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 информационным технологиям и программированию

на тему: **«Работа с коллекциями»**

Выполнил: студент группы БПИ2403

ФИО: Сон Владимир Сергеевич

Руководитель: Рыбаков Егор Дмитриевич

Москва, 2025

Цель работы: Изучить систему коллекций в Java, включая основные интерфейсы (Collection, List, Set, Map) и их реализации (ArrayList, LinkedList, HashSet, HashMap и др.), освоить принципы работы с коллекциями для эффективного хранения, обработки и управления группами объектов.

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.

Задание 2.

Написать обобщенный класс Stack< T> , 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.

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.

Ход работы:

```java
import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;

public class TopWords {
    public static void main(String[] args) {
        String filePath;
        if (args.length > 0) {
            filePath = args[0];
        } else {
            filePath = "text.txt";
        }
        File file = new File(filePath);

        Scanner scanner = null;
        try {
            scanner = new Scanner(file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            return;
        }

        Map<String, Integer> wordCount = new HashMap<>();
        while (scanner.hasNext()) {
            String word = scanner.next().toLowerCase();
            word = word.replaceAll("[^а-яёa-z]", "");

            if (!word.isEmpty()) {
                wordCount.put(word, wordCount.getOrDefault(word, 0) + 1);
            }
        }
        scanner.close();

        List<Map.Entry<String, Integer>> list = new ArrayList<>(wordCount.entrySet());
        Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
            @Override
            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
                return o2.getValue().compareTo(o1.getValue());
            }
        });

        int count = 0;
        for (Map.Entry<String, Integer> entry : list) {
            if (count >= 10) break;
            System.out.println((count + 1) + ". " + entry.getKey() + " - " + entry.getValue() + " раз");
            count++;
        }
    }
}
```

```java
public class Stack<T> {
    private T[] data;
    private int size;

    @SuppressWarnings("unchecked")
    public Stack(int capacity) {
        data = (T[]) new Object[capacity];
        size = 0;
    }

    public void push(T element) {
        if (size >= data.length) {
            throw new StackOverflowError("Стек переполнен");
        }
        data[size] = element;
        size++;
    }

    public T pop() {
        if (size == 0) {
            throw new EmptyStackException("Стек пуст");
        }
        size--;
        T element = data[size];
        data[size] = null;
        return element;
    }

    public T peek() {
        if (size == 0) {
            throw new EmptyStackException("Стек пуст");
        }
        return data[size - 1];
    }

    public boolean isEmpty() {
        return size == 0;
    }

    public int getSize() {
        return size;
    }
```

```java
    public static class EmptyStackException extends RuntimeException {
        public EmptyStackException(String message) {
            super(message);
        }
    }

    public static void main(String[] args) {
        Stack<Integer> stack = new Stack<>(10);

        stack.push(1);
        stack.push(2);
        stack.push(3);

        System.out.println(stack.peek()); // 3
        System.out.println(stack.getSize()); // 3
        System.out.println(stack.pop()); // 3
        System.out.println(stack.getSize()); // 2
        System.out.println(stack.peek()); // 2

        stack.push(4);
        System.out.println(stack.pop()); // 4
        System.out.println(stack.pop()); // 2


        Stack<String> stringStack = new Stack<>(5);
        stringStack.push("Hello");
        stringStack.push("World");
        stringStack.push("Java");

        System.out.println(stringStack.pop());
        System.out.println(stringStack.pop());
        System.out.println(stringStack.peek());

        System.out.println(stringStack.isEmpty());
        stringStack.pop();
        System.out.println(stringStack.isEmpty());
    }
}
```

```java
import java.util.*;
import java.util.concurrent.CopyOnWriteArrayList;

class Product {
    private String name;
    private double price;
    private int quantitySold;

    public Product(String name, double price, int quantitySold) {
        this.name = name;
        this.price = price;
        this.quantitySold = quantitySold;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public int getQuantitySold() {
        return quantitySold;
    }

    public double getTotalRevenue() {
        return price * quantitySold;
    }

    @Override
    public String toString() {
        return String.format("%-20s | Цена: %8.2f руб | Продано: %4d шт | Выручка: %10.2f руб",
                             name, price, quantitySold, getTotalRevenue());
    }
}
```

```java
public class StoreSalesManager {
    public static void main(String[] args) {
        CopyOnWriteArrayList<Product> products = new CopyOnWriteArrayList<>();

        products.add(new Product("Молоко", 85.50, 150));
        products.add(new Product("Хлеб", 25.00, 200));
        products.add(new Product("Масло сливочное", 320.00, 80));
        products.add(new Product("Сыр", 450.00, 60));
        products.add(new Product("Яйца куриные", 120.00, 95));
        products.add(new Product("Йогурт", 75.00, 110));
        products.add(new Product("Кефир", 68.00, 130));
        products.add(new Product("Творог", 180.00, 70));

        displayProducts(products);
        calculateStatistics(products);
        demonstrateConcurrentSafety(products);
    }

    private static void displayProducts(CopyOnWriteArrayList<Product> products) {
        int counter = 1;
        for (Product p : products) {
            System.out.printf("%2d. %s%n", counter++, p);
        }
    }

    private static void calculateStatistics(CopyOnWriteArrayList<Product> products) {
        double totalRevenue = 0;
        int totalQuantity = 0;
        Product mostPopular = null;
        double maxRevenue = 0;
        Product bestRevenue = null;

        for (Product p : products) {
            double revenue = p.getTotalRevenue();
            totalRevenue += revenue;
            totalQuantity += p.getQuantitySold();

            if (mostPopular == null || p.getQuantitySold() > mostPopular.getQuantitySold()) {
                mostPopular = p;
            }

            if (revenue > maxRevenue) {
                maxRevenue = revenue;
                bestRevenue = p;
            }
        }
```

```
85          System.out.println("Общая сумма продаж:" + String.format("%,15.2f", totalRevenue) + " руб");
86          System.out.println("Всего продано единиц:" + String.format("%,15d", totalQuantity) + " шт");
87          System.out.println("Количество наименований:" + String.format("%,15d", products.size()));
88          System.out.println("Средняя выручка на товар:" +
89                                  String.format("%,15.2f", totalRevenue / products.size()) + " руб");
90
91          if (mostPopular != null) {
92              System.out.println("Самый популярный товар (по количеству):");
93              System.out.println("    " + mostPopular.getName() + " - " +
94                                      mostPopular.getQuantitySold() + " шт");
95          }
96
97          if (bestRevenue != null) {
98              System.out.println("Лидер по выручке:");
99              System.out.println("    " + bestRevenue.getName() + " - " +
100                                     String.format("%.2f", bestRevenue.getTotalRevenue()) + " руб");
101         }
102     }
103
104     private static void demonstrateConcurrentSafety(CopyOnWriteArrayList<Product> products) {
105         int count = 0;
106         for (Product p : products) {
107             count++;
108             if (count == 3) {
109                 products.add(new Product("Сметана", 95.00, 85));
110             }
111             if (count <= 4) {
112                 System.out.println("    " + count + ". " + p.getName());
113             }
114         }
115         System.out.println("\nТекущее количество товаров:" + products.size());
116     }
117 }
```

Вывод: В ходе работы были рассмотрены основы работы с коллекциями в Java. Изучены ключевые интерфейсы (Collection, List, Set, Queue, Map) и их наиболее распространённые реализации (ArrayList, HashSet, HashMap и др.), а также особенности их использования в различных сценариях. Применение коллекций позволяет эффективно организовать хранение и обработку данных, обеспечивая гибкость и производительность кода.