Assignment 5

- Inheritance
- Access specifiers
- super

1. University Management System

Description:

Create a program for a university management system. The program should have the following classes:

- `University`
- `Department`
- `Professor`
- `Student`

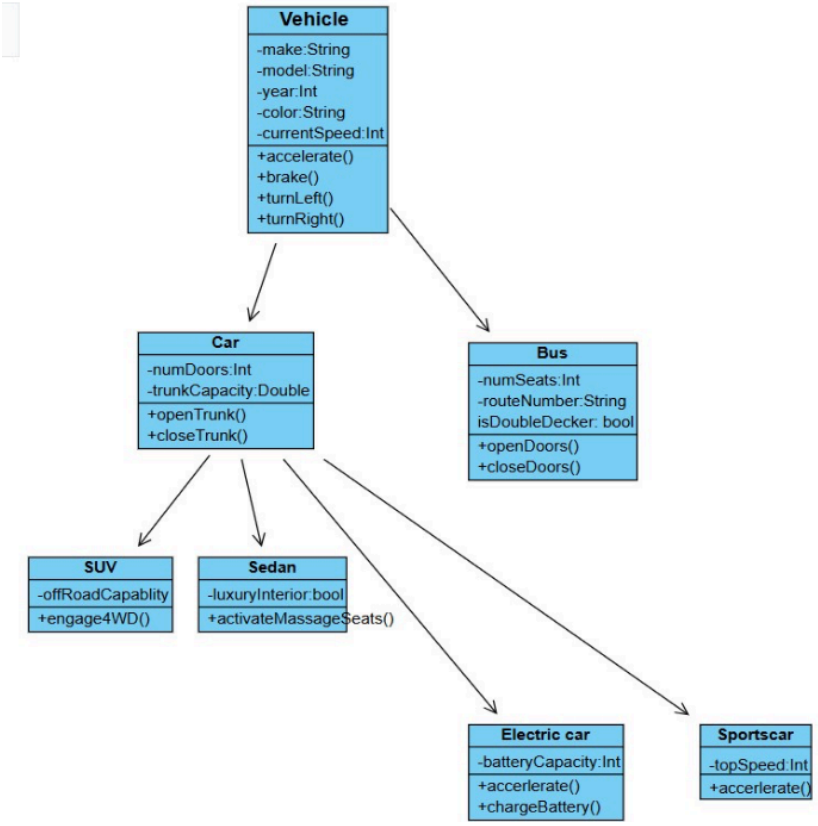
Requirements:

1. The `University` class should contain a list of departments.
2. The `Department` class should contain a list of professors and a list of students.
3. The `Professor` class should contain information about the professor (e.g., name, id, department).
4. The `Student` class should contain information about the student (e.g., name, id, major).

Tasks:

1. Create instances of `University`, `Department`, `Professor`, and `Student`.
2. Populate the university with departments, professors, and students.
3. Display the details of the university, including its departments, professors, and students.

2. Design a vehicle management system that involves multiple levels of inheritance, as well as multiple classes inheriting from the same base class. The system includes a Vehicle class as the base class, and several derived classes like Car, Bus. Additionally, extend the Car class further into more specialized classes like ElectricCar and SportsCar. You also have scenarios where different classes like SUV and Sedan inherit from the Car class.



3. Write a Java program for a simple banking system. Create the following classes:

- Account - A base class with the following members:
  - Private field accountNumber (String)
  - Protected field balance (double)
  - Public method deposit(double amount) which adds the amount to balance.
  - Public method getBalance() which returns the current balance.
- SavingsAccount - A subclass of Account with an additional feature:
  - Protected field interestRate (double)
  - Public method addInterest() which adds interest to the balance based on interestRate.

Instantiate a SavingsAccount object, deposit an amount, add interest, and print the balance.

4. Write a Java program to demonstrate how access specifiers affect access from different packages. Create two packages: **company** and department. In the company package, define the following classes:

- Employee - A class with:

    - Protected field name (String)
    - Public method getName() to get the name.
- Manager - A subclass of Employee with:
    - Public method printName() to print the name.

In the department package, define a class **Office** which tries to access and print the **name** field from an Employee instance and show how the **name** field can be accessed and modified in the Manager class.

5. In a company, the employee management system is designed to manage different

types of employees, such as officers and managers. The system needs to store common details about employees like name, age, address, phone number, and salary, as well as specific details about officers and managers. Officers have a specialization field, and managers have a department field.
Requirements:
1. Common Employee Details:
○ The system should allow for retrieving and updating common details of any employee.
○ There should be a method to calculate the annual salary of an employee.
2. Officer Specific Details:
● The system should manage the specialization field specific to officers.
● There should be a method to calculate the salary of an officer, factoring in their specialization.
3. Manager Specific Details:
● The system should manage the department field specific to managers.
● There should be a method to calculate the salary of a manager, factoring in their Department.

6. Write a java program to create a simple class hierarchy for a vehicle management system with two packages:

- **Package** vehicles:
    - **Class** Vehicle:
        - Field: licensePlate (String)
        - Public method getLicensePlate() to return the license plate.
- **Package** management:
    - **Class** Car (extends vehicles.Vehicle):
        - Additional Field: model (String)
        - Constructor to initialize all fields (using super for Vehicle fields)
        - Public method getDetails() to return the license plate and model.

7. Imagine you are building a computer program for a large company that needs to track both its transportation fleet and its staff members. First, create a general Vehicle category that stores a model name and fuel level, then create two specific types, Car and Truck, that inherit from it and have their own unique ways of describing how they drive. Second, create a general Employee category that stores a name and ID number, then create two specific types, Manager and Developer, that inherit from it and have their own unique ways of describing their daily work tasks. For all four of these specific types, you must use a constructor to pass the basic information (like name or model) up to the parent category.

Hint: user super