



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»
«Вариант 18»

Студент группы ИУ9-21Б: Пенкин А. Д.

Преподаватель: Посевин Д. П.

Москва 2023

1 Цель

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

2 Условие

Класс многоугольников с порядком на основе максимальной длины стороны многоугольника. Во время выполнения лабораторной работы требуется разработать на языке Java класс. 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.

3 Код решения

1. Figure.java

```
import static java.lang.Math.*;

public class Figure implements Comparable<Figure >{
    private int count;
    private double[] sizes = new double[10000];
    private double max;
    public Figure(double[] x){
        for (int i = 0; i < x.length; i++) {
            sizes[i] = x[i];
            if (i == 0){
                max = x[i];
            }
            else if (max < x[i]) {
                max = x[i];
            }
        }
        count = x.length;
    }

    public int compareTo(Figure o) {
        if (this.max > o.max){
            return 1;
        }
    }
}
```

```

    } else if (this.max < o.max) {
        return -1;
    } else {
        return 0;
    }
}

public String toString(){
    String res = "values:";
    for (int i = 0; i < count; i++){
        res += String.format(" %f", this.sizes[i]);
    }
    res +=String.format("; n: %d; ", this.count) + String.format("max: %f", max);
    return res;
}
}

```

2. Test.java

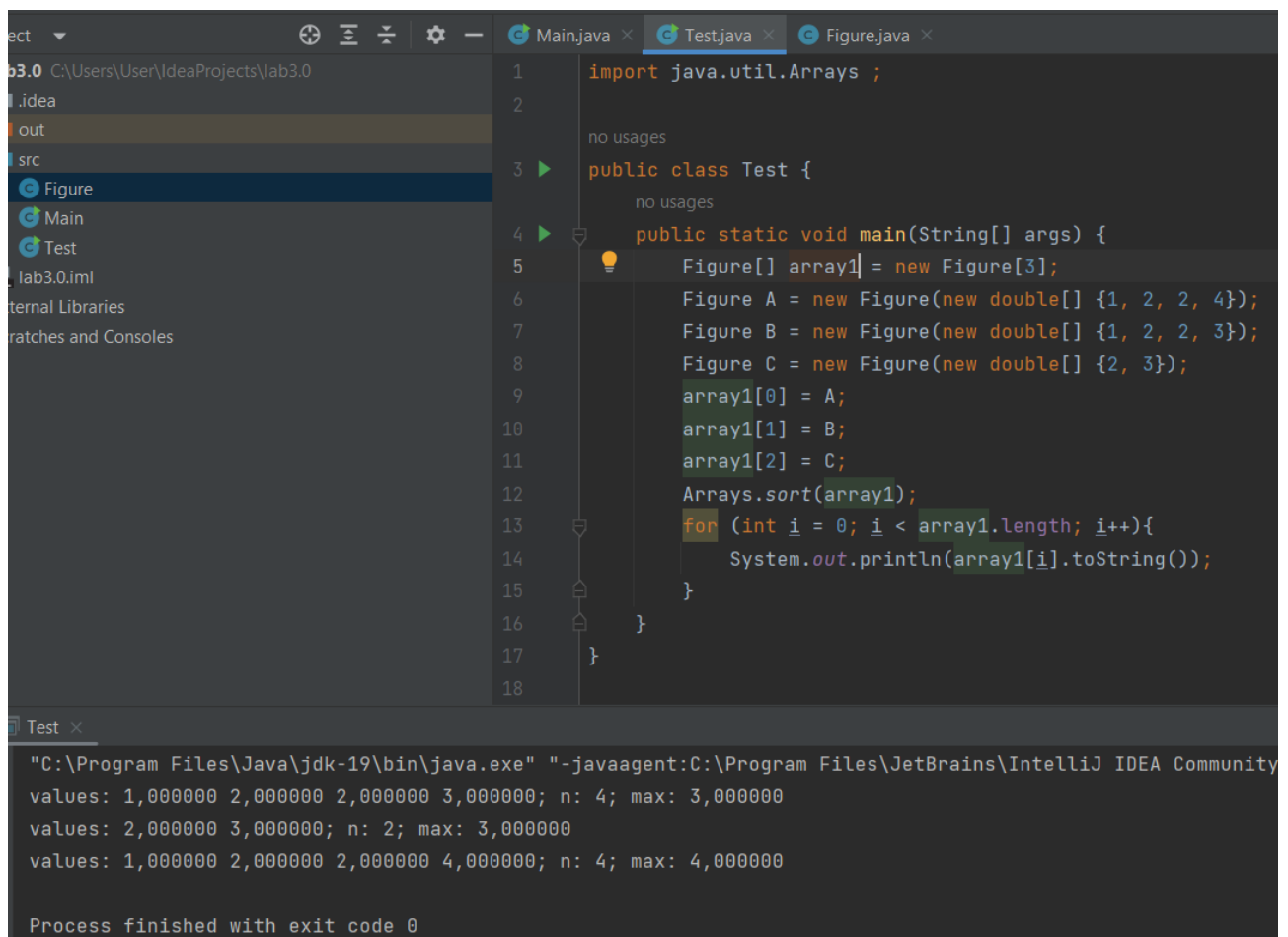
```

import java.util.Arrays ;

public class Test {
    public static void main(String[] args) {
        Figure[] array1 = new Figure[3];
        Figure A = new Figure(new double[] {1, 2, 2, 4});
        Figure B = new Figure(new double[] {1, 2, 2, 3});
        Figure C = new Figure(new double[] {2, 3});
        array1[0] = A;
        array1[1] = B;
        array1[2] = C;
        Arrays.sort(array1);
        for (int i = 0; i < array1.length; i++){
            System.out.println(array1[i].toString());
        }
    }
}

```

4 Результаты работы программы



The screenshot shows the IntelliJ IDEA IDE with three tabs: Main.java, Test.java, and Figure.java. The Test.java tab is active, displaying the following code:

```
1 import java.util.Arrays ;
2
3 public class Test {
4     public static void main(String[] args) {
5         Figure[] array1 = new Figure[3];
6         Figure A = new Figure(new double[] {1, 2, 2, 4});
7         Figure B = new Figure(new double[] {1, 2, 2, 3});
8         Figure C = new Figure(new double[] {2, 3});
9         array1[0] = A;
10        array1[1] = B;
11        array1[2] = C;
12        Arrays.sort(array1);
13        for (int i = 0; i < array1.length; i++){
14            System.out.println(array1[i].toString());
15        }
16    }
17 }
18
```

The output window at the bottom shows the following text:

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
values: 1,000000 2,000000 2,000000 3,000000; n: 4; max: 3,000000
values: 2,000000 3,000000; n: 2; max: 3,000000
values: 1,000000 2,000000 2,000000 4,000000; n: 4; max: 4,000000

Process finished with exit code 0
```

Рис. 1 — Тест программы 1

```
public Figure(double[] x){
    for (int i = 0; i < x.length; i++) {
        sizes[i] = x[i];
        if (i == 0){
            max = x[i];
        }
        else if (max < x[i]) {
            max = x[i];
        }
    }
    count = x.length;
}
```

Рис. 2 — конструктор

```

public int compareTo(Figure o) {
    if (this.max > o.max){
        return 1;
    } else if (this.max < o.max) {
        return -1;
    } else {
        return 0;
    }
}

public String toString(){
    String res = "values:";
    for (int i = 0; i < count; i++){
        res += String.format(" %f", this.sizes[i]);
    }
    res +=String.format("; n: %d; ", this.count) + String.format("max: %f", max);
    return res;
}
}

```

Рис. 3 — метод toString и compareTo