Ex.No.6

String Handling

23.09.2020

Mugilan E.S.

2019202033

1.

Displaying a Sentence with Its Words Reversed. Write an application that inputs a line of text, tokenizes the line with String method split and outputs the tokens in reverse order. Use space characters as delimiters.

ReverseString.java

```
package lab.six;
import java.util.Arrays;
import java.util.Collections;
import java.util.Scanner;
public class ReverseString {
	public static void reversePrintString(String str) {
		System.out.println("\nOriginal String: " + str);
		String[] tokens = str.split(" ");
		Collections.reverse(Arrays.asList(tokens));
		StringBuilder reversed = new StringBuilder();
		for (String string : tokens) {
			reversed.append(string).append(" ");
		}
		System.out.println("\nReversed String: " + reversed);
	}
	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		System.out.println("Print String in Reverse Order");
		System.out.print("Enter the String: ");
		String str = in.nextLine();
		reversePrintString(str);
	}
}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
Print String in Reverse Order
Enter the String: This is a reverse String

Original String: This is a reverse String

Reversed String: String reverse a is This

Process finished with exit code 0
```

2.

Displaying Strings in Uppercase and Lowercase. Write an application that inputs a line of text and outputs the text twice—once in all uppercase letters and once in all lowercase letters.

PrintCaseString.java

```
package lab.six;

import java.util.Scanner;
public class PrintCaseString {
	private static void printChangedCase(String str) {
		System.out.println("\nGiven String: " + str);
		System.out.println("Uppercase String: " + str.toUpperCase());
		System.out.println("Lowercase String: " + str.toLowerCase());
	}
	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		System.out.println("Print Given string in Upper and Lower Cases");
		System.out.print("Enter a string: ");
		String str = in.nextLine();
		printChangedCase(str);
	}
}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Print Given string in Upper and Lower Cases
Enter a string: Mugilan is ALWAYS awesome.

Given String: Mugilan is ALWAYS awesome.
Uppercase String: MUGILAN IS ALWAYS AWESOME.
Lowercase String: mugilan is always awesome.

Process finished with exit code 0
```

3.

(Searching Strings) Write an application that inputs a line of text and a search character and uses String method indexOf to determine the number of occurrences of the character in the text.

SearchCount.java

```
ppackage lab.six;
import java.util.Scanner;
public class SearchCount {
	private static boolean isEmpty(String str) {
		return str == null || str.length() == 0;
	}
```

```
    private static int countOccurrences(String search, String str) {
        if (isEmpty(search) || isEmpty(str)) {
            return 0;
        }
        int index = 0, count = 0;
        while (true) {
            index = str.indexOf(search, index);
            if (index == -1) {
                break;
            }
            count++;
            index += search.length();
        }
        return count;
    }
    public static void main(String[] args) {
        System.out.println("Searching and Counting Occurrences");
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a String: ");
        String bigString = in.nextLine();
        System.out.print("Enter the Search Character: ");
        String searchString = in.nextLine();
        int count = countOccurrences(searchString, bigString);
        System.out.println("Number of Occurrences = " + count);
    }
}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
Searching and Counting Occurrences
Enter a String: This is a string about mugilan
Enter the Search Character: a
Number of Occurrences = 3

Process finished with exit code 0
```

4.

Creating Three-Letter Strings from a Five-Letter Word. Write an application that reads a five-letter word from the user and produces every possible three-letter string that can be derived from the letters of that word. For example, the three-letter words produced from the word “bathe” include “ate,” “bat,” “bet,” “tab,” “hat,” “the” and “tea.”

