```
package phase;

import java.util.ArrayList;
import java.util.Scanner;

class Property {
    private String location;
    private double baseValue;
    private double builtuparea; // Added builtuparea property
    private int age;
    private double propertyTax;

    public Property(String location, double baseValue, int age, double
builtuparea) {
        this.location = location;
        this.baseValue = baseValue;
        this.age = age;
        this.builtuparea = builtuparea;
        this.propertyTax = 0.00;
    }

    public void calculatePropertyTax() {
        if ("Y".equalsIgnoreCase(location)) {
            propertyTax = (builtuparea * age * baseValue) + (0.5 * builtuparea);
        } else {
            propertyTax = builtuparea * age * baseValue;
        }
    }

    public double getPropertyTax() {
        return propertyTax;
    }

    public String getLocation() {
        return location;
    }

    public double getBaseValue() {
        return baseValue;
    }

    public int getAge() {
        return age;
    }

    public double getBuiltuparea() { // Added getBuiltuparea() method
        return builtuparea;
    }
}

class Vehicle {
    private String registrationNumber;
    private String brand;
    private double purchaseCost;
    private double velocity;
    private int capacity;
    private int vehicleType; // 1: Petrol, 2: Diesel, 3: CNG/LPG
    private double vehicleTax;
    public Vehicle(String registrationNumber, String brand, double purchaseCost,
double velocity, int capacity,
            int vehicleType) {
        this.registrationNumber = registrationNumber;
        this.brand = brand;
        this.purchaseCost = purchaseCost;
        this.velocity = velocity;
        this.capacity = capacity;
        this.vehicleType = vehicleType;
        this.vehicleTax = 0.0;
    }

    public void calculateVehicleTax() {
        switch (vehicleType) {
            case 1: // Petrol
                vehicleTax = velocity + capacity + 0.1 * purchaseCost;
                break;
            case 2: // Diesel
                vehicleTax = velocity + capacity + 0.11 * purchaseCost;
                break;
            case 3: // CNG/LPG
                vehicleTax = velocity + capacity + 0.12 * purchaseCost;
                break;
            default:
                break;
        }
    }

    public double getVehicleTax() {
        return vehicleTax;
    }

    public String getRegistrationNumber() {
        return registrationNumber;
    }

    public String getBrand() {
        return brand;
    }

    public double getPurchaseCost() {
        return purchaseCost;
    }

    public double getVelocity() {
        return velocity;
    }

    public int getCapacity() {
        return capacity;
    }

    public int getVehicleType() {
        return vehicleType;
    }
}

public class CalculatorTax {
    private static final String USERNAME = "admin"; // Change to your desired
username
    private static final String PASSWORD = "password"; // Change to your desired
password

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ArrayList<Property> properties = new ArrayList<>();
        ArrayList<Vehicle> vehicles = new ArrayList<>();

        System.out.print("Enter username: ");
        String enteredUsername = scanner.nextLine();

        System.out.print("Enter password: ");
        String enteredPassword = scanner.nextLine();

        if (authenticateUser(enteredUsername, enteredPassword)) {
            System.out.println("Authentication successful. Welcome, " +
enteredUsername + "!\n");

            while (true) {
                System.out.println("Main Menu:");
                System.out.println("1. Property Tax");
                System.out.println("2. Vehicle Tax");
                System.out.println("3. Total");
                System.out.println("4. Exit");
                System.out.print("Enter your choice: ");
                int choice = scanner.nextInt();
                scanner.nextLine(); // Consume newline

                switch (choice) {
                    case 1:
                        handlePropertyTax(properties, scanner);
                        break;
                    case 2:
                        handleVehicleTax(vehicles, scanner);
                        break;
                    case 3:
                        calculateTotalTax(properties, vehicles);
                        break;
                    case 4:
                        System.out.println("Exiting the application. Goodbye!");
                        System.exit(0);
                    default:
                        System.out.println("Invalid choice. Please select a valid
option.");
                }
            }
        } else {
            System.out.println("Authentication failed. Exiting the application.");
        }
    }

    private static boolean authenticateUser(String username, String password) {
        return username.equals(USERNAME) && password.equals(PASSWORD);
    }

    private static void handlePropertyTax(ArrayList<Property> properties, Scanner
scanner) {
        while (true) {
            System.out.println("\nProperty Tax 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 Menu");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    // Add property details logic
                    System.out.print("Enter Base Value: ");
                    double propertyBaseValue = scanner.nextDouble();
                    System.out.print("Enter Built-up Area: ");
                    double builtUpArea = scanner.nextDouble();
                    System.out.print("Enter Age of Construction: ");
                    int age = scanner.nextInt();
                    System.out.print("Is the property in the city? (Y/N): ");
                    String location = scanner.next();

                    Property property = new Property(location, propertyBaseValue,
age, builtUpArea);
                    property.calculatePropertyTax();
                    properties.add(property);
                    System.out.println("Property added successfully.");
                    break;
                case 2:
                    // Calculate property tax logic
                    double totalPropertyTax = 0;
                    for (Property prop : properties) {
                        prop.calculatePropertyTax();
                        totalPropertyTax += prop.getPropertyTax();
                    }
