МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ

по лабораторной работе №2

Выполнил:
студент 3 курса
группы ПО-9
Тусюк Т.В.

Проверил:
Крощенко А.А.

Брест 2024

Цель работы: приобрести базовые навыки работы с файловой системой в Java.
Задание 1:
9) Напишите программу, которая использует генерацию случайных чисел для создания предложений.
Программа должна использовать 4 массива строк, называемые noun (существительные),adjective (прилагательные), verb (глаголы) и preposition (предлоги).
Указанные массивы должны считываться из файла.
Программа должна создавать предложение, случайно выбирая слова из каждого массива в следующем порядке: noun, verb, preposition, adjective, noun.
Слова должны быть разделены пробелами.
При выводе окончательного предложения, оно должно начинаться с заглавной буквы и заканчиваться точкой. Программа должна генерировать 20 таких предложений.
Выполнение задания:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Random;

public class Task_01 {

    public static void main(String[] args) {
        String nounsFile = "nouns.txt";
        String adjectivesFile = "adjectives.txt";
        String verbsFile = "verbs.txt";
        String prepositionsFile = "prepositions.txt";

        ArrayList<String> nouns = readFile(nounsFile);
        ArrayList<String> adjectives = readFile(adjectivesFile);
        ArrayList<String> verbs = readFile(verbsFile);
        ArrayList<String> prepositions = readFile(prepositionsFile);

        if (nouns.isEmpty() || adjectives.isEmpty() || verbs.isEmpty() ||
prepositions.isEmpty()) {
            System.err.println("Ошибка: Не удалось найти необходимые данные в
файлах.");
            return;
        }

        Random random = new Random();
        for (int i = 0; i < 20; i++) {
            String sentence = generateSentence(nouns, adjectives, verbs, prepositions,
random);
            System.out.println(sentence);
        }
    }

    private static ArrayList<String> readFile(String fileName) {
        ArrayList<String> words = new ArrayList<>();
```

```
        try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                words.add(line.trim());
            }
        } catch (IOException e) {
            System.err.println("Ошибка при чтении файла " + fileName + ": " +
e.getMessage());
        }
        return words;
    }

    private static String generateSentence(ArrayList<String> nouns, ArrayList<String>
adjectives,
                                           ArrayList<String> verbs, ArrayList<String>
prepositions, Random random) {
        String noun = getRandomElement(nouns, random);
        String adjective = getRandomElement(adjectives, random);
        String verb = getRandomElement(verbs, random);
        String preposition = getRandomElement(prepositions, random);
        String secondNoun = getRandomElement(nouns, random);

        // Формирование предложения
        String sentence = noun + " " + verb + " " + preposition + " " + adjective + "
" + secondNoun + ".";
        // Преобразование первой буквы в заглавную
        sentence = sentence.substring(0, 1).toUpperCase() + sentence.substring(1);
        return sentence;
    }

    private static String getRandomElement(ArrayList<String> list, Random random) {
        int index = random.nextInt(list.size());
        return list.get(index);
    }
}
```

Результат:

