Copilot

Claro! Vou criar um exemplo simples de um sistema de cadastramento de carros em um estacionamento em C#. Aqui está um programa de console que emula o funcionamento de um estacionamento:

```
using System;
using System.Collections.Generic;

namespace CarParkingSystem
{
    class Program
    {
        static void Main(string[] args)
        {
            Parking parking = Parking.Instance; // Initialize the parking

            while (true)
            {
                Console.WriteLine("1. Add car");
                Console.WriteLine("2. Remove car");
                Console.WriteLine("3. Check available parking spaces");
                Console.WriteLine("4. Exit");
                Console.Write("Enter your choice: ");
                int choice = int.Parse(Console.ReadLine());

                switch (choice)
                {
                    case 1:
                        Console.Write("Enter car ID: ");
                        string carId = Console.ReadLine();
                        Console.Write("Enter car balance: ");
                        double balance = double.Parse(Console.ReadLine());
                        Console.Write("Enter car type (Passenger/Truck/Bus/Motorcycle): ");
                        CarType carType = Enum.Parse<CarType>(Console.ReadLine(), true);
                        parking.AddCar(carId, balance, carType);
                        break;
                    case 2:
                        Console.Write("Enter car ID to remove: ");
                        string carToRemove = Console.ReadLine();
                        parking.RemoveCar(carToRemove);
                        break;
                    case 3:
                        Console.WriteLine($"Available parking spaces: {parking.AvailableSpaces}");
                        break;
                    case 4:
                        Environment.Exit(0);
                        break;
                    default:
                        Console.WriteLine("Invalid choice. Try again.");
                        break;
                }
            }
        }
    }

    public enum CarType
    {
        Passenger,
        Truck,
        Bus,
        Motorcycle
    }

    public class Car
    {
        public string Id { get; set; }
        public double Balance { get; set; }
        public CarType Type { get; set; }
    }
    public class Parking
    {
        private static readonly Parking instance = new Parking();
        private List<Car> cars = new List<Car>();
        private double totalRevenue = 0;

        private Parking() { }

        public static Parking Instance => instance;

        public int AvailableSpaces => Settings.ParkingSpace - cars.Count;

        public void AddCar(string id, double balance, CarType type)
        {
            cars.Add(new Car { Id = id, Balance = balance, Type = type });
        }

        public void RemoveCar(string id)
        {
            Car car = cars.Find(c => c.Id == id);
            if (car != null)
            {
                totalRevenue += Settings.ParkingPrices[car.Type];
                cars.Remove(car);
            }
        }
    }

    public static class Settings
    {
        public static int Timeout { get; } = 3; // N seconds
        public static Dictionary<CarType, double> ParkingPrices { get; } = new Dictionary<CarType, double>
        {
            { CarType.Passenger, 3 },
            { CarType.Truck, 5 },
            { CarType.Bus, 2 },
            { CarType.Motorcycle, 1 }
        };
        public static int ParkingSpace { get; } = 50;
        public static double Fine { get; } = 1.2; // Penalty factor
    }
}
```

Este programa permite adicionar e remover carros do estacionamento, verificar o número de vagas disponíveis e calcular a receita total do estacionamento. Lembre-se de adaptar o código conforme suas necessidades específicas. 😊