Мусаев Тимур Русланович

**Отчет  
 Тема: Парковка**

**Цель работы**: разработка программы для учета автомобилей, парковок

**Задачи**:

Описать классы "Автомобиль" и "Стоянка".

Класс "Автомобиль" должен содержать информацию о машине: Марку, Модель, Цвет, Номер, Время прибытия.

Добавьте возможность добавления машин на стоянку, просмотра информации о них и их вывоз.

Предусмотреть защиту от ввода некорректных данных для всех членов всех классов

Предусмотреть повторное предложение Пользователю на ввод данных, в случае их некорректности

Предусмотреть назначение Автомобилям уникальных идентификаторов

Предусмотреть выбор Предприятия на приема и переода работников

Написать класс для сохранения состояния программы в текстовом файле, на примере стоянки:

при изменении данных перезаписывать файл,

при новом запуске программы восстанавливать данные из файла

написать класс для логирования событий в текстовый файл

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace ConsoleApp15

{

class Автомобиль

{

public string Марка { get; set; }

public string Модель { get; set; }

public string Цвет { get; set; }

public string Номер { get; set; }

public DateTime ВремяПрибытия { get; set; }

public Автомобиль(string марка, string модель, string цвет, string номер)

{

Марка = марка;

Модель = модель;

Цвет = цвет;

Номер = номер;

ВремяПрибытия = DateTime.Now;

}

public override string ToString()

{

return $"{Марка} {Модель}, Цвет: {Цвет}, Номер: {Номер}, Время прибытия: {ВремяПрибытия}";

}

}

class Стоянка

{

private List<Автомобиль> автомобили = new List<Автомобиль>();

private string файлСостояния = "состояние.txt";

private string файлЛога = "лог.txt";

public Стоянка()

{

ВосстановитьСостояние();

}

public void ПрипарковатьАвтомобиль(Автомобиль автомобиль)

{

автомобили.Add(автомобиль);

Console.WriteLine($"Автомобиль {автомобиль.Марка} {автомобиль.Модель} припаркован.");

Логировать($"прибытие автомобиля {автомобиль.Марка} {автомобиль.Модель}");

СохранитьСостояние();

}

public void ВывезтиАвтомобиль(string номер)

{

Автомобиль уезжающийАвтомобиль = автомобили.Find(а => а.Номер == номер);

if (уезжающийАвтомобиль != null)

{

автомобили.Remove(уезжающийАвтомобиль);

Console.WriteLine($"Автомобиль {уезжающийАвтомобиль.Марка} {уезжающийАвтомобиль.Модель} с номером {номер} выезжает.");

Логировать($"убытие автомобиля {уезжающийАвтомобиль.Марка} {уезжающийАвтомобиль.Модель}");

СохранитьСостояние();

}

else

{

Console.WriteLine($"Автомобиль с номером {номер} не найден на стоянке.");

}

}

public void ПросмотретьАвтомобили()

{

Console.WriteLine("Список автомобилей на стоянке:");

foreach (var автомобиль in автомобили)

{

Console.WriteLine(автомобиль);

}

}

private void СохранитьСостояние()

{

using (StreamWriter writer = new StreamWriter(файлСостояния))

{

foreach (var автомобиль in автомобили)

{

writer.WriteLine($"{автомобиль.Марка};{автомобиль.Модель};{автомобиль.Цвет};{автомобиль.Номер};{автомобиль.ВремяПрибытия}");

}

}

}

private void ВосстановитьСостояние()

{

if (File.Exists(файлСостояния))

{

автомобили.Clear();

using (StreamReader reader = new StreamReader(файлСостояния))

{

string line;

while ((line = reader.ReadLine()) != null)

{

string[] parts = line.Split(';');

if (parts.Length == 5)

{

string марка = parts[0];

string модель = parts[1];

string цвет = parts[2];

string номер = parts[3];

DateTime времяПрибытия = DateTime.Parse(parts[4]);

Автомобиль автомобиль = new Автомобиль(марка, модель, цвет, номер);

автомобиль.ВремяПрибытия = времяПрибытия;

автомобили.Add(автомобиль);

}

}

}

}

}

private void Логировать(string событие)

{

string логСобытия = $"{DateTime.Now:HH:mm dd.MM.yyyy}: {событие}";

using (StreamWriter writer = new StreamWriter(файлЛога, true))

{

writer.WriteLine(логСобытия);

}

}

}

class Program

{

static void Main()

{

Стоянка стоянка = new Стоянка();

while (true)

{

Console.WriteLine("Выберите действие:");

Console.WriteLine("1. Припарковать автомобиль");

Console.WriteLine("2. Вывезти автомобиль");

Console.WriteLine("3. Просмотреть автомобили");

Console.WriteLine("4. Выход");

int выбор = int.Parse(Console.ReadLine());

switch (выбор)

{

case 1:

Console.Write("Введите марку автомобиля: ");

string марка = Console.ReadLine();

Console.Write("Введите модель автомобиля: ");

string модель = Console.ReadLine();

Console.Write("Введите цвет автомобиля: ");

string цвет = Console.ReadLine();

Console.Write("Введите номер автомобиля: ");

string номер = Console.ReadLine();

Автомобиль новыйАвтомобиль = new Автомобиль(марка, модель, цвет, номер);

стоянка.ПрипарковатьАвтомобиль(новыйАвтомобиль);

break;

case 2:

Console.Write("Введите номер автомобиля, который нужно вывезти: ");

string номерАвтомобиля = Console.ReadLine();

стоянка.ВывезтиАвтомобиль(номерАвтомобиля);

break;

case 3:

стоянка.ПросмотретьАвтомобили();

break;

case 4:

Environment.Exit(0);

break;

default:

Console.WriteLine("Неверный выбор. Попробуйте снова.");

break;

}

}

}

}

}