Object Oriented Programming - Lab 3

Date : 01-09-2021

Problem 1

Write a Java program to implement a Banking Application. The application should create and manipulate 'Account' objects (Use array of objects). The following operations should be available:

a. Account creation
b. Account Balance checking
c. Credit / Debit operations
d. Transfer between two accounts
e. Account Login

This should be done before performing the operations b-d. Login should be based on account number, password.

All accounts should have the following attributes:

a. Account number
b. Account name
c. Password (Use plain text for this assignment)
d. Balance
e. Number of credit, debit, and transfer operations across all accounts.

Note:

- Don't use unnecessary print statements in your program. For all the banking operations, use the account number for identification of accounts. Use proper accessor and mutator methods for the fields.
- Use proper scope for all members of any classes.
- Don't take any input from the user. Initialize the array of objects, statically in your program. Hardcode any necessary inputs.

What to turn in for problem 1

1. BankApplication.java which will contain the following classes
   a. Class Account
   b. Class BankApplication
2. A Screenshot (jpg/png) of program execution with sample output.

Problem 2:

Create three java classes namely, **point, rectangle, and circle** that has the following properties:

1. The class point has the members coordinateX and coordinateY.
   a. Define a method called 'print' to print the details: coordinateX and coordinateY.
2. The class rectangle has the members pointOne, pointTwo (a pair of diagonally opposite corners).
   a. Define methods getHeight and getWidth.
   b. Define a method getAreaRectangle to calculate the area of a rectangle from the given points. (it should use the getHeight and getWidth functions)
   c. Define a method isSmaller(rectangle), which takes as argument another rectangle object and returns 'true' if the other rectangle is smaller.
   d. Define a method called 'print' to print all details: the points, the height, the width and the area (Note: for printing points you should appropriately call the print method of the point class)
3. The class circle has the members center-point and radius
   a. Define a method getAreaCircle to calculate the area of a circle from its radius (Tips: You may use 'Math.PI' constant as the value of pi and Math.sqrt() function to compute the square of a number)
   b. Define a method isSmaller(circle), which takes as argument another circle object and returns 'true' if the other circle is smaller.
   c. Define a method called 'print' to print all details: the points, the radius and the area (Note: for points you should appropriately call the print method of the point class)
4. Cross comparison:
   a. In the rectangle class, implement a method isSmallerCircle(Circle) which takes as argument a circle and returns true if the area of the circle is smaller than the rectangle.
   b. In the circle class, implement a method isSmallerRectangle(rectangle) which takes as argument a rectangle and returns true if the area of the rectangle is smaller than the circle.

Write all of the above in a file called ShapeComaprisonMain.java for writing the main method. In main, generate your rectangles and circles, as many as you need, to test all of the 'isSmaller' functions.Remember to test all functions with both positive and negative examples.

What to turn in for problem 2

3. ShapeComparisonMain.java which will contain the following classes
   a. Class point
   b. Class rectangle
   c. Class circle
   d. Public class ShapeComparisonMain
4. In the output, first print details of the two objects that you are comparing and then print the result of the comparison. Take a screenshot of running the program and attach the JPG/PNG file(s). (Note: There are four 'isSmaller' methods. So, even if you do just one positive example and one negative example, there will be **eight such comparisons**. One screenshot may not capture all eight. So, it's ok to split into two screenshots, if necessary.)

5. **DO NOT ZIP**. submit the files directly.