

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена трудового Красного Знамени федеральное государственное  
бюджетное  
образовательное учреждение высшего образования  
«Московский технический университет связи и информатики»**

Кафедра Математическая кибернетика и информационные технологии

Отчет по лабораторной работе №5.

Выполнил: студент группы БВТ2402

Безматерных Иван Алексеевич

Москва, 2025

**Цель работы:** приобрести практические навыки работы с основными интерфейсами и реализациями коллекций (Collections Framework) в Java, а также закрепить понимание принципов обобщённого программирования (Generics).

## Задания:

### Задание 1.

Написать программу, которая считывает текстовый файл и выводит на экран топ-10 самых часто встречающихся слов в этом файле. Для решения задачи использовать коллекцию Map, где ключом будет слово, а значением — количество его повторений в файле.

## Код программы:

```
1 import java.io.File;
2 import java.io.FileNotFoundException;
3 import java.util.*;
4
5 public class TopWords {
6
7     Run main | Debug main | Run | Debug
8     public static void main(String[] args) {
9         String filePath = "C:\\\\Users\\\\ivanb\\\\OneDrive\\\\Рабочий стол\\\\6-я лаба\\\\text.txt";
10
11     File file = new File(filePath);
12
13     Scanner scanner = null;
14     try {
15         scanner = new Scanner(file);
16     } catch (FileNotFoundException e) {
17         e.printStackTrace();
18         return;
19     }
20
21     Map<String, Integer> map = new HashMap<>();
22
23     while (scanner.hasNext()) {
24         String word = scanner.next().toLowerCase().replaceAll(regex: "[^a-zA-Za-zA-Z]", replacement: "");
25         if (!word.isEmpty()) {
26             map.put(word, map.getOrDefault(word, defaultValue: 0) + 1);
27         }
28     }
29
30     scanner.close();
31
32     List<Map.Entry<String, Integer>> list = new ArrayList<>(map.entrySet());
33
34     Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
35         @Override
36         public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
37             return o2.getValue().compareTo(o1.getValue());
38         }
39     });
40
41     for (int i = 0; i < Math.min(a: 10, list.size()); i++) {
42         System.out.println(list.get(i).getKey());
43     }
44 }
45 }
```

## Вывод программы:

```
or
and
univer
cat
lebron
curry
world
dog
symphony
basketball
```

## Задание 2.

Написать обобщенный класс Stack, который реализует стек на основе массива. Класс должен иметь методы push для добавления элемента в стек, pop для удаления элемента из стека и peek для получения верхнего элемента стека без его удаления.

## Код программы:

```
1  public class Stack<T> {
2      private T[] data;
3      private int size;
4
5      public Stack(int capacity) {
6          data = (T[]) new Object[capacity];
7          size = 0;
8      }
9
10     public void push(T element) {
11         if (size < data.length) {
12             data[size] = element;
13             size++;
14         }
15     }
16
17     public T pop() {
18         if (size > 0) {
19             size--;
20             T element = data[size];
21             data[size] = null;
22             return element;
23         }
24         return null;
25     }
26
27     public T peek() {
28         if (size > 0) {
29             return data[size - 1];
30         }
31         return null;
32     }
33 }
```

### **Задание 3.**

Разработать программу для учета продаж в магазине. Программа должна позволять добавлять проданные товары в коллекцию, выводить список проданных товаров, а также считать общую сумму продаж и наиболее популярный товар. Использовать TreeSet для хранения списка проданных товаров.

**Код программы:**

```

1 import java.util.*;
2
3 public class StoreSales {
4     // TreeSet для хранения списка проданных товаров
5     private TreeSet<String> soldProducts;
6     // TreeMap для подсчета количества продаж каждого товара
7     private TreeMap<String, Integer> productCount;
8     private double totalSales;
9
10    public StoreSales() {
11        soldProducts = new TreeSet<String>();
12        productCount = new TreeMap<String, Integer>();
13        totalSales = 0;
14    }
15
16    // Добавить проданный товар
17    public void addProduct(String productName, double price) {
18        // Добавляем товар в TreeSet
19        soldProducts.add(productName);
20
21        // Обновляем счетчик продаж в TreeMap
22        productCount.put(productName, productCount.getOrDefault(productName, defaultValue: 0) + 1);
23
24        // Обновляем общую сумму продаж
25        totalSales += price;
26    }
27
28    // Выводим список проданных товаров (отсортированный по алфавиту)
29    public void printSoldProducts() {
30        System.out.println("Список проданных товаров (отсортированный по алфавиту):");
31        for (String product : soldProducts) {
32            System.out.println(product + " - продано " + productCount.get(product) + " раз(а)");
33        }
34    }
35
36    // Посчитать общую сумму продаж
37    public double getTotalSales() {
38        return totalSales;
39    }
40
41    // Найти самый популярный товар
42    public String findMostPopularProduct() {
43        if (productCount.isEmpty()) {
44            return "Нет проданных товаров";
45        }
46
47        String mostPopular = null;
48        int maxCount = 0;
49
50        for (Map.Entry<String, Integer> entry : productCount.entrySet()) {
51            if (entry.getValue() > maxCount) {
52                maxCount = entry.getValue();
53                mostPopular = entry.getKey();
54            }
55        }
56
57        return mostPopular;
58    }
59
60    Run main | Debug main | Run | Debug
61    public static void main(String[] args) {
62        StoreSales store = new StoreSales();
63
64        // Добавляем проданные товары
65        store.addProduct(productName: "Хлеб", price: 50);
66        store.addProduct(productName: "Молоко", price: 80);
67        store.addProduct(productName: "Хлеб", price: 50);
68        store.addProduct(productName: "Макар.", price: 120);
69        store.addProduct(productName: "Молоко", price: 80);
70        store.addProduct(productName: "Хлеб", price: 50);
71        store.addProduct(productName: "Джем", price: 60);
72        store.addProduct(productName: "Рыбка", price: 100);
73        store.addProduct(productName: "Макар.", price: 120);
74
75        // Выводим список проданных товаров
76        store.printSoldProducts();
77
78        // Считаем общую сумму продаж
79        System.out.println("\nОбщая сумма продаж: " + store.getTotalSales() + " руб.");
80
81        // Найти самый популярный товар
82        System.out.println("Самый популярный товар: " + store.findMostPopularProduct());
83    }

```

## Вывод программы:

```
Список проданных товаров (отсортированный по алфавиту):  
Масло - продано 2 раз(а)  
Молоко - продано 2 раз(а)  
Сахар - продано 1 раз(а)  
Хлеб - продано 3 раз(а)  
Яблоки - продано 1 раз(а)  
  
Общая сумма продаж: 710.0 руб.  
Самый популярный товар: Хлеб
```

**Вывод:** в процессе выполнения лабораторной работы я приобрел практические навыки работы с основными интерфейсами и реализациями коллекций (Collections Framework) в Java, а также закрепил понимание принципов обобщённого программирования (Generics).