JAVA - J2EE Batch 2

Name – Aman Yadav

E-mail : prakashaman5@gmail.com

Phone: +919519131321

Assignment-7

2 PROBLEM STATEMENT – SET CREATION AND SORTING

In this problem, a aet of unique words should be returned from the input paragraph, based on the input order.

Implement the changes inside UniqueWordExtractor and necessary files if you want to write.

```
    * Expected Order of Unique words is decided by an Enum containing
following constants
      ALPHABETIC_ASCENDING    - unique words in ascending order
      ALPHABETIC_DESCENDING   - unique words in descending order
      LENGTH_THEN_ALPHABETIC_ASCENDING - unique words in ascending order of
    word length and then alphabetically ascending
      INPUT_ORDER             - unique words in the same order as appearing
    in the paragraph
```

→

cartCheckOut.java | *UniqueWordExtractor.java

```java
package com.main;

import java.util.*;

enum Order {
    ALPHABETIC_ASCENDING, ALPHABETIC_DESCENDING, LENGTH_THEN_ALPHABETIC_ASCENDING, INPUT_ORDER
}

public class UniqueWordExtractor {

    public static List<String> extractUniqueWords(String paragraph, Order order) {
        if (paragraph == null || paragraph.isEmpty()) {
            return Collections.emptyList();
        }

        String[] words = paragraph.trim().split("\\s+");

        Set<String> uniqueWords = new LinkedHashSet<>(Arrays.asList(words));

        List<String> uniqueWordsList = new ArrayList<>(uniqueWords);

        switch (order) {
        case ALPHABETIC_ASCENDING:
            Collections.sort(uniqueWordsList);
            break;
        case ALPHABETIC_DESCENDING:
            Collections.sort(uniqueWordsList, Collections.reverseOrder());
            break;
        case LENGTH_THEN_ALPHABETIC_ASCENDING:
            Collections.sort(uniqueWordsList,
                    Comparator.comparingInt(String::length).thenComparing(Comparator.naturalOrder()));
            break;
        case INPUT_ORDER:
            break;
        default:
            throw new IllegalArgumentException("Unknown order type");
        }
```

```
38
39
40          return uniqueWordsList;
41      }
42
43      public static void main(String[] args) {
44          String paragraph = "This is a sample paragraph with some sample words.";
45          System.out.println("Alphabetic Ascending Order: " + extractUniqueWords(paragraph, Order.ALPHABETIC_ASCENDING));
46          System.out
47                  .println("Alphabetic Descending Order: " + extractUniqueWords(paragraph, Order.ALPHABETIC_DESCENDING));
48          System.out.println("Length then Alphabetic Ascending Order: "
49                  + extractUniqueWords(paragraph, Order.LENGTH_THEN_ALPHABETIC_ASCENDING));
50          System.out.println("Input Order: " + extractUniqueWords(paragraph, Order.INPUT_ORDER));
51      }
52  }
53
```

Console ×

<terminated> UniqueWordExtractor [Java Application] D:\my\eclipse-jee-2024-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.

```
Alphabetic Ascending Order: [This, a, is, paragraph, sample, some, with, words.]
Alphabetic Descending Order: [words., with, some, sample, paragraph, is, a, This]
Length then Alphabetic Ascending Order: [a, is, This, some, with, sample, words., paragraph]
Input Order: [This, is, a, sample, paragraph, with, some, words.]
```