

Group Assignment part 1 - Basic Garage System

System to help administer a Garage

Introduction

As a base for your group assignment a console application should be built to handle all the basic functionality of a garage. This system will cover the techniques you've learned during the FUB alongside LINQ and the ability to break down a bigger problem into smaller parts. The application should handle Vehicles of different types and functionality to assist workers in the garage.

The programs purpose

The program in this iteration is supposed to help workers in the garage to check in vehicles that customers want to park. A customer can drop of their vehicle, and a worker will take the vehicle information into the system, get a parking spot and drive the vehicle to that spot. The system should also handle a fee, based on which type of vehicle it is, and how long it's been parked.

This iteration focuses on base functionality, and should be thought of as a base for the coming group assignments, which will add a database, server functionality and a web UI. This iteration shall handle all the functionality listed below, and shouldn't crash when given the wrong input.

Functionality and Requirements (this iteration)

The program should have the following functionality:

- Possibility to see basic information for all vehicles parked in the garage
- Possibility to see all information about a specific vehicle parked in the garage
- Possibility to check in a vehicle to the garage
 - This should assign a parking spot to the vehicle automatically
 - This requires a Vehicle object to be created
- Possibility to checkout a vehicle from the garage
 - This should show a checkout screen with information about the vehicle, where it's parked and the fee
 - The checkout screen should let the worker either finish the checkout or cancel it
- Possibility to search for a specific vehicle
 - Matching registration number
- Possibility to search for multiple vehicles
 - For example, by vehicle type or date parked

The program should fulfill the following requirements:

- All the functionality listed above
- Contain at least 4 different classes
- Contain the following vehicle types (may be more than the following)
 - Motorcycle
 - Car
 - Buss
 - Truck
- Allow a worker (user) to access the functionality described through a console application
- The program should not crash when asking for I/O
 - Exceptions should be handled

Tips

- Discuss and design a logical base layout as a group, make sure everyone understands your product
- Try to break the project down in smaller parts, this tends to be easier to handle
- Don't over think this project, keep it simple and as close to the requirements as possible
- Once again, rather good code and a poor menu than a good menu with poor code

Expected Techniques

- Loops
- If/else or switch cases
- Console I/O
- LINQ
- Use of correct datatypes
- Properties
- OOP
- DateTime manipulation
- Collections

Extra assignments

If you are done with the project before the assigned deadline, feel free to extend the program with the functionality listed below.

- Allow the worker to save the vehicle information to XML before shutting down the program
- Allow the worker to load vehicle information from XML
- Set a maximum allowed parking date, giving an increased fee if the vehicle is parked past that date
- Create another use case for the application "Self Service", where a user can input information about his/her vehicle, get assigned a parking spot and see directions how to get there. This should give the user a code, which he/she can use to access the checkout screen when he/she wants to checkout his/her vehicle.

Best of luck!