Assignement No. 6

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

**package** org.example.instance;

**public** **class** InstanceCounter {

**private** **static** **int** *instanceCount*=0;

**public** InstanceCounter() {

*instanceCount*++;

}

**public** **static** **int** getInstanceCount() {

**return** *instanceCount*;

}

**public** **static** **void** setInstanceCount(**int** instanceCount) {

InstanceCounter.*instanceCount* = instanceCount;

}

**public** **static** **void** main(String args[]) {

InstanceCounter n1 = **new** InstanceCounter();

InstanceCounter n2 = **new** InstanceCounter();

InstanceCounter n3 = **new** InstanceCounter();

System.***out***.println("Number of Instances : "+InstanceCounter.*getInstanceCount*());

}

}



2) 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.logger;

**public** **class** Logger {

**private** **static** Logger *instance* = **null**;

**private** StringBuilder logMessages;

**private** Logger() {

logMessages = **new** StringBuilder();

}

**public** **static** Logger getInstance() {

**if** (*instance* == **null**) {

**synchronized** (Logger.**class**) {

**if** (*instance* == **null**) {

*instance* = **new** Logger();

}

}

}

**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) {

Logger logger = Logger.*getInstance*();

logger.log("Application started.");

logger.log("User logged in.");

logger.log("Error: Invalid input.");

System.***out***.println("Current Log:");

System.***out***.println(logger.getLog());

logger.clearLog();

System.***out***.println("Log after clearing:");

System.***out***.println(logger.getLog());

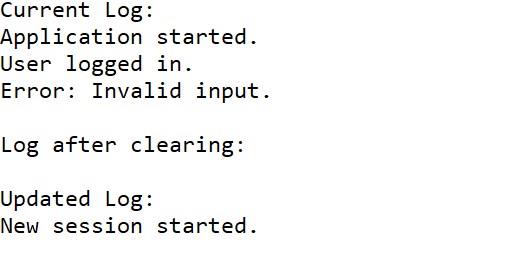
logger.log("New session started.");

System.***out***.println("Updated Log:");

System.***out***.println(logger.getLog());

}

}



3) 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))

Understand the problem statement and use static and non-static fields and methods appropriately. Implement static and non-static initializers, constructors, getter and setter methods, and a toString() method to handle the initialization and representation of employee data.

Write a menu-driven program in the main method to test the functionalities.

package org.example.demo4;

import java.util.ArrayList;

import java.util.Scanner;

public class Program {

public static int totalEmployees=0;

public static double totalSalaryExpense=0.0;

private int id;

private String name;

private double salary;

static {

totalEmployees=0;

totalSalaryExpense=0.0;

}**public** Program(**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** **double** calculateTotalSalaryExpense() {

**return** *totalSalaryExpense*;

}

**public** **static** **void** applyRaise(**double** percentage, ArrayList<Program> employees) {

**for**(Program emp : employees) {

**double** raise = emp.salary \* (percentage / 100);

emp.salary += raise;

*totalSalaryExpense* += raise;

}

}

**public** **void** updateSalary(**double** newSalary) {

*totalSalaryExpense* -= **this**.salary;

**this**.salary = newSalary;

*totalSalaryExpense* += newSalary;

}

**public** **int** getid() {

**return** id;

}

**public** String getname() {

**return** name;

}

**public** **double** getsalary() {

**return** salary;

}

**public** String toString() {

**return** "Employee ID : "+id+ " Employee name : "+name+" Employee Salary : "+salary;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

ArrayList<Program> employees = **new** ArrayList<>();

**while**(**true**) {

System.***out***.println("Menu :");

System.***out***.println("1.Add new Employee :");

System.***out***.println("2.Display All Employees :");

System.***out***.println("3.Apply Salary raise :");

System.***out***.println("4.Update Employee Salary :");

System.***out***.println("5.Show Total Employee :");

System.***out***.println("6.Show Total Salary Expense :");

System.***out***.println("7.Exit");

System.***out***.println("Choose an option :");

**int** choice = sc.nextInt();

**switch**(choice) {

**case** 1:

System.***out***.print("Enter Employee ID: ");

**int** id = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Employee Name: ");

String name = sc.nextLine();

System.***out***.print("Enter Employee Salary: ");

**double** salary = sc.nextDouble();

employees.add(**new** Program(id, name, salary));

System.***out***.println("Employee added successfully!");

**break**;

**case** 2:

System.***out***.println("\nEmployees:");

**for** (Program emp : employees) {

System.***out***.println(emp);

}

**break**;

**case** 3:

System.***out***.print("Enter percentage raise: ");

**double** percentage = sc.nextDouble();

Program.*applyRaise*(percentage, employees);

System.***out***.println("Salary raise applied to all employees!");

**break**;

**case** 4:

System.***out***.print("Enter Employee ID to update salary: ");

**int** empId = sc.nextInt();

Program empToUpdate = **null**;

**for** (Program emp : employees) {

**if** (emp.getid() == empId) {

empToUpdate = emp;

**break**;

}

}

**if** (empToUpdate != **null**) {

System.***out***.print("Enter new salary: ");

**double** newSalary = sc.nextDouble();

empToUpdate.updateSalary(newSalary);

System.***out***.println("Salary updated successfully!");

} **else** {

System.***out***.println("Employee not found.");

}

break;

**case** 5: // Show Total Employees

System.***out***.println("Total Employees: " + Program.*getTotalEmployees*());

**break**;

**case** 6: // Show Total Salary Expense

System.***out***.println("Total Salary Expense: $" + Program.*calculateTotalSalaryExpense*());

**break**;

**case** 7: // Exit

System.***out***.println("Exiting the program...");

sc.close();

System.*exit*(0);

**default**:

System.***out***.println("Invalid option, please try again.");

}

}

}

}

