CS101- Algorithms and Programming I

Lab 08

Lab Objectives: `Classes and Objects`

- For all labs in CS 101, your solutions must conform to the CS101 style guidelines (rules!)

1. Create a Java class, `Project`, that has the following functionality:

   Constant Data Members:
   - `TAX`: tax rate to be applied to the project. Fixed at 13%.
   - `OVERHEAD`: the project overhead, fixed at 50000TL
   - `EMP_HOURS_WEEK`: standard number of hours each employee works per week (45)

   Static data members:

   - `projectCounter`: counts the number of projects created. Initialized to 5000.

   Instance Data Members:

   - `projectId`: stores the unique id number of the project (example: 2021-5000)
   - `projectName`: stores the name of the project.
   - `projectType`: stores the character representing the project type.
   - `personHours`: stores the estimated person hours for the project.
   - `ratePerHour`: stores the standard rate per person hour.
   - `projectWeeks`: stores the estimated duration of the project in weeks.

   Methods:

   - **Constructor**:
     - Counts the project.
     - Takes `projectName, personHours, ratePerHour, projectWeeks` as parameters.
     - Initializes the `projectName` to the value passed as a parameter.
     - Initializes the `projectId ,projectType, personHours, ratePerHour, projectWeeks` using the set methods.

   - **Accessor methods for the following data members:**
     - `projectName, projectId ,projectType, personHours, ratePerHour, projectWeeks.`

   - **Mutator methods:**
     - `setProjectName`: sets the `projectName` to the value passed as a parameter.
     - `setPersonHours`: sets the `personHours` to the value passed as a parameter if it is a positive value, otherwise sets to zero.
     - `setRatePerHour`: sets the `ratePerHour` to the value passed as a parameter if it is positive, otherwise sets to zero.
     - `setProjectWeeks`: sets the `projectWeeks` to the value passed as a parameter if it is positive, otherwise sets to zero.
     - `setProjectId`: private method that sets the `projectId` using the current year plus the `projectCounter`. (ex:2021-5010 if 10 projects have been created)

    - `setProjectType`: sets the type according to the calculated project cost. If the `projectCost` is over 1000000 it is a (m)ajor project, between 500000 and 1000000 it is a (l)arge-scale, between 0 and 500000, (s)tandard project. Projects with a project cost of 0 are (i)nactive.

- **Service Methods:**
    - `deactivateProject()`: sets the project to (i)nactive. Set `personHours, ratePerHour` to zero.
    - `activateProject()`: sets the `personHours, ratePerHour` to values passed as parameters. Updates projectType according to the calculated project cost.
    - `calculateProjectCost()`: calculates and returns the cost of the project. Cost is the sum of the human resource cost (personHours multiplied by ratePerHour), and the project overhead, increased by the tax rate. Note: if the cost before overhead is less than 20000, the `OVERHEAD` is not added.
    - `calculatePersonResources()`: calculates and returns the number of employees required for the project, using the `personHours, projectWeeks`. Assume each employee is contracted for `EMP_HOURS_WEEK`.
    - `toString()`: returns a String representation of a project. See sample output for format details. Active and inactive projects will display differently as shown below.

2. Create a Tester Application for your Project class which does the following:
    - Create 2 Projects.
    - Display the Projects.
    - Update the projects:
        - Change their `personWeeks` and `ratePerHour`.
        - Update the `projectType`.
        - Deactivate one of the projects.
        - Display the updated projects.

Sample Output:

```
Project Name: ArcTech Business Solution
Project ID: 2023-5001
Project Type: m
Team Size: 5
Estimated Project Cost: 2855284.0

Project Name: SunMarkets POS Implementation
Project ID: 2023-5002
Project Type: s
Team Size: 1
Estimated Project Cost: 170404.0


-------INACTIVE PROJECT------
Project Name: HealthTech Hospital
Project ID: 2023-5003
```

3. Create a class `Department`:
    - **Instance Data Members:**
        - `deptName`: stores the name of the department.
        - `deptCode`: stores the code of the department.

    - **Methods:**
        - **Constructor:**
            - Initializes the department name and department code using the ones passed as parameters.
        - **Accessor methods for:**
            - `deptName, deptCode.`
        - **Other methods:**
            - `equals()`: instance method that takes an Object as a parameter, and returns true if the target department and the Department passed as a parameter are the same, false if not.
            - `toString()`: returns a String representation of a department. See sample output for format details.

4. Create a class `Employee:`
    - Constant data member:
        - WORKING_DAYS: there are 261 working days per year.
    - **Instance Data Members:**
        - `employeeName`: stores the name of the employee.
        - `dailyRate`: stores the double daily pay rate of the employee.
        - `department`: stores the Department of the employee.
        - `project`: stores the project the employee has been assigned to.
    - **Methods:**
        - **Constructor:**
            - Initializes the employee name, rate and project using the ones passed as parameters. Also takes the department name and code as parameters, initializes a new department using the ones passed as a parameter.
        - **Constructor:**
            - Copy constructor: creates a new `Employee` object using the data from the `Employee` passed as a parameter. The new employee will be assigned to the same project.
        - **Accessor / Mutator methods for:**
            - `employeeName, dailyRate, department, project.`
        - **Other methods:**
            - `calculateYearlySalary()`: Calculates and returns yearly salary.
            - `toString()`: returns a String representation of an Employee. See sample output for format details.

b. Create and application, EmployeeApp that does the following:
    - Create a Project.
    - Create 3 Employees who are assigned to the project.
    - Create a new Employee that is a copy of the first.
    - Display the 4 Employees.
    - Compare the Department of all Employees, and display Employees with matching Departments.

Sample Output:

```
Employees:

Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: ITProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0


Employee Name: Rocca, Denis Yearly Salary: 45675.0
DeptName: Human Resources Dept Code: HRProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0


Employee Name: Anders, Jamie Yearly Salary: 71775.0
DeptName: Human Resources Dept Code: HRProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0


Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: ITProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0

--------- end employee list ----------
_____________________
Employees with Matching Departments (1)

Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: ITProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0

Employee Name: Karakus, Zana Yearly Salary: 56115.0
DeptName: Information Technology Dept Code: ITProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0

_____________________
Employees with Matching Departments (2)
```

```
Employee Name: Rocca, Denis Yearly Salary: 45675.0
DeptName: Human Resources Dept Code: HRProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0

Employee Name: Anders, Jamie Yearly Salary: 71775.0
DeptName: Human Resources Dept Code: HRProject Name: SunMarkets POS
Implementation
Project ID: 2023-5001
Project Type: l
Team Size: 1
Estimated Project Cost: 532004.0
```