



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 15

Студент группы ИУ9-22Б Павлов И. П.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель работы

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

2 Условие

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.

3 Реализация основного класса

```
public class Main {
    public static void main(String[] args) {
        Board myBoard = new Board();
        Queen queen1 = new Queen(5, 5);
        Queen queen2 = new Queen(6, 6);
        Queen queen3 = new Queen(1, 7);

        myBoard.addQueen(queen1);
        myBoard.addQueen(queen2);
        myBoard.addQueen(queen3);

        System.out.println("Отсортированные ферзи (метод сортировки " +
            "внутри шахматной доски):");
        myBoard.getQueens();
    }
}
```

4 Реализация шахматной доски

```
import java.util.ArrayList;
import java.util.Collections;

import static java.lang.Math.abs;

public class Board {

    private ArrayList<Queen> queens = new ArrayList<>();
    private int elemCounter;

    private void countBeats() {
```

```

        for(int i = 0; i < this.elemCounter; i++) {
            Queen queen1 = this.queens.get(i);
            int x1 = queen1.getX();
            int y1 = queen1.getY();

            for (int j = i; j < this.elemCounter; j++) {
                Queen queen2 = this.queens.get(j);
                int x2 = queen2.getX();
                int y2 = queen2.getY();
                if (x1 == x2 || y1 == y2 || abs(x1 - x2) == abs(y1 - y2)) {
                    queen1.setBeatCounter(queen1.getBeatCounter() + 1);
                    queen2.setBeatCounter(queen2.getBeatCounter() + 1);
                }
            }
        }
    }

    public void addQueen(Queen entity) {
        this.queens.add(entity);
        this.elemCounter++;
    }

    public void getQueens() {
        countBeats();
        Collections.sort(this.queens);
        for(int i = 0; i < this.elemCounter; i++) {
            System.out.println(this.queens.get(i));
        }
    }
}

```

5 Реализация ферзя

```

public class Queen implements Comparable<Queen> {

    private final int x;
    private final int y;
    private int beatCounter;

    public Queen(int x, int y) {
        this.x = x;
        this.y = y;
        this.beatCounter = 0;
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public int getBeatCounter() {

```

```

        return this.beatCounter;
    }

    public void setBeatCounter(int x) {
        this.beatCounter = x;
    }

    public int compareTo(Queen entity) {
        return Integer.compare(this.beatCounter, entity.getBeatCounter());
    }

    public String toString() {
        return "Координаты ферзя: (" + this.x + ", " + this.y + "). Бьёт "
            + this.beatCounter + " ферзей";
    }
}

```

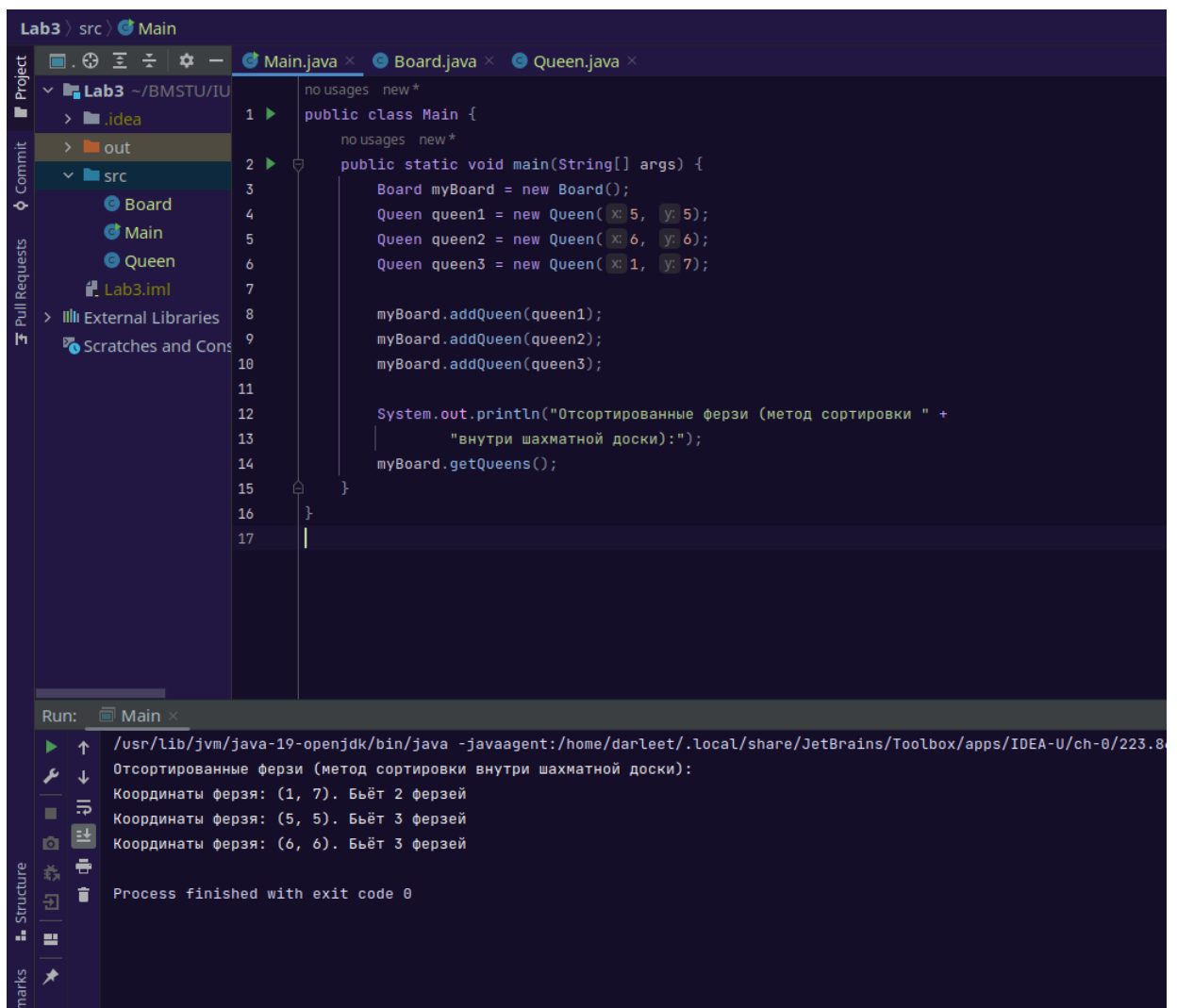


Рис. 1: Вывод программы