420-202-RE Data Structures and Object-oriented Programming

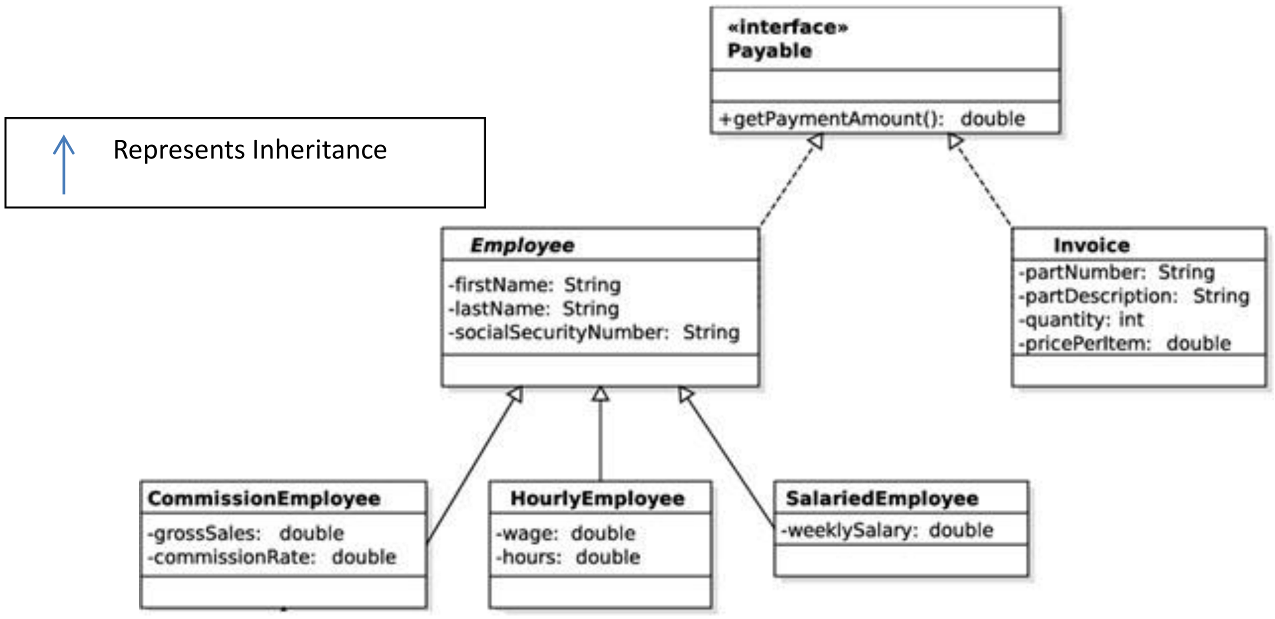
Lab Task 2

Sakkaravarthi Ramanathan

Q1) Create an Arraylist with random numbers, then remove all the odd numbers and sort the Arraylist

Input: 3, 11, 26, 12, 99, 345, 36, 5, 18
Output: 12, 18, 26, 36

Q2) Create the appropriate interface classes, objects, subclass for the following diagram and print the output.



Q3) Writing a program that displays the number of times/occurrences of each element repeated in an arraylist

Input : [3, 4,5,5,6,7,3,7]
Output:
3 repeated 2 times
4 repeated 1 times
5 repeated 2 times
6 repeated 1 times
7 repeated 2 times

Q4) create a Student class with attributes id (int), name (String), and score (double), implementing the Comparable interface for sorting based on the id. Additionally, develop a separate ScoreComparator class that implements the Comparator interface, allowing comparison of Student objects based on their scores. In the main program, instantiate Student objects, populate an array or list, and showcase sorting using both natural ordering (via Comparable) and ordering based on scores (via ScoreComparator). Utilize the Collections.sort() method to demonstrate the effectiveness of the implemented interfaces.

Q5) The main idea is to understand how the catch works for different type of exception: Rearrange the code and modify the value of integers d, n, array g to catch the exceptions.

```
class JavaException {
       public static void main(String args[]){
        try {
             int d =0;
             int n = 20;


             int fraction = n / d;

             int g[] = new int[1];
             g[0] = 100;
             int value = g[1];

             String str = null;
             int strLength = str.length();
        } catch (ArithmeticException e) {
             System.out.println("In the catch block due to ArithmeticException = " + e);
        } catch (Exception e) {
             System.out.println("In the catch block due to ArrayIndexOutOfBoundsException = " +
e);
        }
           // there are 4 missing lines in the code
        System.out.println("End Of Main");
    }
 }
```

Q6) The code to find the roots of quadratic equation is given here. Now modify the code by including the appropriate exception handling mechanism for guiding the user so that user will not enter illegal inputs for the program to work. The illegal input could be string, char, special characters. Every exception needs to be bundled with catch statement and include a finally statement for printing the output.

```
public class QuadraticEquationExample1 {
       public static void main(String[] Strings) {
              Scanner input = new Scanner(System.in);
              System.out.print("Enter the value of a: ");
              double a = input.nextDouble();
              System.out.print("Enter the value of b: ");
              double b = input.nextDouble();
              System.out.print("Enter the value of c: ");
              double c = input.nextDouble();
              double d = b * b - 4.0 * a * c;
              if (d > 0.0) {
                     double r1 = (-b + Math.pow(d, 0.5)) / (2.0 * a);
                     double r2 = (-b - Math.pow(d, 0.5)) / (2.0 * a);
                     System.out.println("The roots are " + r1 + " and " + r2);
              } else if (d == 0.0) {
                     double r1 = -b / (2.0 * a);
                     System.out.println("The root is " + r1);
              } else {
                     System.out.println("Roots are not real.");
              }
       }
}
```

Important Note : After completing the code in the IDE, please copy the screenshot of your output into a Word file along with your Java source code and submit it on assignment link. Failure to do so will result in a deduction of grades.