**package** taxcalculationapplication;

**import** java.util.ArrayList;

**import** java.util.List;

**import** java.util.Scanner;

**class** Property {

**private** **double** baseValueofland;

**private** **boolean** isInCity;

**private** **int** ageOfConstruction;

**private** **double** builtuparea;

**public** Property(**double** baseValueofland, **boolean** isInCity, **int** ageOfConstruction, **double** builtuparea) {

**this**.baseValueofland = baseValueofland;

**this**.isInCity = isInCity;

**this**.ageOfConstruction = ageOfConstruction;

**this**.builtuparea = builtuparea;

}

**public** **double** calculatePropertyTax() {

**double** tax;

**if** (isInCity) {

tax = (builtuparea \* ageOfConstruction \* baseValueofland) + (0.5 \* baseValueofland);

} **else** {

tax = builtuparea \* ageOfConstruction \* baseValueofland;

}

**return** tax;

}

@Override

**public** String toString() {

**return** "Base Value: " + baseValueofland + "\nIs in City: " + isInCity + "\nAge of Construction: " + ageOfConstruction

+ "\nBuilt Up Area: " + builtuparea;

}

}

**class** Vehicle {

**private** String registrationNumber;

**private** String brand;

**private** **double** purchaseCost;

**private** **double** maxVelocity;

**private** **int** capacity;

**private** **int** vehicleChoice;

**public** Vehicle(String registrationNumber, String brand, **double** purchaseCost, **double** maxVelocity, **int** capacity, **int** vehicleChoice) {

**this**.registrationNumber = registrationNumber;

**this**.brand = brand;

**this**.purchaseCost = purchaseCost;

**this**.maxVelocity = maxVelocity;

**this**.capacity = capacity;

**this**.vehicleChoice = vehicleChoice;

}

**public** **double** calculateVehicleTax() {

**double** tax = 0;

**switch** (vehicleChoice) {

**case** 1:

tax = maxVelocity + capacity + (0.1 \* purchaseCost);

**break**;

**case** 2:

tax = maxVelocity + capacity + (0.11 \* purchaseCost);

**break**;

**case** 3:

tax = maxVelocity + capacity + (0.12 \* purchaseCost);

**break**;

}

**return** tax;

}

@Override

**public** String toString() {

**return** "Registration Number: " + registrationNumber + "\nBrand: " + brand + "\nPurchase Cost: " + purchaseCost +

"\nMax Velocity: " + maxVelocity + "\nCapacity: " + capacity + "\nVehicle Type: " + vehicleChoice;

}

}

**public** **class** TaxCalculatorApp {

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

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

System.***out***.println("Welcome to the Tax Calculation Application");

System.***out***.print("Enter username: ");

String username = scanner.nextLine();

System.***out***.print("Enter password: ");

String password = scanner.nextLine();

**if** (*authenticate*(username, password)) {

List<Property> properties = **new** ArrayList<>();

List<Vehicle> vehicles = **new** ArrayList<>();

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

System.***out***.println("Tax Calculation Application");

System.***out***.println("1. Property");

System.***out***.println("2. Vehicle");

System.***out***.println("3. View and Calculate Taxes");

System.***out***.println("4. Exit");

System.***out***.print("Enter your choice: ");

**int** mainChoice = scanner.nextInt();

scanner.nextLine(); // Consume newline

**switch** (mainChoice) {

**case** 1:

// Property Options

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

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

System.***out***.println("1. Add Property Details");

System.***out***.println("2. Calculate Property Tax");

System.***out***.println("3. Display All Properties");

System.***out***.println("4. Back to Main Menu");

System.***out***.print("Enter your choice: ");

**int** propertyChoice = scanner.nextInt();

scanner.nextLine(); // Consume newline

**switch** (propertyChoice) {

**case** 1:

System.***out***.print("Enter Base Value of Land: ");

**double** baseValueOfLand = scanner.nextDouble();

System.***out***.print("Is in City (yes/no): ");

String isInCityInput = scanner.next();

**boolean** isInCity = isInCityInput.equals("Y") || isInCityInput.equals("y");

System.***out***.print("Enter Age of Construction: ");

**int** ageOfConstruction = scanner.nextInt();

System.***out***.print("Enter Built-Up Area: ");

**double** builtUpArea = scanner.nextDouble();

Property property = **new** Property(baseValueOfLand, isInCity, ageOfConstruction, builtUpArea);

properties.add(property);

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

**break**;

**case** 2:

System.***out***.println("\nProperty Taxes Calculator:");

System.***out***.println("==============================");

**for** (Property prop : properties) {

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

System.***out***.println("Property Tax: " + prop.calculatePropertyTax());

System.***out***.println("==============================");

}

**break**;

**case** 3:

**break**;

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

**if** (propertyChoice == 2) {

**break**; // Exit the property sub-menu

}

}

**break**;

**case** 2:

// Vehicle Options

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

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

System.***out***.println("1. Add Vehicle");

System.***out***.println("2. Back to Main Menu");

System.***out***.print("Enter your choice: ");

**int** vehicleChoice = scanner.nextInt();

scanner.nextLine(); // Consume newline

**switch** (vehicleChoice) {

**case** 1:

System.***out***.print("Enter Registration Number: ");

String registrationNumber = scanner.next();

System.***out***.print("Enter Brand: ");

String brand = scanner.next();

System.***out***.print("Enter Purchase Cost: ");

**double** purchaseCost = scanner.nextDouble();

System.***out***.print("Enter Max Velocity: ");

**double** maxVelocity = scanner.nextDouble();

System.***out***.print("Enter Capacity: ");

**int** capacity = scanner.nextInt();

Vehicle vehicle = **new** Vehicle(registrationNumber, brand, purchaseCost, maxVelocity, capacity, vehicleChoice);

vehicles.add(vehicle);

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

**break**;

**case** 2:

// Return to the main menu

**break**;

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

**if** (vehicleChoice == 2) {

**break**; // Exit the vehicle sub-menu

}

}

**break**;

**case** 3:

// View and Calculate Taxes

// Include code to display property and vehicle details and calculate taxes (similar to previous example)

System.***out***.println("\nProperty Details and Taxes:");

**for** (Property prop : properties) {

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

System.***out***.println("Property Tax: " + prop.calculatePropertyTax());

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

}

System.***out***.println("\nVehicle Details and Taxes:");

**for** (Vehicle veh : vehicles) {

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

System.***out***.println("Vehicle Tax: " + veh.calculateVehicleTax());

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

}

**break**;

**case** 4:

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

scanner.close();

System.*exit*(0);

**default**:

System.***out***.println("Invalid choice. Please try again.");

}

}

} **else** {

System.***out***.println("Authentication failed. Exiting the application.");

}

scanner.close();

}

**private** **static** **boolean** authenticate(String username, String password) {

**return** username.equals("NancySingh") && password.equals("12345");

}

}