```
PS F:\Java\Spp_Lab2\src> javac Task_01.java
PS F:\Java\Spp_Lab2\src> java Task_01
Дом плыл сквозь большой дерево.
Компьютер шел на красивый собака.
Кот пил рядом с красивый ручка.
Река шел за умный компьютер.
Книга смеялся над умный кот.
Собака ел в красивый книга.
Ручка плакал за быстрый телефон.
Дерево ел на ленивый дерево.
Кот бежал на уродливый дерево.
Компьютер пил внутрь умный машина.
Дом смеялся на медленный машина.
Компьютер летел в большой река.
Дом плакал рядом с ленивый дерево
Ручка прыгнул под умный собака.
Телефон спал на быстрый река.
Река плакал близко к умный дом.
Ручка ел за медленный дерево.
Телефон спал сквозь умный дом.
Компьютер пил над ленивый машина.
Компьютер спал в быстрый дерево.
```

Задание 2:
9) Утилита join объединяет строки двух упорядоченных текстовых файлов на основе наличия
общего поля. По своему функционалу схоже с оператором JOIN , используемого в языке SQL
для реляционных баз данных, но оперирует с текстовыми файлами.
Команда join принимает на входе два текстовых файла и некоторое число аргументов. Если не
передаются никакие аргументы командной строки, то данная команда ищет пары строк в двух
файлах, обладающие совпадающим первым полем (последовательностью символов, отличных
от пробела), и выводит строку, состоящую из первого поля и содержимого обоих строк.

Ключами -1 или -2 задаются номера сравниваемых полей для первого и второго файла, соот-
ветственно. Если в качестве одного из файлов указано – (но не обоих сразу!), то в этом случае

вместо файла считывается стандартный ввод.
Формат использования:
join [-1 номер_поля] [-2 номер_поля] файл1 файл2 [файл3]
Параметры:

• - 1 fiedl_num Задает номер поля в строке для первого файла, по которому будет выпол-
няться соединение.

• - 2 field_num Задает номер поля в строке для второго файла, по которому будет выпол-
няться соединение.

Аргументы:
• файл1, файл2 – входные файлы
• файл3 – выходной файл, куда записывается результат работы программы.
Примеры использования:

Пусть задан файл 1.txt со следующим содержимым:
1 abc
2 lmn
3 pqr
и файл 2.txt со следующим содержимым:
1 abc
3 lmn
9 opq
Тогда, выполнение команды join 1.txt 2.txt даст следующий результат:
1 abc abc
3 pqr lmn
Поскольку в обоих файлах есть строки, чьё первое поле совпадает (1, 3), выполнение команды
join -1 2 -2 2 1.txt 2.txt даст результат
abc 1 1
lmn 2 3

поскольку теперь сравнение выполняется по 2-му полю для первого и второго файла соответ-
ственно.

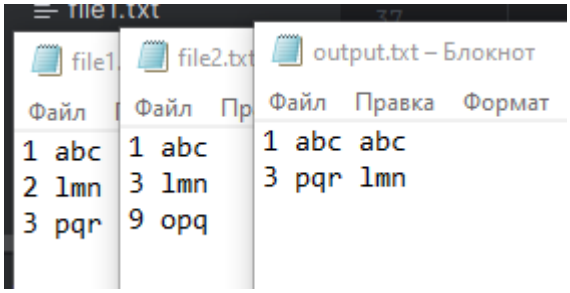
Выполнение задания:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class Task_02 {

    public static void main(String[] args) {
        if (args.length < 2) {
            System.err.println("Использование: java Task_02 файл1 файл2");
            return;
        }

        String file1 = args[0];
        String file2 = args[1];

        try (BufferedReader reader1 = new BufferedReader(new FileReader(file1));
             BufferedReader reader2 = new BufferedReader(new FileReader(file2));
             BufferedWriter writer = new BufferedWriter(new
FileWriter("output.txt"))) {

            Map<String, String> map = new HashMap<>();

            String line;
```

```
            while ((line = reader2.readLine()) != null) {
                String[] parts = line.split("\\s+", 2);
                if (parts.length >= 2) {
                    map.put(parts[0], parts[1]);
                }
            }

            while ((line = reader1.readLine()) != null) {
                String[] parts = line.split("\\s+", 2);
                if (parts.length >= 2 && map.containsKey(parts[0])) {
                    String resultLine = parts[0] + " " + parts[1] + " " +
map.get(parts[0]);
                    writer.write(resultLine);
                    writer.newLine();
                }
            }
        } catch (IOException e) {
            System.err.println("Ошибка при чтении/записи файла: " + e.getMessage());
        }
    }
}
```

Результат:

```
PS F:\Java\Spp_Lab2\src> javac Task_02.java
PS F:\Java\Spp_Lab2\src> java Task_02 file1 file2
Ошибка при чтении/записи файла: file1 (Не удается найти указанный файл)
PS F:\Java\Spp_Lab2\src> java Task_02 file1.txt file2.txt
```



Вывод: приобрел практические навыки обработки параметров командной строки, закрепил базовые знания языка программирования Java при решении практических задач.