ThreeLetter.java

```
package lab.six;

import java.util.Scanner;
```

```java
public class ThreeLetter {
	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		System.out.print("Enter the 5 letter word: ");
		String str = in.nextLine();
		for(int i =0; i<str.length();i++) {
			for(int j = 0; j< str.length();j++) {
				for (int k =0; k < str.length();k++) {
					if (i == j || j == k || k == i) continue;
					System.out.println(str.charAt(i) + "" + str.charAt(j) + "" + str.charAt(k));
				}
			}
		}
	}
}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/
Enter the 5 letter word: bathe
bat
bah
bae
bta
bth
bte
bha
bht
bhe
bea
bet
beh
abt
abh
abe
atb
ath
ate
ahb
aht
ahe
aeb
aet
aeh
tba
tbh
tbe
tab
tah
tae
```

```
thb
tha
the
teb
tea
teh
hba
hbt
hbe
hab
hat
hae
htb
hta
hte
heb
hea
het
eba
ebt
ebh
eab
eat
eah
etb
eta
eth
ehb
eha
eht

Process finished with exit code 0
```

5.

Printing Dates in Various Formats. Dates are printed in several common formats. Two of the more common formats are 04/25/1955 and April 25, 1955 Write an application that reads a date in the first format and prints it in the second format.

DateFormat.java

```java
package lab.six;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;
public class DateFormat {
	public static void main(String[] args) throws ParseException {
		Scanner in = new Scanner(System.in);
		System.out.print("Enter the Date (MM/DD/YYYY): ");
		String dateString = in.nextLine();
		Date dateFormat = new SimpleDateFormat("MM/dd/yyyy").parse(dateString);
		String printString = new SimpleDateFormat("MMMM dd, yyyy").format(dateFormat);
		System.out.println(printString);
	}
}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Enter the Date (MM/DD/YYYY): 04/23/1999
April 23, 1999

Process finished with exit code 0
```

6.

Replacing Substrings and Splitting Strings Sometimes it's useful to replace parts of a string or to split a string into pieces. For this purpose, class String provides methods replaceAll, replaceFirst and split. These methods are demonstrated in the following lines.

Get two sentences as input from user

String First sentence = "This sentence $$$$$$ ends in 5 stars "

Replace **this à the**

Replace **$ à &**

Replace **stars à** Chocolates

String Second sentence = "1, 2, 3, 4, A,B,C,D";

(first two numbers 1,2) à digit

A,…D à Alphabet

StringReplace.java

```
package lab.six;

import java.util.Scanner;
public class StringReplace {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter First Sentence: ");
        String str1 = in.nextLine();
        String newStr = str1.replaceAll("This", "the");
        newStr = newStr.replace("$", "&");
        newStr = newStr.replaceAll("stars", "Chocolates");
        System.out.println("First sentence after Replacing : \n" + newStr);
        System.out.print("\nEnter Second Sentence: ");
        String str2 = in.nextLine();
        newStr = str2.replaceAll("[12]", "digit");
        newStr = newStr.replaceAll("[A-D]", "Alphabet");
        System.out.println("Second sentence after Replacing : \n" + newStr);
    }
}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Enter First Sentence: This sentence $$$$$ ends with 5 stars
First sentence after Replacing :
the sentence &&&&& ends with 5 Chocolates

Enter Second Sentence: 1,2,3,4,A,B,C,D
Second sentence after Replacing :
digit,digit,3,4,Alphabet,Alphabet,Alphabet,Alphabet

Process finished with exit code 0
```

7.

Metric Conversions. Write an application that will assist the user with metric conversions. Your application should allow the user to specify the names of the units as strings (i.e., centimeters, liters, grams, and so on, for the metric system and inches, quarts, pounds, and so on, for the English system) and should respond to simple questions, such as

"How many inches are in 2 meters?"

"How many liters are in 10 quarts?"

