Problem Statement

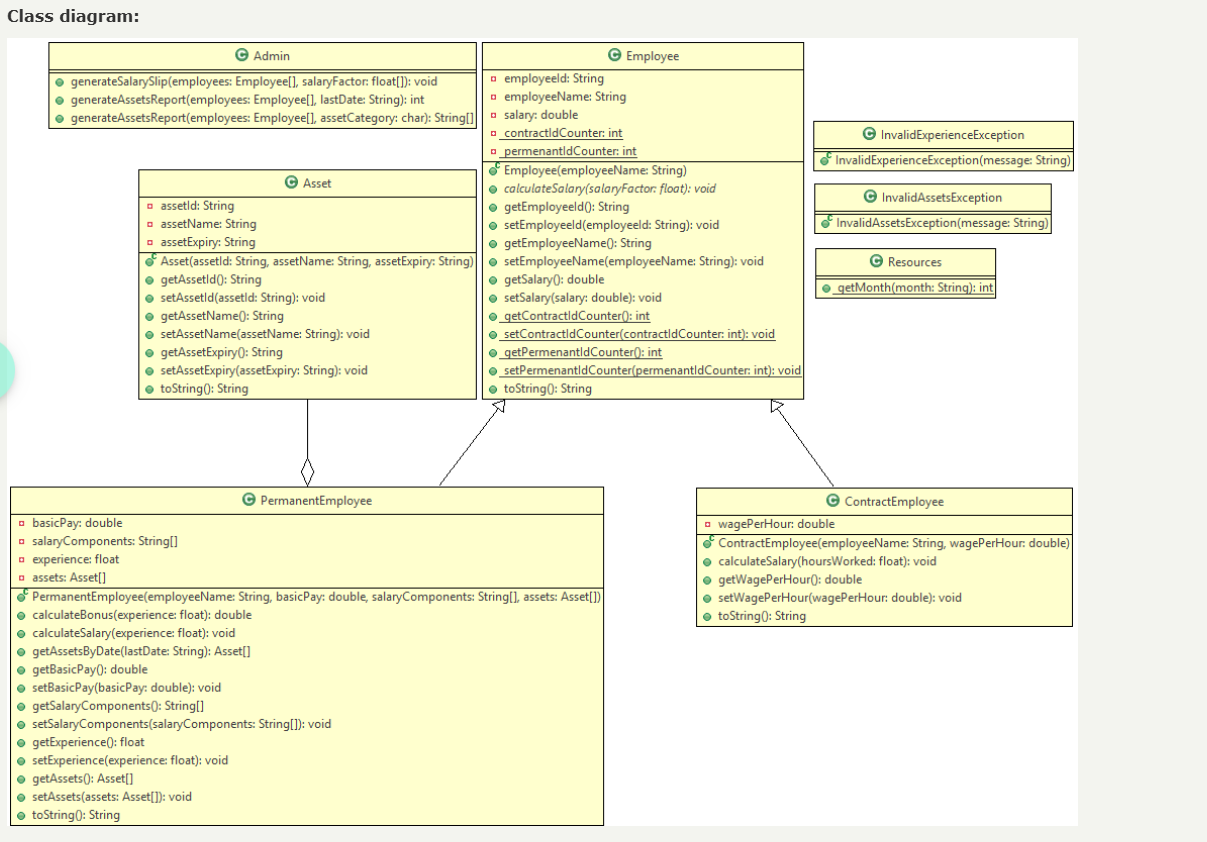
**Instructions:**

* Follow the class diagram strictly. Read the problem statement, functionality and the other details provided carefully and implement the solution.
* DO NOT add any new method apart from the ones mentioned in the class diagram.
* DO NOT modify the code provided to you.
* Use case-sensitive comparisons wherever applicable.

**Problem Statement**

A private sector firm wants to develop an application for automating various operations. You need to help the firm by developing the application based on the class diagram and description given below.

**Class diagram:**



NOTE

contractIdCounter and permenantIdCounter in Employee class are static variables.

calculateSalary in Employee class is an abstract method.

getContractIdCounter, setContractIdCounter, getPermenantIdCounter and setPermenantIdCounter in Employee class are static methods.

getMonth in Resources class is a static method.

Implementation details

Asset class

Asset(String assetId, String assetName, String assetExpiry)

The instance variables should be initialized with the values passed to the constructor appropriately.

setAssetId(String assetId)

assetId instance variable should be set to the passed value if the passed valued is valid.

assetId passed is valid if it satisfies the following rules:

assetId should start with either "DSK" or "LTP" or "IPH" followed by a hyphen (-) and 6 digits and a character which should be either 'H' or 'L' (case-insensitive).

