Assignment 5

16 September 2024 13:33

//Design and implement a class named InstanceCounter to track and count the number of instances created from this class.

```
package org.example;

public class Que1 {
   private static int count = 0;

   public Que1() {
      count++;
   }

   public static int getCount() {
      return count;
   }

   public static void main(String[] args) {
         Que1 obj1 = new Que1();
         Que1 obj2 = new Que1();
      Que1 obj3 = new Que1();

      System.out.println("Number of instances created: " + Que1.getCount());
   }
}
```

```
Number of instances created: 3
```

//Design and implement a class named Logger to manage logging messages for an application. The class should be implemented as a singleton to ensure that only one instance of the Logger exists throughout the application.
//
//The class should include the following methods:
//getInstance(): Returns the unique instance of the Logger class.
//log(String message): Adds a log message to the logger.
//getLog(): Returns the current log messages as a String.
//clearLog(): Clears all log messages

```
package org.example;

public class Que2 {
   private static Que2 instance;
   private StringBuilder logMessages;

   private Que2() {
      logMessages = new StringBuilder();
   }

   public static Que2 getInstance() {
      if (instance == null) {
         instance = new Que2();
      }
      return instance;
   }

   public void log(String message) {
      logMessages.append(message).append("\n");
```

```
    }

    public String getLog() {
        return logMessages.toString();
    }

    public void clearLog() {
        logMessages.setLength(0);
    }

    public static void main(String[] args) {
            Que2 logger = Que2.getInstance();
        logger.log("First log message.");
        logger.log("Second log message.");
        System.out.println("Current log messages:");
        System.out.println(logger.getLog());

        logger.clearLog();
        System.out.println("Log after clearing:");
        System.out.println(logger.getLog());
    }
}
```

```
Current log messages:
First log message.
Second log message.

Log after clearing:
```

```
//Design and implement a class named Employee to manage employee data for a company. The class should include fields to keep track of the total number of employees and the total salary expense, as well as individual employee details such as their ID, name, and salary.
//The class should have methods to:
//Retrieve the total number of employees (getTotalEmployees())
//Apply a percentage raise to the salary of all employees (applyRaise(double percentage))
//Calculate the total salary expense, including any raises (calculateTotalSalaryExpense())
//Update the salary of an individual employee (updateSalary(double newSalary))

package org.example;

import java.util.ArrayList;
import java.util.Scanner;

public class Que3 {
    private static int totalEmployees = 0;
    private static double totalSalaryExpense = 0.0;

    private int id;
    private String name;
    private double salary;

    public Que3(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
        totalEmployees++;
        totalSalaryExpense += salary;
    }

    public static int getTotalEmployees() {
        return totalEmployees;
    }
```

```
public static void applyRaise(double percentage) {
   totalSalaryExpense = 0;
   for (Que3 emp : employees) {
      emp.salary += emp.salary * percentage / 100;
      totalSalaryExpense += emp.salary;
   }
}

public static double calculateTotalSalaryExpense() {
   return totalSalaryExpense;
}

public void updateSalary(double newSalary) {
   totalSalaryExpense -= this.salary;
   this.salary = newSalary;
   totalSalaryExpense += newSalary;
}

@Override
public String toString() {
   return "Employee ID: " + id + ", Name: " + name + ", Salary: " + salary;
}

private static ArrayList<Que3> employees = new ArrayList<>();

public static void main(String[] args) {
   Scanner scanner = new Scanner(System.in);

   while (true) {
      System.out.println("1. Add Employee");
      System.out.println("2. Apply Raise");
      System.out.println("3. Calculate Total Salary Expense");
      System.out.println("4. Update Employee Salary");
      System.out.println("5. Display Total Employees");
      System.out.println("6. Exit");
      System.out.print("Choose an option: ");
      int choice = scanner.nextInt();

      if (choice == 6) {
         break;
      }

      switch (choice) {
         case 1:
            System.out.print("Enter Employee ID: ");
            int id = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            System.out.print("Enter Employee Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter Employee Salary: ");
            double salary = scanner.nextDouble();
            employees.add(new Que3(id, name, salary));
            break;
         case 2:
            System.out.print("Enter raise percentage: ");
            double percentage = scanner.nextDouble();
            applyRaise(percentage);
            break;
         case 3:
            System.out.println("Total Salary Expense: " + calculateTotalSalaryExpense());
            break;
         case 4:
            System.out.print("Enter Employee ID to update salary: ");
            int updateId = scanner.nextInt();
```

```
            System.out.print("Enter new salary: ");
            double newSalary = scanner.nextDouble();
            for (Que3 emp : employees) {
              if (emp.id == updateId) {
                emp.updateSalary(newSalary);
                break;
              }
            }
            break;
          case 5:
            System.out.println("Total Employees: " + getTotalEmployees());
            break;
          default:
            System.out.println("Invalid option.");
        }
      }
    }
}
```

```
1. Add Employee
2. Apply Raise
3. Calculate Total Salary Expense
4. Update Employee Salary
5. Display Total Employees
6. Exit
Choose an option: 1
Enter Employee ID: 23
Enter Employee Name: abcf
Enter Employee Salary: 234567
1. Add Employee
2. Apply Raise
3. Calculate Total Salary Expense
4. Update Employee Salary
5. Display Total Employees
6. Exit
Choose an option:
```