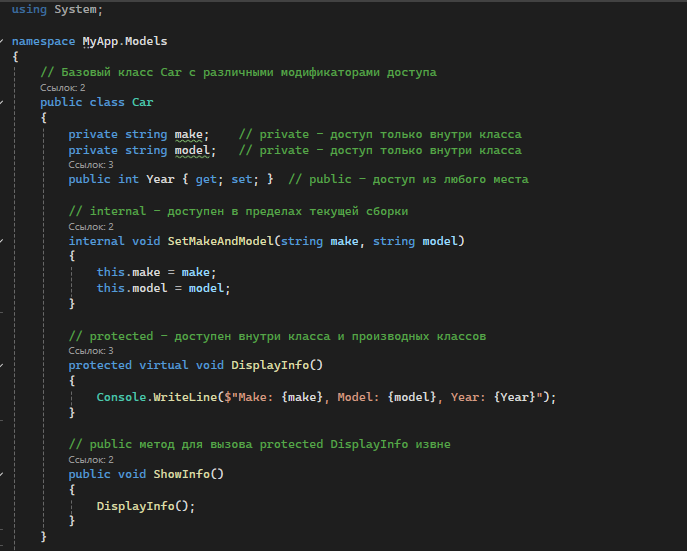
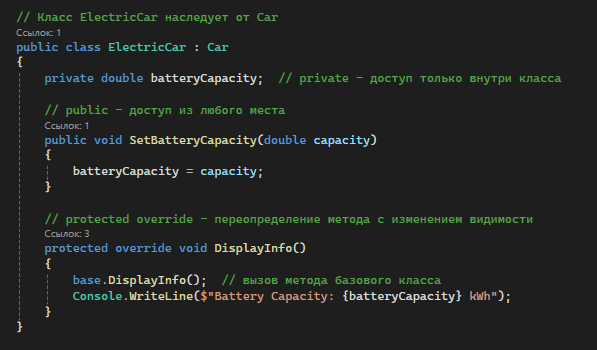
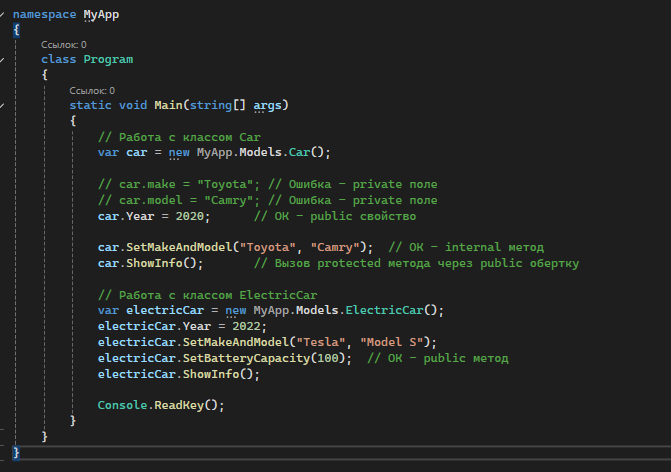
**Практическая работа №2 « Использование модификаторов доступа в C#».**

**Выполнил студент группы Юнязов Богдан Денисович.**

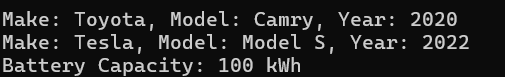
**Практическое задание.**







**Результат:**



**Теоретические вопросы:**

**Вопрос 1: Какие модификаторы доступа есть в C#?**

**Ответ:**

В C# существуют следующие модификаторы доступа:

public - доступен из любого места

private - доступен только внутри класса

protected - доступен внутри класса и производных классов

internal - доступен в пределах текущей сборки

protected internal - доступен в пределах текущей сборки и в производных классах в других сборках

private protected - доступен в пределах текущей сборки только в производных классах

**Вопрос 2: Вам надо определить в классе переменную, которая должна быть доступна из любого места в текущем проекте. Какой модификатор (или модификаторы, если их несколько) вы будете использовать?**

**Ответ:**

Для доступа из любого места в текущем проекте можно использовать:

internal - если нужен доступ только в текущей сборке

public - если нужен доступ из других сборок

**Вопрос 3: В чем различие между модификаторами protected и private protected?**

**Ответ:**

protected: члены доступны внутри класса и в производных классах, независимо от сборки

private protected: члены доступны только в производных классах, которые находятся в той же сборке

**Вопрос 4: Если классы и члены класса не имеют никаких модификаторов, какие модификаторы доступа к ним применяются по умолчанию?**

**Ответ:**

Для классов и структур: internal

Для членов класса: private

**Вопрос 5: Что выведет на консоль следующая программа и почему?**

**Ответ:**

Программа не скомпилируется, так как поле name имеет модификатор доступа по умолчанию (private), и попытка доступа к нему из другого класса (Program) невозможна.

**Ответы на вопросы по свойствам:**

**Вопрос 1: Что будет выведено на консоль?**

**Ответ:**

Выведет 15, так как set-тер свойства Age пустой и не изменяет значение приватного поля age.

**Вопрос 2: Что будет выведено на консоль и почему?**

**Ответ:**

Выведет Tom, так как свойство Name имеет модификатор internal (доступно в пределах сборки), и мы явно устанавливаем значение "Tom".

**Вопрос 3: Что будет выведено на консоль и почему?**

**Ответ:**

Выведет Tom, так как хотя get-тер свойства Name имеет модификатор internal, это не мешает установке значения и его чтению в пределах той же сборки. Модификатор internal для get-тера означает, что чтение свойства будет доступно только в пределах текущей сборки.