Implement other setter and getter methods appropriately.

Resources class

getMonth(String month)

The method should receive a short form of month in String format and return the corresponding month number.

For an invalid month, 0 should be returned.

month received will contain 3 characters starting with uppercase character.

E.g. -

If the month value received is Sep, 9 should be returned

If the month value received is sep, 0 should be returned

InvalidAssetsException class

InvalidAssetsException(String message)

The instance variable should be initialized with the value passed to the constructor appropriately.

InvalidExperienceException class

InvalidExperienceException(String message)

The instance variable should be initialized with the value passed to the constructor appropriately.

Employee class

The static variables contractIdCounter and permenantIdCounter should be initialized to 10000 using a static block.

Employee(String employeeName)

The instance variable employeeName should be initialized with the value passed to the constructor appropriately.

employeeId should be generated either using contractIdCounter static variable or permenantIdCounter static variable as follows:

The value of employeeId should start from 'C10001' and the numerical part should be incremented by 1 for the subsequent values, if the employee is a contract employee.

The value of employeeId should start from 'E10001' and the numerical part should be incremented by 1 for the subsequent values, if the employee is a permenant employee.

setEmployeeName(String employeeName)

employeeName instance variable should be set to the passed value if the passed value is valid.

employeeName passed is valid if it satisfies the following rules:

Employee name can contain only alphabets and spaces. Space is allowed only between words.

Name should contain minimum 2 words.

Each word in the employee name should contain minimum 2 alphabets and it must start with an uppercase alphabet.

setSalary(double salary)

salary instance variable should be set to the passed value if the passed value is greater than 0, else, salary instance variable should be set to 0.

Implement other setter and getter methods appropriately.

ContractEmployee class

ContractEmployee(String employeeName, double wagePerHour)

The inherited member variables should be initialized by invoking the base class constructor.

The member variable of ContractEmployee class should be initialized with the value passed to the constructor appropriately.

calculateSalary(float hoursWorked)

The value of salary should be calculated using the formula given below:

salary = wagePerHour \* hoursWorked

hoursWorked should be validated and checked whether it is greater than or equal to 190

If hoursWorked is less than 190, a portion of the salary should be deducted as per the following:

salary to be deducted will be the product of half of the wagePerHour and the number of hours less than the limit, i.e., 190.

The calculated salary should be rounded to the nearest integer and should be set to the instance variable salary.

Implement the setter and getter methods appropriately.

PermanentEmployee class

PermanentEmployee(String employeeName, double basicPay, String[] salaryComponents, Asset[] assets)

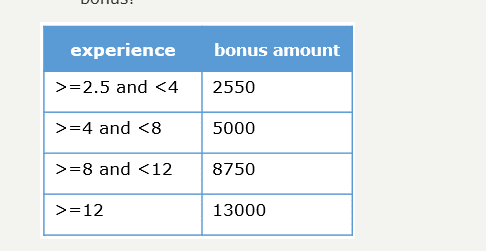
The inherited member variables should be initialized by invoking the base class constructor.

The member variables of PermanentEmployee class should be initialized with the values passed to the constructor appropriately.

calculateBonus(float experience)

Bonus amount should be calculated and returned as per the table given below

If the value of experience is invalid, InvalidExperienceException should be thrown with the message "A minimum of 2.5 years is required for bonus!"



calculateSalary(float experience)

The instance variable experience should be set using the value passed to the method.

The salary should be calculated using the formula given below:

salary = basicPay + DA component + HRA component + bonus amount

DA component and HRA component should be calculated based on the percentage value associated with each component given in the salaryComponents array.

E.g. - If the salaryComponents is ["DA-50", "HRA-45"], DA is 50% of the basicPay and HRA is 45% of the basicPay.

The bonus amount should be calculated by invoking the calculateBonus() method

If InvalidExperienceException is thrown by the calculateBonus() method, the InvalidExperienceException exception should be handled and bonus amount should be set to 0

The calculated salary should be rounded to the nearest integer and should be set to the instance variable salary.

getAssetsByDate(String lastDate)

An Asset array (Asset[]) containing all the allocated assets whose expiry is on or before the lastDate passed to the method should be returned.

The length of the array being returned should be same as that of total number of assets allocated to the employee.

If there are no allocated assets whose expiry is on or before the lastDate passed to the method, InvalidAssetsException should be thrown with the message "No assets found for the given criteria!"

