

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ**

Ордена Трудового Красного Знамени

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Московский технический университет связи и информатики»

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №6

Работа с коллекциями

Выполнил: Студент группы

БВТ2303

Кунецкий Владислав

Москва

2024

Цели работы:

- Изучить все основные коллекции в java
- Применить их на практике для решения различных задач.

Ход работы:

В первом задании было необходимо вывести 10 самых часто встречающихся слов из текста. В реализации этой задачи я использовал хеш-таблицу, со словом в качестве ключа и его частотой в тексте в качестве значения. После считывания слова из текста я помещаю его в хеш-таблицу или если оно уже там есть, прибавляю счетчик, отвечающий за частоту.

Из полученной хеш-таблицы нам нужно создать список, и отсортировать его по частоте. Для этого я использовал ArrayList. После сортировки остается только вывести результат

```
package lab6;

import java.io.*;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;

You, 7 hours ago | 1 author (You)
public class TopWords {
    Run | Debug
    public static void main(String[] args) {
        String filePath = "text.txt";

        File file = new File(filePath);

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

        HashMap<String, Integer> wordsTable = new HashMap<String, Integer>();

        while (scanner.hasNext()) {
            String word = scanner.next().toLowerCase().replaceAll(regex:"[а-я-ёё-ё]", replacement:"");
            if (!word.isEmpty()) {
                wordsTable.put(word, wordsTable.getOrDefault(word, defaultValue:0) + 1);
            }
        }

        scanner.close();

        List<Map.Entry<String, Integer>> list = new ArrayList<>(wordsTable.entrySet());

        You, 7 hours ago | 1 author (You)
        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());
            }
        });

        for (int i = 0; i < 11; i++) {
            System.out.println(list.get(i));
        }
    }
}
```

Скрин 1 – Программа для нахождения самый популярный слов из текста

Второе задание было на реализацию своего стека. В качестве структуры данных для хранения значений я взял массив. Метод push() помещает объект в конец массива, pop() – снимает последний элемент из массива, а peek() – просматривает последний элемент, без его удаления.

Также я реализовал служебные методы resize(), size() и isEmpty()

```
package labo;

You, 6 hours ago | 1 author (You)
public class Stack <T> {
    private T[] data;
    private int size, capacity;

    public Stack(int capacity) {
        this.data = (T[]) new Object[capacity];
        this.size = 0;
        this.capacity = capacity;
    }

    public void push(T element) {
        if (size == capacity) {
            this.resize(capacity * 2);
        }

        data[size] = element;
        this.size++;
    }

    public T pop() {
        if (size == 0) return null;
        T res = data[size - 1];
        data[size - 1] = null;
        this.size--;
        if (size > 0 && size == capacity / 4) {
            resize(capacity / 2);
        }
        return res;
    }

    public T peek() {
        if (size == 0) return null;
        return data[size - 1];
    }

    private void resize(int newCap) {
        T[] tmp = (T[]) new Object[newCap];
        for (int i = 0; i < size; i++) {
            tmp[i] = data[i];
        }
        this.data = tmp;
        this.capacity = newCap;
    }

    public int size() {
        return size;
    }

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

Скрин 2 – Реализация стека

В третьем задании необходимо 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. Для хранения списка товаров необходимо использовать класс ArrayList.

Для начала я реализовал класс представления продуктов.

```
You, 6 seconds ago | 1 author (You)
class Product {
    private String name;
    private double price;

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

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    @Override
    public String toString() {
        return "Product{name='" + name + "', price=" + price + "}";
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Product product = (Product) obj;
        return Double.compare(product.price, price) == 0 && name.equals(product.name);
    }
}
You, 8 seconds ago • Uncommitted changes
```

Скрин 3 – Класс Product

После я написал класс для учета продаж.

```

You, 1 second ago | 1 author (You)
public class SalesAccounting {
    private ArrayList<Product> salesProducts;
    private double totalSales;

    public SalesAccounting() {
        salesProducts = new ArrayList<>();
        totalSales = 0.0;
    }

    public void add(Product product) {
        salesProducts.add(product);
        totalSales += product.getPrice();
    }

    public void displayProducts() {
        if (salesProducts.isEmpty()) {
            System.out.println(x:"Нет проданных товаров.");    "проданных": Unknown word.
        } else {
            System.out.println(x:"Список проданных товаров:");    "Список": Unknown word.
            for (Product product : salesProducts) {
                System.out.println(product);
            }
        }
    }

    public double getTotalSales() {
        return totalSales;
    }

    public Product getMostPopularProduct() {
        if (salesProducts.isEmpty()) {
            return null;
        }

        Map<Product, Integer> productFrequency = new HashMap<>();
        for (Product product : salesProducts) {
            productFrequency.put(product, productFrequency.getOrDefault(product, defaultValue:0) + 1);
        }

        Product mostPopularProduct = null;
        int maxCount = 0;

        for (Map.Entry<Product, Integer> entry : productFrequency.entrySet()) {
            if (entry.getValue() > maxCount) {
                mostPopularProduct = entry.getKey();
                maxCount = entry.getValue();
            }
        }

        return mostPopularProduct;
    }
}

```

Скрин 4 – Класс для учета проданных товаров

Вывод:

В ходе выполнения лабораторной работы №6 была изучена работа с коллекциями в Java и их применение для решения различных задач. В первом задании реализован алгоритм поиска десяти самых часто встречающихся слов в тексте. Для хранения частот использовалась хеш-таблица, а для сортировки данных — список. Это позволило эффективно обработать текст и получить результаты.

Во втором задании был разработан собственный стек на основе массива. Реализация включала основные методы работы со стеком — добавление, удаление и просмотр элементов, а также служебные методы для управления памятью и проверки состояния структуры. Это позволило глубже понять внутреннюю реализацию этой структуры данных.

Третье задание заключалось в создании программы для учета продаж в магазине. Был реализован класс продукта и функциональность для добавления, подсчета общей суммы продаж и определения самого популярного товара с использованием ArrayList. Работа показала удобство и гибкость коллекций Java при решении прикладных задач. Лабораторная работа способствовала развитию навыков проектирования алгоритмов и их реализации с использованием различных структур данных.