



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 № 5**  
**по курсу «Языки и методы программирования»**  
**«Монады в языке Java»**

Студент группы ИУ9-22Б Лавров Р. Д.

Преподаватель Посевин Д. П.

*Москва 2025*

# 1 Задание

Выриант 46: Последовательность символов Unicode с операциями:

1. порождение потока строк, представляющих все подпоследовательности длины  $k$ , являющиеся палиндромами;
2. поиск индекса первой буквы лексикографически наименьшего палиндрома длины  $k$ .

Проверить работу первой операции нужно путём ранжирования палиндромов их потока по количеству содержащихся в них различных букв.

Выриант 47: Множество целых чисел с операциями:

1. порождение потока таких чисел из множества, что один из делителей каждого числа входит в него в степени, не меньшей заданного числа  $k$ ;
2. поиск минимального числа  $x$  из множества такого, что в любое число, не меньшее  $x$ , любой его делитель входит в степени 1.

Проверить работу первой операции нужно путём подсчёта отрицательных, нулевых и положительных чисел из потока.

# 2 Результаты

Исходный код 1– 4.

Листинг 1 — Класс последовательность символов Unicode

```
1 import java.util.*;
2 import java.util.stream.Stream;
3
4 class NameComparator implements Comparator<String> {
5     public int compare(String a, String b) {
6         ArrayList<Character> charsA = new ArrayList<>();
7         ArrayList<Character> charsB = new ArrayList<>();
8         for (int i = 0; i < a.length(); i++) {
9             if (charsA.indexOf(a.charAt(i)) != -1) {charsA.add(a.charAt(i));}
10        }
11        for (int i = 0; i < b.length(); i++) {
12            if (charsB.indexOf(b.charAt(i)) != -1) {charsB.add(b.charAt(i));}
13        }
14        if (charsA.size() > charsB.size()) {return 1;}
15        if (charsA.size() == charsB.size()) {return 0;}
16        return -1;
17    }
18 }
```

## Листинг 2 — Класс последовательность символов Unicode (продолжение)

```
1 public class StringStream {
2     private HashMap<String, Integer> strings;
3     private int count;
4
5     public StringStream(){
6         this.strings = new HashMap<>();
7         this.count = 0;
8     }
9
10    public void add(String str){
11        this.strings.put(str, this.count);
12        this.count++;
13    }
14
15    public Stream<String> makeStream(int k) {
16        ArrayList<String> result = new ArrayList<>();
17        strings.entrySet().stream()
18            .filter(x -> x.getKey().length() == k && isPalindrom(x.getKey()))
19            .forEach(x -> result.add(x.getKey()));
20        return result.stream();
21    }
22
23    public Optional<Integer> findIndex(int k){
24        Optional<Integer> result = Optional.empty();
25        Optional<Map.Entry<String, Integer>> tmp =
26            strings.entrySet().stream()
27                .filter(x -> x.getKey().length() == k && isPalindrom(x.getKey()))
28                .findFirst();
29
30        if (tmp.isPresent()) {
31            result = Optional.ofNullable(tmp.get().getValue());
32        }
33        return result;
34    }
35
36    private boolean isPalindrom(String str){
37        for (int i = 0; i < str.length() / 2; i++){
38            if (str.charAt(i) != str.charAt(str.length() - 1 - i)){
39                return false;
40            }
41        }
42        return true;
43    }
44
45    public static void main(String[] args) {
46        StringStream strings = new StringStream();
47        strings.add("avbhgf");
48        strings.add("abbba");
49        strings.add("aabbbaa");
50        strings.add("gdhs");
51        strings.makeStream(5).sorted(new NameComparator()).forEach(System.out::println);
52        System.out.println(strings.findIndex(5).get());
53    }
54 }
```

### Листинг 3 — Класс множество целых чисел

```

1 import java.util.*;
2 import java.util.stream.*;
3
4 public class NumberStream {
5     private HashMap<Integer, Set<Integer>> numbers;
6
7     public NumberStream() {
8         this.numbers = new HashMap<>();
9     }
10
11     public void add(int number) {
12         Set<Integer> divisors = findDivisors(number);
13         this.numbers.put(number, divisors);
14     }
15     private Set<Integer> findDivisors(int number) {
16         Set<Integer> divisors = new HashSet<>();
17         if (number == 0) return divisors;
18
19         int n = Math.abs(number);
20         for (int i = 1; i <= n; i++) {
21             if (n % i == 0) {
22                 divisors.add(i);
23             }
24         }
25         return divisors;
26     }
27     public Stream<Integer> makeStream(int k) {
28         return numbers.entrySet().stream()
29             .filter (entry -> entry.getValue().stream()
30                 .anyMatch(d -> d > 1 && getExponent(entry.getKey(), d) >= k))
31             .map(Map.Entry::getKey);
32     }
33
34     private int getExponent(int number, int divisor) {
35         if (number == 0) return 0;
36         int n = Math.abs(number);
37         int exponent = 0;
38         while (n % divisor == 0) {
39             exponent++;
40             n /= divisor;
41         }
42         return exponent;
43     }
44
45     public Optional<Integer> findMinSquareFree() {
46         return numbers.keySet().stream().filter (x -> isSquareFree(x)).min(Integer::compareTo);
47     }
48
49     private boolean isSquareFree(int number) {
50         if (number == 0) return false;
51         int n = Math.abs(number);
52         for (int i = 2; i * i <= n; i++) {
53             if (n % (i * i) == 0) {
54                 return false;
55             }
56         }
57         return true;
58     }

```

#### Листинг 4 — Класс множество целых чисел (продолжение)

```
1 public static void main(String[] args) {
2     NumberStream stream = new NumberStream();
3     stream.add(100);
4     stream.add(13);
5     stream.add(40);
6     stream.add(30);
7     stream.add(49);
8     stream.add(1);
9     stream.add(0);
10    stream.add(-8);
11
12    stream.makeStream(2).sorted().forEach(System.out::println);
13    stream.findMinSquareFree().ifPresent(System.out::println);
14
15    long[] counts = stream.makeStream(2)
16        .collect(() -> new long[3],
17            (arr, num) -> {
18                if (num < 0) arr[0]++;
19                else if (num == 0) arr[1]++;
20                else arr[2]++;
21            },
22            (a, b) -> {
23                a[0] += b[0];
24                a[1] += b[1];
25                a[2] += b[2];
26            });
27    System.out.printf("%d, %d, %d\n", counts[0], counts[1], counts[2]);
28 }
29 }
```

Рис. 1 — Результат работы

```
1 error
veter.ok77@MacBook-Pro-Rodion lab5 % cd
-8
40
49
100
1
1, 0, 3
```

Рис. 2 — Результат работы

### **3 Вывод**

Я научился работать с монадами в языке Java