HINT:

The dates will be in the following String format: "YYYY-MON-DD". E.g. - "2023-Dec-05"

getMonth() method of the Resources class should be invoked for validating the month

Implement the setter and getter methods appropriately.

Admin class

generateSalarySlip(Employee[] employees, float[] salaryFactor)

The calculateSalary() method of the appropriate Employee class should be invoked by passing the corresponding value from the salaryFactor array.

The salaryFactor array contains float values which are either hoursWorked or experience.

There is one to one mapping between the employees array and the salaryFactor array.

generateAssetsReport(Employee[] employees, String lastDate)

The total number of allocated assets whose expiry is on or before the lastDate passed to the method for all the employees present in the employees array should be returned

For calculating the total number of allocated assets, getAssetsByDate() method of the PermanentEmployee class by should be invoked by passing the lastDate if the employee is a permanent employee.

If InvalidAssetsException is thrown by the getAssetsByDate() method, the InvalidAssetsException exception should be handled and -1 should be returned

Assumption: employees array will contain contract employees and permanent employees who are allocated with assets.

generateAssetsReport(Employee[] employees, char assetCategory)

The assetIds of all the allocated assets whose assetId starts with the given character for all the employees present in the employees array should be returned. Case-insensitive comparison should be performed.

The length of the array being returned should be thrice the length of employees array.

Assumption: employees array will contain contract employees and permanent employees who are allocated with assets.

Test the functionalities using the provided Tester class.

Sample Output

Details of all available assets

Details of asset1

Asset Id: DSK-876761L

Asset Name: Dell-Desktop

Asset Valid Till: 2020-Dec-01

Details of asset2

Asset Id: DSK-876762L

Asset Name: Acer-Desktop

Asset Valid Till: 2021-Mar-31

Details of asset3

Asset Id: DSK-876763L

Asset Name: Dell-Desktop

Asset Valid Till: 2022-Jun-12

Details of asset4

Asset Id: LTP-987123H

Asset Name: Dell-Laptop

Asset Valid Till: 2021-Dec-31

Details of asset5

Asset Id: LTP-987124h

Asset Name: Dell-Laptop

Asset Valid Till: 2021-Sep-20

Details of asset6

Asset Id: LTP-987125L

Asset Name: HP-Laptop

Asset Valid Till: 2022-Oct-25

Details of asset7

Asset Id: LTP-987126l

Asset Name: HP-Laptop

Asset Valid Till: 2021-Oct-02

Details of asset8

Asset Id: IPH-110110h

Asset Name: VoIP

Asset Valid Till: 2021-Dec-12

Details of asset9

Asset Id: null

Asset Name: VoIP

Asset Valid Till: 2020-Dec-31

Details of asset10

Asset Id: IPH-110130h

Asset Name: VoIP

Asset Valid Till: 2020-Nov-30

Correcting all the invalid assetIds

Details of asset9

Asset Id: IPH-110120h

Asset Name: VoIP

Asset Valid Till: 2020-Dec-31

Initiating salary calculation...

Details of employees

Details of permanentEmployee1

Employee Id: E10001

Employee Name: Roger Fed

Salary: 32000.0

Experience: 3.9

Assets Allocated: DSK-876761L IPH-110130h

Details of permanentEmployee2

Employee Id: E10002

Employee Name: null

Salary: 25200.0

Experience: 2.3

Assets Allocated: LTP-987125L IPH-110120h

Details of permanentEmployee3

Employee Id: E10003

Employee Name: James Peter

Salary: 41075.0

Experience: 4.0

Assets Allocated: LTP-987123H

Details of permanentEmployee4

Employee Id: E10004

Employee Name: Catherine Maria

Salary: 47750.0

Experience: 8.1

Assets Allocated: DSK-876762L LTP-987124h

Details of permanentEmployee5

Employee Id: E10005

Employee Name: Jobin Nick

Salary: 55000.0

Experience: 12.5

Assets Allocated: No assets allocated!

Details of contractEmployee1

Employee Id: C10001

Employee Name: null

Salary: 13195.0

Details of contractEmployee2

Employee Id: C10002

Employee Name: Ricky Neol

Salary: 15298.0

Reports

Number of allocated assets expiring on or before 2021-Dec-31: 6

Sorry, report cannot be generated!

All the allocated desktop assets

DSK-876761L

DSK-876762L

All the allocated laptop assets

LTP-987125L

LTP-987123H

LTP-987124h

All the allocated VoIP assets

IPH-110130h

IPH-110120h