Your application should recognize invalid conversions. For example, the question "How many feet are in 5 kilograms?" is not meaningful because "feet" is a unit of length, whereas "kilograms" is a unit of mass.

```
package lab.six;

import java.util.Scanner;
class ConVolume {
    public double litreToQuart(double l, String unit) {
        double q;
        if (unit.equalsIgnoreCase("liter")) {
            q = l / 1.137f;
        } else {
            q = l / 1137;
        }
        return q;
    }
    public double quartzToLitre(double q) {
        return q * 1.137f;
    }
}
class ConLength extends ConVolume {
    public double meterToInch(double meter) {
        return meter * 39.37;
    }
    public double inchToMeter(double inch) {
        return inch / 39.37;
```

```
    }
}
class ConMass extends ConLength {
    public double gramToPound(double gram) {
        return gram / 454;
    }
    public double poundToGram(double pound) {
        return pound * 454;
    }
}
class Conversion extends ConMass {
    public Scanner scanObj = new Scanner(System.in);
    public String units, outUnit;
    double value;
    void run() {
        inputGet();
    }
    void inputGet() {
        System.out.print("Enter the Input for Metric Conversion\nEnter the " +
                "value: ");
        value = scanObj.nextDouble();
        scanObj.nextLine();
        System.out.print("Unit for Input: ");
        units = scanObj.nextLine();
        System.out.print("Unit for Output: ");
        outUnit = scanObj.nextLine();
        process();
    }
    void process() {
        if (units.contains("meter")) {
            if (outUnit.equalsIgnoreCase("inch")) {
                if (units.equalsIgnoreCase("centimeter")) {
                    System.out.println("Conversion of " + value + " " + units + " = " +
meterToInch(value / 100) + " inch");
                } else {
                    System.out.println("Conversion of " + value + " " + units + " = " +
meterToInch(value) + " inch");
                }
            } else {
                System.out.println(units + " Cannot convert into " + outUnit);
            }
        } else if (units.contains("inch")) {
            if (outUnit.equalsIgnoreCase("meter")) {
                if (outUnit.equalsIgnoreCase("centimeter")) {
```

```
                        System.out.println("Conversion of " + value + " " + units + " = " +
(inchToMeter(value) * 100) + " centimeter");
                    } else {
                        System.out.println("Conversion of " + value + " " + units + " = " +
inchToMeter(value) + " meter");
                    }
                } else {
                    System.out.println(units + " Cannot convert into " + outUnit);
                }
            } else if (units.contains("liter")) {
                if (outUnit.equalsIgnoreCase("quart")) {
                    System.out.println("Conversion of " + value + " " + units +
                            " = " + litreToQuart(value, units) + " quart");
                } else {
                    System.out.println(units + " Cannot convert into " + outUnit);
                }
            } else if (units.contains("quart")) {
                if (outUnit.equalsIgnoreCase("liter")) {
                    System.out.println("Conversion of " + value + " " + units +
                            " = " + quartzToLitre(value) + " liter");
                } else {
                    System.out.println(units + " Cannot convert into " + outUnit);
                }
            } else if (units.contains("gram")) {
                if (outUnit.equalsIgnoreCase("pound")) {
                    System.out.println("Conversion of " + value + " " + units +
                            " = " + gramToPound(value) + " pound");
                } else {
                    System.out.println(units + " Cannot convert into " + outUnit);
                }
            } else if (units.contains("pound")) {
                if (outUnit.equalsIgnoreCase("gram")) {
                    System.out.println("Conversion of " + value + " " + units +
                            " = " + poundToGram(value) + " gram");
                } else {
                    System.out.println(units + " Cannot convert into " + outUnit);
                }
            } else {
                System.out.println(units + " Cannot convert into " + outUnit);
            }
        }
}
public class MetricConversion {
    public static void main(String[] args) {
```

```
            new Conversion().run();
        }
}
```

Output:

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Enter the Input for Metric Conversion
Enter the value: 2
Unit for Input: meter
Unit for Output: inch
Conversion of 2.0 meter = 78.74 inch

Process finished with exit code 0
```

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Enter the Input for Metric Conversion
Enter the value: 10
Unit for Input: quart
Unit for Output: liter
Conversion of 10.0 quart = 11.369999647140503 liter

Process finished with exit code 0
```

```
/Library/Java/JavaVirtualMachines/openjdk-14.0.2.jdk/Contents/Home/bin/java -javaagent:/Users/mugilan-codes/Library/Application Support/J
Enter the Input for Metric Conversion
Enter the value: 5
Unit for Input: gram
Unit for Output: feet
gram Cannot convert into feet

Process finished with exit code 0
```

Source Code:

https://github.com/Mugilan-Codes/java-lab-exercises