Module 2
Java Development

Assignment 10- MADP202- July 30, 2019

Due: Thursday July 30, 6:30 pm

Submission: Please create a pull-request to the corresponding git branch.

Late submission Penalty Policy:

- Lack of submission of 3 or more assignments: 50% of your entire assignments (7.5% of your course mark).
- Lack of submission of 5 or more assignments: 100% of your entire assignments (15% of your course mark).
- Remember you need to get at least 70% to pass the course.

Topics:

Functional Interfaces (SAM = Single Abstract Method)

Problem 1: Built-in Functional Interfaces

This is the link of Java documents for Functional interfaces:
https://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html

This is the link to examples of using Functional interfaces
http://www.java2s.com/Tutorials/Java_Lambda/java_util_function_Reference.htm

In this assignment you are going to come up with one example for each of the functional interfaces. To this end, for each build-in functional interface, come up with a problem and write a 1,2 line problem description that can be solved using Functional interfaces.

For example:

Interface BiFunction<T,U,R>:

Using BiFunction, write a method which takes two array lists of integer and append the second array list to the first one and return a new arraylist.

```
1.  import java.util.ArrayList;
2.  import java.util.Arrays;
3.  import java.util.function.BiFunction;
4.
5.  public class BiFunctionExample {
6.
7.      public static void main(String[] args) {
8.          // TODO Auto-generated method stub
9.
10.         BiFunction<ArrayList<Integer>, ArrayList<Integer>,ArrayList<Integer>> bi =
11.                 (list1, list2) -> {
12.
13.                     ArrayList<Integer> list3 = new ArrayList<>();
14.                     for(Integer i : list1)
15.                     {
16.                         list3.add(i);
17.                     }
18.
19.                     for(Integer i : list2)
20.                     {
21.                         list3.add(i);
```

```
                    }

                return list3;
            };

        ArrayList<Integer> l1 = new ArrayList<>(Arrays.asList(1, 2,3,4,5,6));
        ArrayList<Integer> l2 = new ArrayList<>(Arrays.asList(7, 8,9,10,11,12));
        System.out.println(bi.apply(l1, l2).size());

    }

}
```

You are supposed to do this for every single Functional Interface (43 examples)

Problem 2: Custom Functional Interface

Create a functional interface yourself and use an example to show how it works. You don't need to define any default method for your functional interface.