KIRAN KUMAR D NATIKAR
EMP ID : 2576955
WRITTEN PART:

Define two classes: Vehicle and Property to represent vehicle and property information, respectively.

- Create two operations classes: VehicleOperations and PropertyOperations to manage the collection and operations related to vehicles and properties.
- In the Vehicle class, define attributes like registration number, brand, cost, velocity, capacity, vehicle type, and vehicle tax. Similarly, in the Property class, define attributes like ID, base value of land, location, age of property, and property tax.
- Implement constructors, getters, and setters for both classes to initialize and manipulate the object properties.
- In the VehicleOperations class, manage a collection of Vehicle objects using an ArrayList.
- Implement methods to add vehicle details (addVehicleDetails), view vehicle details (viewVehicleDetails), and calculate vehicle tax (calculateVehicleTax) based on predefined tax rules.
- In the PropertyOperations class, manage a collection of Property objects using an ArrayList.
- Implement methods to add property details (addPropertyDetails), view property details (viewPropertyDetails), and calculate property tax (calculatePropertyTax) based on predefined tax rules.
- In the taxCalculation class (the main class):
- Present a login prompt to the user with a predefined username and password.
- Upon successful login, create instances of PropertyOperations and VehicleOperations.
- Start an infinite loop to provide a menu for various tax-related operations.
- The menu includes options for property tax, vehicle tax, calculating the total tax, and exiting the application.
- For each option, take user input as needed and perform the corresponding operations by calling methods from PropertyOperations and VehicleOperations.
- Calculate and display the total property tax, total vehicle tax, and the overall total tax payable.
- Implement methods to calculate the total tax for both properties and vehicles (calculateTotalTax).
- Exit the application when the user chooses the exit option.
- Ensure appropriate error handling for invalid user input and exceptions.

This code represents a basic tax calculation application where users can input property and vehicle details, calculate taxes, and view the results. It demonstrates object-oriented programming principles and basic user interaction in a console-based Java application.