                    System.out.println("Total Property Tax: $" +
totalPropertyTax);
                    break;
                case 3:
                    // Display all properties logic
                    if (properties.isEmpty()) {
                        System.out.println("No properties to display.");
                    } else {
                        displayPropertyTaxTable(properties);
                    }
                    break;
                case 4:
                    return; // Return to the main menu
                default:
                    System.out.println("Invalid choice. Please select a valid
option.");
            }
        }
    }

    private static void handleVehicleTax(ArrayList<Vehicle> vehicles, Scanner
scanner) {
        while (true) {
            System.out.println("\nVehicle Tax Menu:");
            System.out.println("1. Add Vehicle Details");
            System.out.println("2. Calculate Vehicle Tax");
            System.out.println("3. Display All Vehicles");
            System.out.println("4. Back to Menu");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    // Add vehicle details logic
                    System.out.print("Enter Registration Number: ");
                    String regNumber = 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 Maximum Velocity (km/h): ");
                    double velocity = scanner.nextDouble();
                    System.out.print("Enter Capacity (number of seats): ");
                    int capacity = scanner.nextInt();
                    System.out.println("Select Vehicle Type:");
                    System.out.println("1. Petrol-driven");
                    System.out.println("2. Diesel-driven");
                    System.out.println("3. CNG/LPG-driven");
                    int vehicleType = scanner.nextInt();

                    Vehicle vehicle = new Vehicle(regNumber, brand, purchaseCost, 
velocity, capacity, vehicleType);
                    vehicle.calculateVehicleTax();
                    vehicles.add(vehicle);
                    System.out.println("Vehicle added successfully.");
                    break;
                case 2:
                    // Calculate vehicle tax logic
                    double totalVehicleTax = 0;
                    for (Vehicle veh : vehicles) {
                        veh.calculateVehicleTax();
                        totalVehicleTax += veh.getVehicleTax();
                    }
                    System.out.println("Total Vehicle Tax: $" + totalVehicleTax);
                    break;
                case 3:
                    // Display all vehicles logic
                    if (vehicles.isEmpty()) {
                        System.out.println("No vehicles to display.");
                    } else {
                        displayVehicleTaxTable(vehicles);
                    }
                    break;
                case 4:
                    return; // Return to the main menu
                default:
                    System.out.println("Invalid choice. Please select a valid 
option.");
            }
        }
    }
    private static void calculateTotalTax(ArrayList<Property> properties,
ArrayList<Vehicle> vehicles) {
        // Calculate and display the total tax (property tax + vehicle tax)
        double totalPropertyTax =
properties.stream().mapToDouble(Property::getPropertyTax).sum();
        double totalVehicleTax =
vehicles.stream().mapToDouble(Vehicle::getVehicleTax).sum();
        double totalTax = totalPropertyTax + totalVehicleTax;

            System.out.println("Total Tax Table:");
            System.out.println("--------------------------------------------------
-----------");
            System.out.printf("%-20s %-20s %-20s%n", "Total Property Tax", "Total
Vehicle Tax", "Total Tax");
            System.out.println("--------------------------------------------------
-----------");
            System.out.printf("$%-19.2f $%-19.2f $%-19.2f%n", totalPropertyTax,
totalVehicleTax, totalTax);
            System.out.println("--------------------------------------------------
-----------");
        }


    private static void displayPropertyTaxTable(ArrayList<Property> properties) {
        System.out.println("Property Tax Table:");
        System.out.println("------------------------------------------------------
--------------------------");
        System.out.printf("%-15s %-15s %-10s %-15s %-20s%n", "Location", "Base
Value", "Age", "BuiltUpArea", "Property Tax ($)");
        System.out.println("------------------------------------------------------
--------------------------");
        for (Property prop : properties) {
            System.out.printf("%-15s $%-14.2f %-9d %-14.2f $%-19.2f%n",
prop.getLocation(), prop.getBaseValue(),
                    prop.getAge(), prop.getBuiltuparea(), prop.getPropertyTax());
        }
        System.out.println("------------------------------------------------------
--------------------------");
    }

    private static void displayVehicleTaxTable(ArrayList<Vehicle> vehicles) {
        System.out.println("Vehicle Tax Table:");
        System.out.println(
                "-----------------------------------------------------------------
---------------------------");
        System.out.printf("%-20s %-20s %-20s %-20s %-20s %-20s %-20s%n", "Reg
Number", "Brand", "Purchase Cost",
                "Velocity (km/h)", "Capacity", "Vehicle Type", "Vehicle Tax ($)");
        System.out.println(
                "-----------------------------------------------------------------
---------------------------");
        for (Vehicle veh : vehicles) {
            System.out.printf("%-19s %-19s $%-19.2f %-19.2f %-19d %-19s $%-
19.2f%n", veh.getRegistrationNumber(),
                    veh.getBrand(), veh.getPurchaseCost(), veh.getVelocity(),
veh.getCapacity(),
                    getVehicleTypeName(veh.getVehicleType()),
veh.getVehicleTax());
        }
        System.out.println(
                "-----------------------------------------------------------------
---------------------------");
    }

    private static String getVehicleTypeName(int vehicleType) {
        switch (vehicleType) {
            case 1:
                return "Petrol";
            case 2:
                return "Diesel";
            case 3:
                return "CNG/LPG";
            default:
                return "Unknown";
        }
    }
}
```