**Модификаторы доступа в С#**

**Практическая работа:  
Задание:** Использование модификаторов доступа в C#

**Цель:**

Научиться применять различные модификаторы доступа для управления видимостью классов и их членов, а также понять, как это влияет на инкапсуляцию и безопасность кода.

**Описание задания:**

Создайте новый проект в Visual Studio или другой IDE, поддерживающей C#.

Создайте класс Car в пространстве имен MyApp.Models с следующими свойствами и методами:

Поле make (тип string) с модификатором доступа private.

Поле model (тип string) с модификатором доступа private.

Свойство Year (тип int) с модификатором доступа public.

Метод Set Make And Model (с модификатором доступа internal), который принимает два параметра: make и model, и устанавливает соответствующие поля.

Метод DisplayInfo (с модификатором доступа protected), который выводит информацию о машине в консоль.

Создайте класс ElectricCar, который наследует класс Car:

Добавьте поле batteryCapacity (тип double) с модификатором доступа private.

Создайте метод SetBatteryCapacity (с модификатором доступа public), который устанавливает значение batteryCapacity.

Переопределите метод DisplayInfo, чтобы он также выводил информацию о batteryCapacity.

Создайте класс Program в пространстве имен MyApp с методом Main:

Создайте экземпляр класса Car и попробуйте установить значения для полей make и model (должно быть запрещено, так как они private).

Используйте метод SetMakeAndModel для установки значений make и model.

Создайте экземпляр класса Electric Car и установите значение battery Capacity.

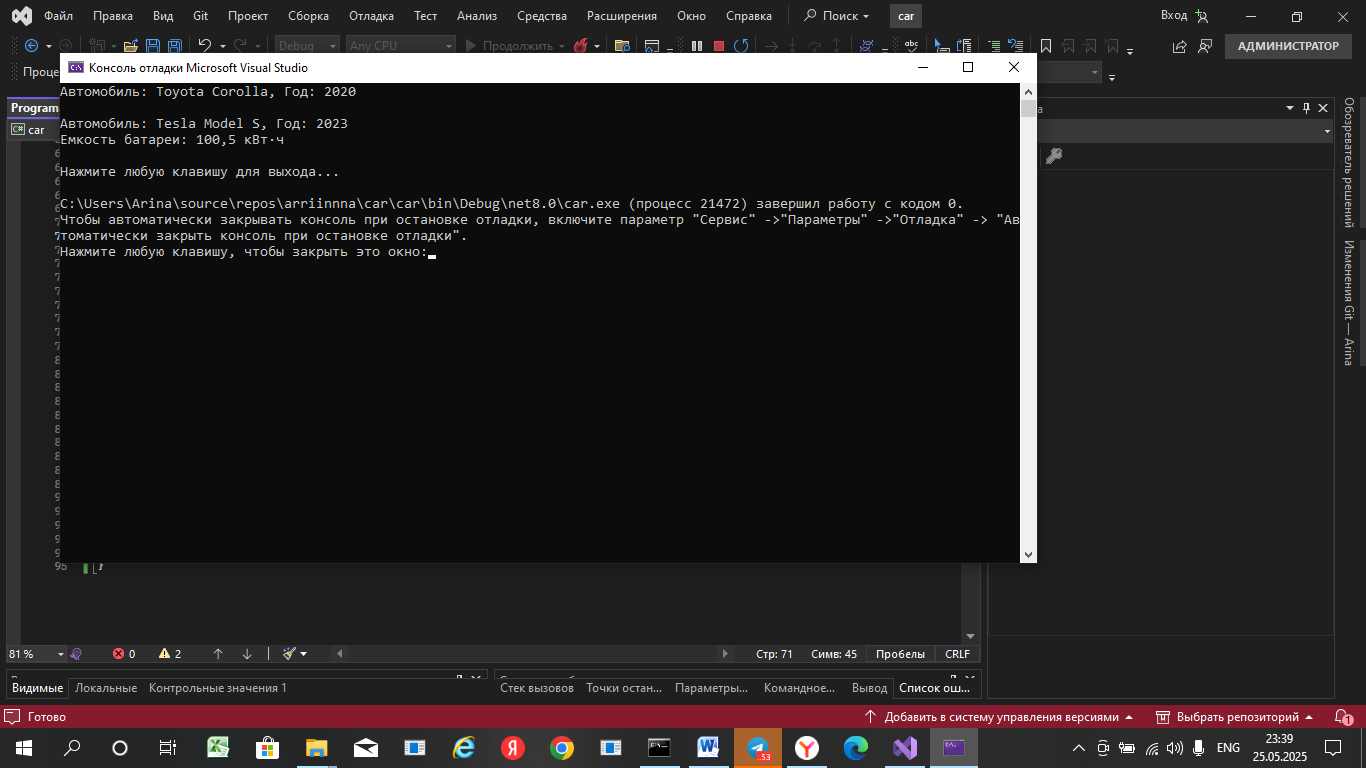
Вызовите метод DisplayInfo для обоих классов.

Добавьте комментарии к каждому классу и методу, объясняющие, какой модификатор доступа  используется и почему.

**Ожидаемый результат:**

В результате выполнения задания вы получите проект, который демонстрирует использование различных модификаторов доступа в C#. Вы научитесь управлять видимостью членов класса, что является важной частью инкапсуляции и проектирования безопасного и поддерживаемого кода.

Что у меня получилось:



using System;

namespace MyApp

{

//Класс Car демонстрирует инкапсуляцию и различные модификаторы доступа

public class Car

{

//private поля — недоступны извне

private string make;

private string model;

//public свойство — доступно откуда угодно

public int Year { get; set; }

//internal метод — доступен только в пределах текущей сборки

internal void SetMakeAndModel(string make, string model)

{

this.make = make;

this.model = model;

}

//protected метод — доступен только в этом классе и унаследованных классах

protected virtual void DisplayInfo()

{

Console.WriteLine($"Автомобиль: {make} {model}, Год: {Year}");

}

//public метод для вызова защищенного DisplayInfo извне

public void ShowInfo()

{

DisplayInfo();

}

}

//Класс ElectricCar наследует Car и добавляет собственное поведение

public class ElectricCar : Car

{

//private поле — емкость батареи

private double batteryCapacity;

//public метод — установка значения батареи

public void SetBatteryCapacity(double capacity)

{

batteryCapacity = capacity;

}

//protected override — переопределяет метод родителя и добавляет вывод батареи

protected override void DisplayInfo()

{

base.DisplayInfo(); // вызывает родительский метод

Console.WriteLine($"Емкость батареи: {batteryCapacity} кВт·ч");

}

//public метод для вызова защищенного DisplayInfo извне

public new void ShowInfo()

{

DisplayInfo();

}

}

//Главный класс с точкой входа

class Program

{

static void Main()

{

//Создание экземпляра Car

Car car = new Car();

car.Year = 2020;

//Попытка прямого доступа к make и model будет ошибкой — они private

//car.make = "Toyota"; // Ошибка

//car.model = "Corolla"; // Ошибка

//Используем internal метод для установки

car.SetMakeAndModel("Toyota", "Corolla");

//Вывод информации

car.ShowInfo();

Console.WriteLine();

//Создание экземпляра ElectricCar

ElectricCar electricCar = new ElectricCar();

electricCar.Year = 2023;

electricCar.SetMakeAndModel("Tesla", "Model S");

electricCar.SetBatteryCapacity(100.5);

//Вывод информации

electricCar.ShowInfo();

Console.WriteLine("\nНажмите любую клавишу для выхода...");

Console.ReadKey();

}

}

}