COMP-2120 - Fall 2020

Lab 7 Activities

Activity 1.

Java interfaces are used for common behaviours/services to hierarchically unrelated classes. One of the common services is paying, that could be used in various classes.
First, develop an interface, `Payable`, in which there is a method `getPaymentAmount()` that returns a double amount that must be paid for an object of any class that implements the interface.
Then, develop two unrelated classes, `Employee`, and `Invoice`, with some standard instance variables and methods, and both classes implement `Payable` interface. In `Employee` class, payment is calculated by multiplying `workingHours` by `hourlyRate`, and in `Invoice`, payment is calculated by multiplying `numberOfproduct` by `unitPrice` and then adding tax.

Activity 2.

One of the common interfaces of the Java API is `Comparable`, which has a method, `compareTo`. Implement this interface in the Shape class that we had in the previous lab such that it can compare the areas of two shapes and send back the proper result.

Activity 3.

Human being can eat different things. For instance, we eat fruits, chicken, fish, vegetables, etc. First create a Java Interface, called `Edible`, which has one service method, called `howToEat()`. Then, create an abstract class, called `Animal`, and couple of subclasses, such as `Chicken`, `meat`, `Fish`, etc. Then for each of these subclasses that are edible, implement the `Edible` interface and therefore complete the implementation of the service method `howToEat()`. Also, create a class `Fruit`, and some subclasses for it, like `Orange`, `Apple`, `Banana`, etc. and implement the `Edible` interface for them. As an example, for instance the following method can be written for `Chicken` class:

```
public class Chicken extends Animal implements Edible {

        . . .

        public String howToEat( ) {

                return "Chicken: Fry it and enjoy eating with ketchup, fries and Coke…!";
        }

        . . .

}
```

Activity 4.

You have already developed Shape class along with some of its subclasses such as Circle, Oval, Triangle and Square. Now, extend your development by implementing the Comparable interface in Shape class. This means you should implement the compareTo method inside this class based on the areas of two shapes. Then you are able to compare two shapes. Write a tester class to test your new development.