



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 № 4**  
**по курсу «Языки и методы программирования»**  
**«Реализация итераторов в языке Java»**

Студент группы ИУ9-22Б Лавров Р. Д.

Преподаватель Посевин Д. П.

*Москва 2025*

# 1 Задание

1. Последовательность булевских значений размера  $n$  с итератором по элементам последовательности (последовательность должна быть представлена массивом байтов, по восемь булевских значений на байт).

2. Множество отрезков на плоскости с итератором по всем точкам пересечения этих отрезков.

# 2 Результаты

Исходный код 1–3.

## Листинг 1 — Класс последовательность булевских значений

```

1 import java.util.Iterator;
2 import java.util.NoSuchElementException;
3
4 public class BooleanSequence implements Iterable<Integer> {
5     private int[] data;
6     private int length;
7
8     public BooleanSequence(boolean[] data, int n) {
9         this.length = n / 8 + (n % 8 == 0 ? 0 : 1);
10        this.data = new int[this.length];
11        for (int i = 0; i < this.length; i++) {
12            for (int j = 0; j < 8; j++) {
13                if (i*8 + j < n){
14                    this.data[i] += Math.pow(2, 7-j) * (data[i*8 + j] ? 1 : 0);
15                }
16            }
17        }
18    }
19
20    public void SetNewData(boolean[] data, int n){
21        this.length = n / 8 + (n % 8 == 0 ? 0 : 1);
22        this.data = new int[this.length];
23        for (int i = 0; i < this.length; i++) {
24            for (int j = 0; j < 8; j++) {
25                if (i*8 + j < n){
26                    this.data[i] += Math.pow(2, 7-j) * (data[i*8 + j] ? 1 : 0);
27                }
28            }
29        }
30    }
31
32    public Iterator<Integer> iterator() {
33        return new BooleanIterator();
34    }
35
36    private class BooleanIterator implements Iterator<Integer> {
37        private int currentIndex = 0;
38
39        public boolean hasNext() {
40            return currentIndex < length;
41        }
42
43        public Integer next() {
44            return data[currentIndex++];
45        }
46    }
47
48    public static void main(String[] args) {
49        boolean[] data = {true, false, true, false, true, false, true, false,
50                           true, true, true, false, false, false, false, false};
51        BooleanSequence seq = new BooleanSequence(data, data.length);
52
53        for (int b : seq) {
54            System.out.println(b);
55        }
56    }
57 }

```

## Листинг 2 — Класс множество отрезков на плоскости

```

1 import java.util.ArrayList;
2 import java.util.Iterator;
3
4 public class LineSet implements Iterable<Double[]> {
5     private ArrayList<Double[]> lines;
6     private ArrayList<Double[]> points;
7
8     public LineSet(ArrayList<Double[]> lines) {
9         this.lines = lines;
10        this.points = new ArrayList<Double[]>();
11
12        for (int i = 0; i < this.lines.size(); i++) {
13            for (int j = i + 1; j < this.lines.size(); j++) {
14                Double[] intersection = intersection(this.lines.get(i), this.lines.get(j));
15                if (intersection != null) {
16                    points.add(intersection);
17                }
18            }
19        }
20    }
21
22    public void SetLines(ArrayList<Double[]> new_lines) {
23        this.lines = new_lines;
24        for (int i = 0; i < this.lines.size(); i++) {
25            for (int j = i + 1; j < this.lines.size(); j++) {
26                Double[] intersection = intersection(this.lines.get(i), this.lines.get(j));
27                if (intersection != null) {
28                    points.add(intersection);
29                }
30            }
31        }
32    }
33
34    private Double[] intersection(Double[] a, Double[] b) {
35        double x1 = a[0];
36        double y1 = a[1];
37        double x2 = a[2];
38        double y2 = a[3];
39        double x3 = b[0];
40        double y3 = b[1];
41        double x4 = b[2];
42        double y4 = b[3];
43
44        double denom = (x1 - x2) * (y3 - y4) - (y1 - y2) * (x3 - x4);
45
46        if (denom == 0) {
47            return null;
48        }
49        double x = ((x1 * y2 - y1 * x2) * (x3 - x4) - (x1 - x2) * (x3 * y4 - y3 * x4)) / denom;
50        double y = ((x1 * y2 - y1 * x2) * (y3 - y4) - (y1 - y2) * (x3 * y4 - y3 * x4)) / denom;
51        if (isBetween(x, x1, x2) && isBetween(y, y1, y2) && isBetween(x, x3, x4) && isBetween(y, y3, y4)) {
52            return new Double[]{x, y};
53        }
54
55        return null;
56    }

```

### Листинг 3 — Класс множество отрезков на плоскости (продолжение)

```
1  private boolean isBetween(double val, double end1, double end2) {
2      return val >= Math.min(end1, end2) && val <= Math.max(end1, end2);
3  }
4
5  public Iterator<Double[]> iterator() {
6      return new LineSetIter();
7  }
8
9  private class LineSetIter implements Iterator<Double[]> {
10     private int pos;
11
12     public LineSetIter() {
13         this.pos = 0;
14     }
15
16     public boolean hasNext() {
17         return pos < points.size();
18     }
19
20     public Double[] next() {
21         Double[] result = points.get(pos);
22         pos++;
23         return result;
24     }
25 }
26
27 public static void main(String[] args) {
28     ArrayList<Double[]> lines = new ArrayList<Double[]>();
29
30     lines.add(new Double[]{2.0, 5.0, 10.0, 12.0});
31     lines.add(new Double[]{24.0, 46.0, 31.0, 21.0});
32     lines.add(new Double[]{85.0, 42.0, 13.0, 31.0});
33     lines.add(new Double[]{40.0, 74.0, 63.0, 27.0});
34
35     LineSet linesIter = new LineSet(lines);
36     for (Double[] point : linesIter) {
37         System.out.println(point[0] + ", " + point[1]);
38     }
39 }
40 }
```

```
● veter.ok77@MacBook-Pro-Rodion lab4 % cd "/Users/veter.ok77/Desktop/"  
170  
224  
○ veter.ok77@MacBook-Pro-Rodion lab4 %
```

Рис. 1 — Результат работы

```
● veter.ok77@MacBook-Pro-Rodion lab4 % cd "/Users/veter.ok77/Desktop/"  
27.57645178476292, 33.226957911561  
57.70057739895518, 37.82925488039593  
○ veter.ok77@MacBook-Pro-Rodion lab4 %
```

Рис. 2 — Результат работы

### 3 Вывод

Я научился реализовывать итераторы в языке Java