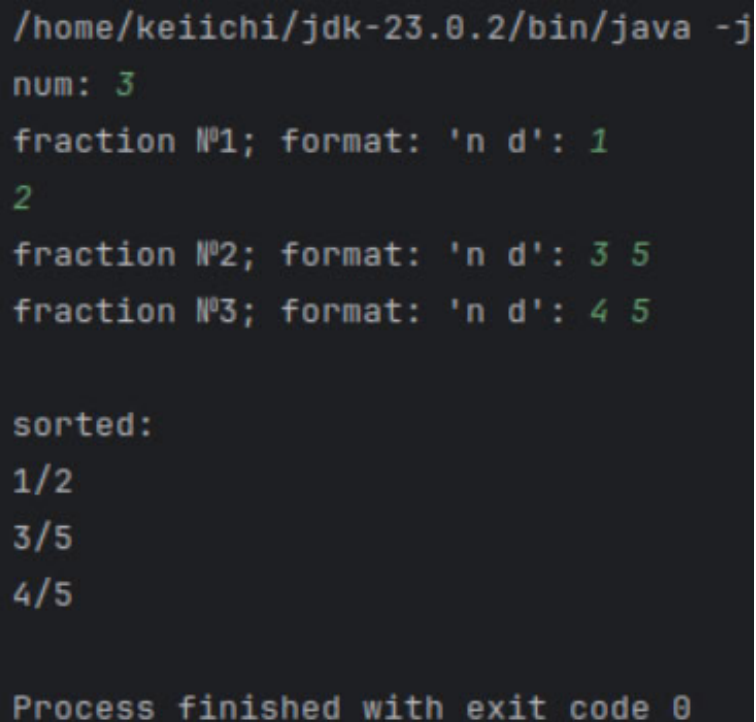
Результат запуска представлен на рисунке 1.

## Листинг 1 — Файл Rational.java

```
1 public class Rational implements Comparable<Rational> {
2     int n;
3     int d;
4
5     public Rational(int n, int d) {
6         if (d == 0) {
7             System.out.println("cant be 0");
8             System.exit(0); //
9         }
10        int gcd = findGCD(n, d);
11        this.n = n / gcd;
12        this.d = d / gcd;
13        if (this.d < 0) {
14            this.n = -this.n;
15            this.d = -this.d;
16        }
17    }
18
19    private int findGCD(int a, int b) {
20        while (b != 0) {
21            int tmp = b;
22            b = a % b;
23            a = tmp;
24        }
25        return a;
26    }
27
28    public String toString() { return n + "/" + d; }
29
30    public int compareTo(Rational other) {
31        if (this.n * other.d < other.n * this.d) {
32            return -1;
33        } else if (this.n * other.d > other.n * this.d) {
34            return 1;
35        } else {
36            return 0;
37        }
38    }
39 }
```

## Листинг 2 — Файл Test.java

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Test {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         System.out.print("num: ");
8         int count = sc.nextInt();
9
10        Rational[] fractions = new Rational[count];
11
12        for (int i = 0; i < count; i++) {
13            System.out.print("fraction " + (i + 1) + "; format: 'n d': "
14                );
15            int n = sc.nextInt();
16            int d = sc.nextInt();
17
18            fractions[i] = new Rational(n, d);
19        }
20
21        Arrays.sort(fractions);
22        System.out.println("\nsorted: ");
23        for (Rational r : fractions) {
24            System.out.println(r);
25        }
26        sc.close();
27    }
28 }
```



```
/home/keiichi/jdk-23.0.2/bin/java -j
num: 3
fraction №1; format: 'n d': 1
2
fraction №2; format: 'n d': 3 5
fraction №3; format: 'n d': 4 5

sorted:
1/2
3/5
4/5

Process finished with exit code 0
```

Рис. 1 — Результат