CP 215 Object Oriented Programming in Java

PRACTICAL_5

5.1 The objective of this practical is to emphasis on methods, object, variables and constructors.

1. Write a java app that contains a version of class *Account* that maintains an instance variables *name* and *balance* of a bank account. A typical bank services *many* accounts, each with its own balance. Every instance (i.e., object) of class Account contains its own copies of both the *name* and the *balance*. Provide constructor to initialice instance variables. Provide method setName, getName, deposite and getBalance to set name, get name, deposite money to balance and get balance respectively. Then provide another class called AccountTest and create two objects to test the Account class.

2. Modify class Account (in question 1) to provide a method called withdraw that withdraws money from an Account. Ensure that the withdrawal amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the method should print a message indicating "Withdrawal amount exceeded account balance." Modify class *AccountTest* to test method withdraw.

3. Write a java program that declares two classes—*Employee* and *EmployeeTest*. Class Employee declares private static variable *coun*t and public static method *getCount*. The static variable count maintains a count of the number of objects of class Employee that have been created so far. This class variable has to be initialized to zero. If a static variable is not initialized, the compiler assigns it a default value—in this case 0, the default value for type int. EmployeeTest method main instantiates two Employee objects to test the Employee class.

4. Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide methods that calculate the rectangle's perimeter and area. It has set and get methods for both length and width. The set methods should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0. Write a program to test class Rectangle.

5. Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12—this interest should be added to savings-Balance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest for each of 12 months and print the new balances for both savers. Next, set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.