

Day 4 – Assignment

Pratik K Kamble
Employee ID : 46263548
13 – 09 – 2022

Q1: Write a program to get the firstname, lastname and year of birth as input and return the generated password. Follow the below table for more understanding.

Class Name	StringUtil
Method	public String generatePassword(String firstName,String lastName, String yob): This should return the generated password.
Sample Input	PanKaj SHaRma 1983
Output	ankhar83
Note	User may give input in mixed case, your generated output must be in lowercase only
Class Name	StringUtilTest
Method	main(String[] args): For testing the StringUtil method

Solution –

```
package com.Assignment_day4;

import java.util.Scanner;

2 usages
public class StringUtil {
    1 usage
    public String generatePassword(String firstName,String lastName,
                                   String yob) {
        return firstName.substring(1,4).toLowerCase()
            + lastName.substring(1,4).toLowerCase()
            + yob.substring( beginIndex: 2).toLowerCase();
    }
}

class StringUtilTest {
    public static void main(String[] args) {
        StringUtil str = new StringUtil();
        Scanner scan = new Scanner(System.in);
        String firstName = scan.next();
        String lastName = scan.next();
        String yob = scan.next();
        scan.close();

        String generatedpsw = str.generatePassword(firstName, lastName, yob);
        System.out.println("Generated Password is: " + generatedpsw);
    }
}
```

D:\soCKET\Capgemini\java\bin\java.exe "-javaagent:D:\Joining Capgemini\P
PanKaj
SHaRma
1983
Generated Password is: ankhar83
Process finished with exit code 0

Q2: Create a method in the StringUtil class as, getCommand(). This method will receive on url and will return the command from this url. Follow below table for more understanding.

Class Name	StringUtil
Method	public String getCommand(String url): This method will receive the url and return the command associated with that url.
Sample Input	www.sample.com/admin/createProject.html
Output	createProject
Note	Check with multiple and different type of urls
Class Name	StringUtilTest
Method	main(String[] args): For testing the StringUtil method

Solution –

```

package com.Assignment_day4;

import java.util.Scanner;

2 usages
public class StringUtil2 {
    1 usage
    public String getCommand(String url) {
        String[] strofarr = url.split("/");
        String last = strofarr[strofarr.length - 1];
        String[] cmder = last.split("\\.");
        return cmder[0];
    }
}

class StringUtilTest2 {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String url = scan.next();
        scan.close();

        StringUtil2 strutil = new StringUtil2();

        String command = strutil.getCommand(url);

        System.out.println("Command is: " + command);
    }
}

```

Console Output:

```

D:\socket\Capgemin1\java\bin\java.exe "-javaagent:D:\Joining
www.sample.com/admin/createProject.html
Command is: createProject
Process finished with exit code 0

```

Q3: You need to work on the StringExercises class, that will have some set of static methods. You need to implement these methods as per the description. For testing different operations of the StringExercises class. Create a main class and accordingly call the methods with appropriate parameters.

Class Name	StringExercises
public static int findMarc(String string) {return -2; }	Searches for "Marc" in a string. string is a non null String return the index of the first occurrence of the "Marc" in the string. Otherwise -1 if not found
public static int findSubstring(String string, String substring) {return -2;}	Searches for substring within a string string is a non-null string substring a non-null string return the index of the first occurrence of the substring within the string, or -1 if not found
public static boolean contains(String string, String substring) {return false;}	Returns true if and only if the string contains the substring. string is a non-null string substring is a non-null string return true if and only if the string contains the substring
public static String[] splitIntoWords(String string) { return null;}	Splits a string into words, using whitespace to delimit the words. you can use the String.split method as follows to get an array of the words: String[] words = s.split("\\s+"); string a non-null string return an array representing the words in the string.
public static String firstFour(String string) { return null;}	Returns the substring representing the first four characters of the string. string a non-null string of length>=4 return the substring representing the first four characters of the string.
public static String firstN(String string, int n) { return null;}	Returns the substring representing the first n characters of the string. string a non-null string of length>=n n an integer>=0 return the substring representing the first n characters of the string
public static String lastFour(String string) { return null;}	Returns the substring representing the last four characters of the string. string a non-null string of the length >=4 return the substring representing the last four characters of the string

Solution –

```
import java.lang.*;
import java.util.Arrays;

public class StringExercises {

    1 usage
    public static int findMarc(String string) {
        if (string.contains("Marc")){
            return string.indexOf("Marc");
        }
        return -1;
    }

    1 usage
    public static int findSubstring(String string, String substring) {
        if (string.contains(substring)){
            return string.indexOf(substring);
        }
        return -1;
    }

    1 usage
    public static boolean contains(String string, String substring) {
        if (string.contains(substring)){
            return true;
        }
        return false;
    }

    1 usage
    public static String[] splitIntoWords(String string) {
        String[] words = string.split( regex: "\\s+");
        return words;
    }
}
```

```
public static String firstFour(String string) {
    String sub4 = string.substring(0, 4);

    return sub4;
}

1 usage
public static String firstN(String string, int n) {
    String subN = string.substring(0, n);

    return subN;
}

1 usage
public static String lastFour(String string) {
    if (string.length() >= 4) {
        String last4 = string.substring( beginIndex: string.length() - 4);
        return last4;
    }
    return "null";
}

public static void main(String[] args) {
    System.out.println(findMarc( string: "You are not Marc"));
    System.out.println(findSubstring( string: "You are not Marc", substring: "are"));
    System.out.println(contains( string: "You are not Marc", substring: "are"));
    System.out.println(Arrays.toString(splitIntoWords( string: "You are not Marc")));
    System.out.println(firstFour( string: "Youare not Marc"));
    System.out.println(firstN( string: "YouarenotMarc", in: 8));
    System.out.println(lastFour( string: "You are not Marc"));
}
}
```

