CAR BOOKING PROJECT

A Simple Car Management System in Java

Abstract

Car Booking Dealing Project in Java is a simple yet comprehensive system for managing cars. It allows users to add cars to the system, view a list of available cars, and exit the program. The project is implemented using classes like Car and Car Booking System, with methods for adding cars and displaying car details. The main class, Car Booking Dealing Project, handles user interaction through a menu-based system. This documentation provides a detailed overview of the project's structure, functionality, and potential enhancements, making it a valuable resource for Java developers looking to build similar systems.

Name: Attaur Rehman

Reg #: SU-21-01-001-002

Section: B

Dept: Computer Science

Semester: 5th

Subject: Prog in Java

Project: Car Booking & Dealing

Submit to: Sir Ayub Ashraf

Importance

This documentation provides a detailed explanation of the Car Booking Dealing Project, including information about data types, return types, parameterized functions, non-parameterized functions, classes, objects, variables, and more. This project is important as it serves as a simple and practical example for Java learners to understand basic concepts like object-oriented programming, user input handling, and data management. It provides a hands-on opportunity to practice coding skills and can be a stepping stone for more complex projects in the future.

Let’s discuss each one by one.

Car class:

Represents a car with its brand, model, year, and price.

Constructor:

```
public Car(String brand, String model, int year, double price) {
 this.brand = brand;
   this.model = model;
    this.year = year;
    this.price = price;
}
```

Method:

Displays the car details in a user-friendly format.

Format: Brand: Toyota, Model: Camry, Year: 2024, Price: $35,000.00

```
public void display() {
        // Implement code to format and display car details
}
```

Data Types:

- Brand: String
- Model: String

- Year: int
- Price: double

Return Type:

- Constructor: N/A
- Method: void

CarBookingSystem Class:

Purpose:

Manages the car system by allowing users to add cars and display the list of cars.

Attributes:

- cars (Car[]): An array of Car objects representing the cars in the system.
- numOfCars (int): The number of cars currently in the system.

Methods:

Adds a new car to the system. Displays an error message if the system is full.

```
public void addCar(Car car) {

   // Implement code to add car

   // Check for maximum car limit (100) before adding

}
```

Displays the details of all cars in the system.

```
public void displayCars() {

// Implement code to display all cars
}
```

Data Types:

- cars: Car[]
- numOfCars: int

Return Type:

- addCar(): void
- displayCars(): void

Parameterized Function:

- addCar(Car car)

Non-Parameterized Function:

- displayCars()

Main Class (CarBookingDealingProject):

Purpose:

Implements a simple car management system with user interaction.

Features:

- Users can add a car by providing brand, model, year, and price.
- Users can view the list of all cars in the system.
- Users can exit the program.

Usage:

- Run the program.
- Follow the on-screen menu to perform actions:
- Add a car
- Display cars
- Exit the program

Code Functionality:

Main Program:

- Creates a Scanner object for user input and a CarBookingSystem object.
- Uses an infinite loop to keep the program running until the user chooses to exit.
- Displays a menu for adding cars, displaying cars, or exiting.
- Calls the appropriate method from the CarBookingSystem class based on user input.

Adding Cars:

- Prompts the user to enter details for a new car.
- Creates a new Car object with the entered details.
- Attempts to add the new car to the system using the addCar(Car car) method.
- Displays an error message if the system is full.

Displaying Cars:

- Calls the displayCars() method of the CarBookingSystem class to display details of all cars in the system.

Exiting the Program:

Exits the program using System.exit(0) when the user chooses to exit.

Enhancements:

Specify data types for user input (e.g., String for brand, int for year, double for price) for improved user experience.

Functionality:

- Users can add cars to the system by providing details such as brand, model, year, and price.
- Users can view a list of all cars currently available in the system.
- The system prevents adding more than 100 cars to ensure it doesn't exceed its capacity.
- The program runs in an infinite loop, allowing users to perform actions until they choose to exit.
- User interactions are managed through a simple menu-based system.
- The program uses classes to represent cars and the car booking system, demonstrating basic object-oriented programming principles.