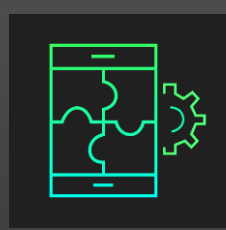




Модификаторы доступа



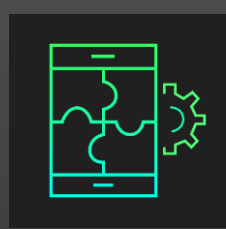
Модификаторы доступа — это ключевые слова в языке программирования C#, которые контролируют уровень доступа к классам, членам классов (полям, свойствам, методам) и интерфейсам. Они определяют, откуда можно обращаться к данным и методам, что позволяет реализовать инкапсуляцию и защиту данных.

Основные модификаторы доступа:

- **file:** доступны только из того самого файла, где они определены.
- **default:** зависит от элемента (если class то как и internal)
- **public:** доступно везде.
- **private:** доступно только внутри класса.
- **protected:** доступно в классе и наследниках.
- **internal:** доступно в пределах той же сборки.
- **protected internal:** доступно в той же сборке и в наследниках.
- **private protected:** доступно в классе и наследниках, но только в пределах той же сборки.



Модификаторы доступа



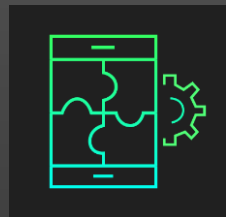
Зачем они нужны ?

Основная цель модификаторов доступа — управлять видимостью и доступом к элементам программного кода, чтобы:

- 1. Скрыть внутреннюю реализацию:** Защитить внутренние детали класса от внешнего доступа, что уменьшает связанность и повышает безопасность кода.
- 2. Упрощение поддержки и тестирования:** Изменение внутренних деталей класса не затрагивает код, который зависит от этого класса, если доступ к внутренним членам ограничен.
- 3. Определение интерфейса класса:** Ясно указать, какие члены класса предназначены для использования другими частями программы, а какие являются внутренними реализациями.



Модификаторы доступа



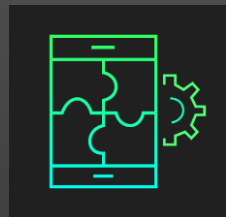
Public

Элементы класса доступны из любого места

```
public class Car {  
    public string Model { get; set; } // Доступно везде  
    public void Start() { /* код запуска */ }  
}
```



Модификаторы доступа



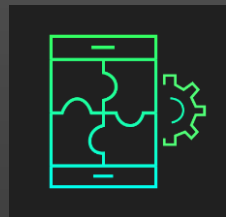
Private

Элементы класса доступны только внутри самого класса. Не доступны из других классов

```
public class Car {  
  
    private string engineNumber; // Доступно только внутри Car  
    private void initializeEngine() { /* код инициализации */ }  
}
```



Модификаторы доступа



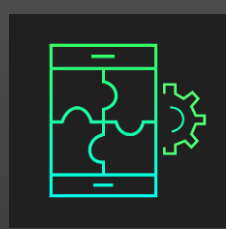
Protected

Элементы класса доступны внутри самого класса
и в производных классах (наследниках)

```
public class Vehicle {  
    protected int speed; // Доступно в Vehicle наследниках  
    protected void Accelerate() { /* код для ускорения */ }  
}  
  
public class Car : Vehicle {  
    public void IncreaseSpeed(int increment) {  
        speed += increment; // Доступно здесь  
    }  
}
```



Модификаторы доступа



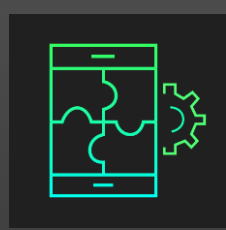
Internal

Элементы класса доступны только в пределах той же сборки (assembly). Не доступны из других сборок

```
internal class DatabaseConnection {  
    internal void Connect(); { /* код для подключения */ }  
}
```



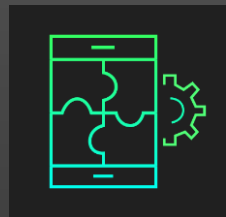
Модификаторы доступа



Protected Internal

Элементы класса доступны как в пределах той же сборки, так и в производных классах

```
public class Vehicle {  
    protected internal int maxSpeed; //Доступно в той же сборке и в наследниках }  
    protected internal void DisplayMaxSpeed() { /* код отображения макс. скорости */  
}  
}
```



Модификаторы доступа

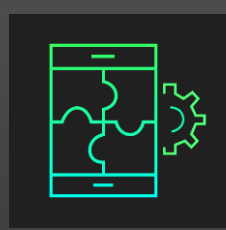
Private Protected

Элементы доступны только в пределах своего класса и в производных классах, но только в пределах той же сборки

```
public class Vehicle {  
    private protected int fuelCapacity; // Доступно в  
    Vehicle и наследниках той же сборки  
    private protected void Refuel() { /* код заправки */ }  
}  
  
public class Car : Vehicle {  
    public void SetFuelCapacity(int capacity) {  
        fuelCapacity = capacity; // Доступно здесь  
    }  
}
```




Модификаторы доступа



Default

В C# по умолчанию применяется модификатор доступа Default, если вы не укажете явно какой-либо из них

В зависимости от типа элемента (класс, интерфейс, структура и т.д.) применяются разные модификаторы.

public – для interface

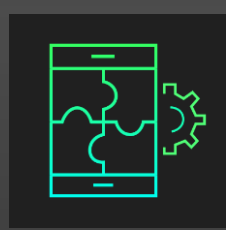
Internal – для class, struct и delegate

private – для методов, полей, свойств,
вложенных классов



Модификаторы доступа

File



Модификатор доступа, который позволяет членам класса быть доступными только для других классов, находящихся в том же исходном файле. Элементы, помеченные этим модификатором, не доступны за пределами данного файла, даже если классы находятся в одной сборке.



Модификаторы доступа

Таблица доступа модификаторов

Модификатор доступа	Класс (собственный)	Наследуемый класс (в другой сборке)	Тот же класс (в другой сборке)	Тот же класс (в той же сборке)	Другие классы (в той же сборке)	Другие классы (в другой сборке)
public	Да	Да	Да	Да	Да	Да
private	Да	Нет	Нет	Да	Нет	Нет
protected	Да	Да	Нет	Да	Нет	Нет
internal	Да	Нет	Нет	Да	Да	Нет
protected internal	Да	Да	Нет	Да	Да	Нет
private protected	Да	Нет	Нет	Да	Нет	Нет
file	Да	Нет	Нет	Да	Нет	Нет

- **Класс (собственный):** Означает доступ к модификаторам внутри самого класса.
- **Наследуемый класс (в другой сборке):** Проверяет возможность доступа к элементам из производного класса в другой сборке.
- **Тот же класс (в другой сборке):** Проверяет доступ к элементам того же класса, но из другой сборки.
- **Тот же класс (в той же сборке):** Проверяет доступ к элементам того же класса в одной сборке.
- **Другие классы (в той же сборке):** Проверяет доступ к элементам из других классов в одной сборке.
- **Другие классы (в другой сборке):** Проверяет доступ к элементам из других классов в другой сборке.