

Java Program: Vehicle Details

```
import java.util.Scanner;
import java.time.Year;
class Vehicle {
    String brand;
    String model;
    int year;
    double basePrice;
    Vehicle(String brand, String model, int year, double basePrice) {
        this.brand = brand;
        this.model = model;
        this.year = year;
        this.basePrice = basePrice;
    }
    void displayInfo() {
        System.out.println("Brand: " + brand);
        System.out.println("Model: " + model);
        System.out.println("Year: " + year);
        System.out.printf("Base Price: %.2f\n", basePrice);
    }
    void calculateResaleValue() {
        int currentYear = Year.now().getValue();
        int age = currentYear - year;
        double resaleValue = basePrice * Math.pow(0.85, age);
        System.out.printf("Resale Value: %.2f\n", resaleValue);
    }
}
class Car extends Vehicle {
    int numberOfDoors;
    Car(String brand, String model, int year, double basePrice, int numberOfDoors) {
        super(brand, model, year, basePrice);
        this.numberOfDoors = numberOfDoors;
    }
    void calculateInsurance() {
        double insurance = 5000 * basePrice + 1000 * numberOfDoors;
        System.out.printf("Insurance: %.2f\n", insurance);
    }
    void calculateMaintenanceCost() {
        int age = Year.now().getValue() - year;
        double maintenanceCost = 3000 * age + 500 * numberOfDoors;
        System.out.printf("Maintenance: %.2f\n", maintenanceCost);
    }
    void totalDetails() {
        System.out.println("\n--- Car Details ---");
        displayInfo();
        calculateResaleValue();
        calculateInsurance();
        calculateMaintenanceCost();
        System.out.println("Number of Doors: " + numberOfDoors);
    }
}
class Motorcycle extends Vehicle {
    boolean hasSidecar;
    Motorcycle(String brand, String model, int year, double basePrice, boolean hasSidecar) {
        super(brand, model, year, basePrice);
        this.hasSidecar = hasSidecar;
    }
    void calculateInsurance() {
        double insurance;
        if (hasSidecar) {
            insurance = 2500 * basePrice + 1500;
        } else {
            insurance = 2500 * basePrice;
        }
        System.out.printf("Insurance: %.2f\n", insurance);
    }
    void calculateMaintenanceCost() {
        int age = Year.now().getValue() - year;
        double maintenanceCost;
        if (hasSidecar) {
            maintenanceCost = 2000 * age + 1000;
        } else {
            maintenanceCost = 2000 * age;
        }
    }
}
```

```

    }
    System.out.printf("Maintenance: %.2f\n", maintenanceCost);
}
void totalDetails() {
    System.out.println("\n--- Motorcycle Details ---");
    displayInfo();
    calculateResaleValue();
    calculateInsurance();
    calculateMaintenanceCost();
    System.out.println("Sidecar: " + hasSidecar);
}
}
public class VehicleDetails {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("-----Enter Car Details-----");
        System.out.print("Enter Brand: ");
        String carBrand = sc.nextLine();
        System.out.print("Enter Model: ");
        String carModel = sc.nextLine();
        System.out.print("Enter Year: ");
        int carYear = sc.nextInt();
        System.out.print("Enter Base Price: ");
        double carBasePrice = sc.nextDouble();
        System.out.print("Enter Number of Doors: ");
        int numberOfDoors = sc.nextInt();
        sc.nextLine();
        Car car = new Car(carBrand, carModel, carYear, carBasePrice, numberOfDoors);
        System.out.println("-----Enter Motorcycle Details-----");
        System.out.print("Enter Brand: ");
        String bikeBrand = sc.nextLine();
        System.out.print("Enter Model: ");
        String bikeModel = sc.nextLine();
        System.out.print("Enter Year: ");
        int bikeYear = sc.nextInt();
        System.out.print("Enter Base Price: ");
        double bikeBasePrice = sc.nextDouble();
        System.out.print("Does it have a sidecar (true/false)? ");
        boolean hasSidecar = sc.nextBoolean();
        Motorcycle motorcycle = new Motorcycle(bikeBrand, bikeModel, bikeYear, bikeBasePrice, hasSidecar);
        System.out.println("\n-----Vehicle Details-----");
        car.totalDetails();
        motorcycle.totalDetails();
        sc.close();
    }
}

```