12
4
true
[You, are, not, Marc]
Youa
Youareno
Marc
Process finished with exit code 0

Q4: Create a class named as MatchPatternUtil, and you will have to create few methods in this. Follow the below table for method description. For testing the MatchPatternUtil class, create a main class and test functionality with appropriate input.

Class Name	MatchPatternUtil
public static int[] matchabSequence(String string){ return null;}	Return the array of int, which holds the index number of matched pattern. string is non-null string input: abaaaba output: 0 4
public static int[] findDigitLocation(String string){ return null;}	Return the array of int, which holds the index number of matched pattern. string is non-null string input: a12c3e456f output: 1 2 4 6 7 8
public static int[] findWhiteSpaces(String string){ return null;}	Return the array of int, which holds the index number of matched pattern. string is non-null string input: a 1 56 _Z output: 1 3 6
public static int[] findPattern(String string){ return null;}	Return the array of int, which holds the index number of matched pattern. string is non-null string input: 12 0x 0x12 0Xf 0Xg output: 6 11 What to do : Find a set of characters in which the first character is a “0”, the second character is either an “x” of an “X”, and the third character is either a digit from “0” to “9”, a letter from “a” to “f” or an upper case letter “A” to “F.”

Solution –

```
import org.jetbrains.annotations.NotNull;

import java.util.Arrays;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class MatchPatternUtil {
    1 usage
    public static int @NotNull [] matchabSequence(String string) {
        Pattern pattern = Pattern.compile( regex: "ab");
        Matcher match = pattern.matcher(string);

        int size = string.split( regex: "ab").length - 1;
        int[] seq = new int[size];
        int i = 0;
        while (match.find()) {
            seq[i] = match.start();
            i++;
        }
        return seq;
    }

    1 usage
    public static int @NotNull [] findDigitLocation(String string) {
        Pattern pat = Pattern.compile( regex: "\\d");
        Matcher mat = pat.matcher(string);
        int size = string.split( regex: "\\d").length - 1;
        int[] seq = new int[size];
        int i = 0;
        while (mat.find()) {
            seq[i] = mat.start();
            i++;
        }
    }
}
```

D:\socket\Capgemini\java\bin\java.exe "-javaagent:D:\Joining Capg
Indexes of ab are: [0, 4]
Indexes of digits are: [1, 2, 4, 6, 7, 8]
Indexes of whitespaces are: [1, 3, 6, 9]
Indexes of patterns are: [6, 11]
Process finished with exit code 0

```
}
    return seq;
}

1 usage
public static int @NotNull [] findWhiteSpaces(String string) {
    Pattern pat = Pattern.compile( regex: " ");
    Matcher mat = pat.matcher(string);
    int size = string.split( regex: " ").length;
    int[] seq = new int[size];
    int i = 0;
    while (mat.find()) {
        seq[i] = mat.start();
        i++;
    }
    return seq;
}

1 usage
public static int @NotNull [] findPattern(String string) {
    Pattern pattern = Pattern.compile( regex: "[xx][\\da-fA-F]");
    Matcher match = pattern.matcher(string);

    int size = string.split( regex: "[xx][\\da-fA-F]").length - 1;
    int[] seq = new int[size];
    int i = 0;
    while (match.find()) {
        seq[i] = match.start();
        i++;
    }

    return seq;
}
```

D:\socket\Capgemini\java\bin\java.exe "-javaagent:D:\Joining Capg
Indexes of ab are: [0, 4]
Indexes of digits are: [1, 2, 4, 6, 7, 8]
Indexes of whitespaces are: [1, 3, 6, 9]
Indexes of patterns are: [6, 11]
Process finished with exit code 0

```
public static void main(String[] args) {
    int[] matchpattern = matchabSequence( string: "abaaaba");
    System.out.println("Indexes of ab are: " + Arrays.toString(matchpattern));

    int[] digitlocation = findDigitLocation( string: "a12c3e456f");
    System.out.println("Indexes of digits are: " + Arrays.toString(digitlocation));

    int[] whiteSpaces = findWhiteSpaces( string: "a 1 56 _Z ");
    System.out.println("Indexes of whitespaces are: " + Arrays.toString(whiteSpaces));

    int[] pattern = findPattern( string: "12 0x 0x12 0Xf 0Xg ");
    System.out.println("Indexes of patterns are: " + Arrays.toString(pattern));
}
}
```

D:\socket\Capgemini\java\bin\java.exe "-javaagent:D:\Joining Capg
Indexes of ab are: [0, 4]
Indexes of digits are: [1, 2, 4, 6, 7, 8]
Indexes of whitespaces are: [1, 3, 6, 9]
Indexes of patterns are: [6, 11]
Process finished with exit code 0