Конструкторы

Конструктор с параметрами, без параметров, по умолчанию, приватный/закрытый, статический

Статический вызывается единожды при обращении к классу. Статический конструктор используется для инициализации любых статических данных или для выполнения определенного действия, которое требуется выполнить только один раз.

Закрытые конструкторы используются, чтобы не допустить создания экземпляров класса при отсутствии полей или методов экземпляра

Конструктор по умолчанию в самом конструкторе содержит значение

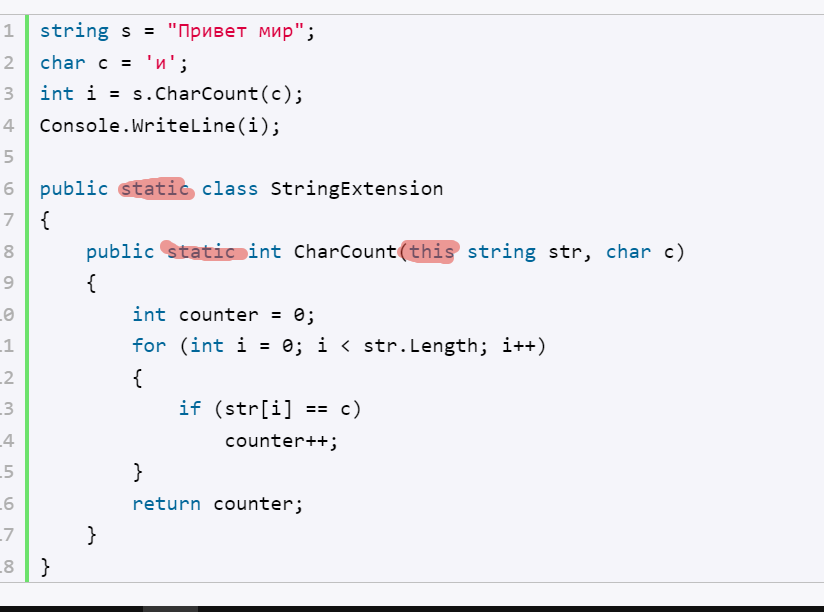
Статические

Статические классы содержат только статические члены (поля, конструкторы, свойства, методы).

Статические классы нельзя создать экземпляр.

Методы расширения

Методы расширения позволяют добавлять новые методы в уже существующие типы без создания нового производного класса



Ссылочный типы

Типы значений:

* Целочисленные типы (byte, sbyte, short, ushort, int, uint, long, ulong)
* Типы с плавающей запятой (float, double)
* Тип decimal
* Тип bool
* Тип char
* Перечисления enum
* Структуры (struct)

Ссылочные типы:

* Тип object
* Тип string
* Классы (class)
* Интерфейсы (interface)
* Делегаты (delegate)

Ограничения на обобщения

void SendMessage<T>(T message) **where** T: Message

Ограничение на наследование от объекта, от интерфейса, от структура, new()(ограничение на конструктор без параметров)

Ref и Out параметры

Ref (входной параметр) при изменении в методе параметра то значение переменной так же меняется

Out (выходной параметр) обязательно должно быть присвоено значение, можно определить несколько параметров out, определяется в сигнатуре метода

Action и Func

Это делегаты

Делегат **Action** представляет некоторое действие, которое ничего не возвращает, то есть в качестве возвращаемого типа имеет тип void. Как правило, этот делегат передается в качестве параметра метода и предусматривает вызов определенных действий в ответ на произошедшие действия. До 16 параметров

Func возвращает результат действия и может принимать параметры. Последним параметром указывается тип возвращаемого значения

Делегаты

Делегаты представляют такие объекты, которые указывают на методы. То есть делегаты - это указатели на методы и с помощью делегатов мы можем вызвать данные методы.

События

События сигнализируют системе о том, что произошло определенное действие. И если нам надо отследить эти действия, то как раз мы можем применять события.

События объявляются в классе с помощью ключевого слова **event**, после которого указывается тип делегата, который представляет событие:

|  |  |
| --- | --- |
| 1  2 | delegate void AccountHandler(string message);  event AccountHandler Notify; |

**3**Notify?.Invoke($"На счет поступило: {sum}");   // 2.Вызов события

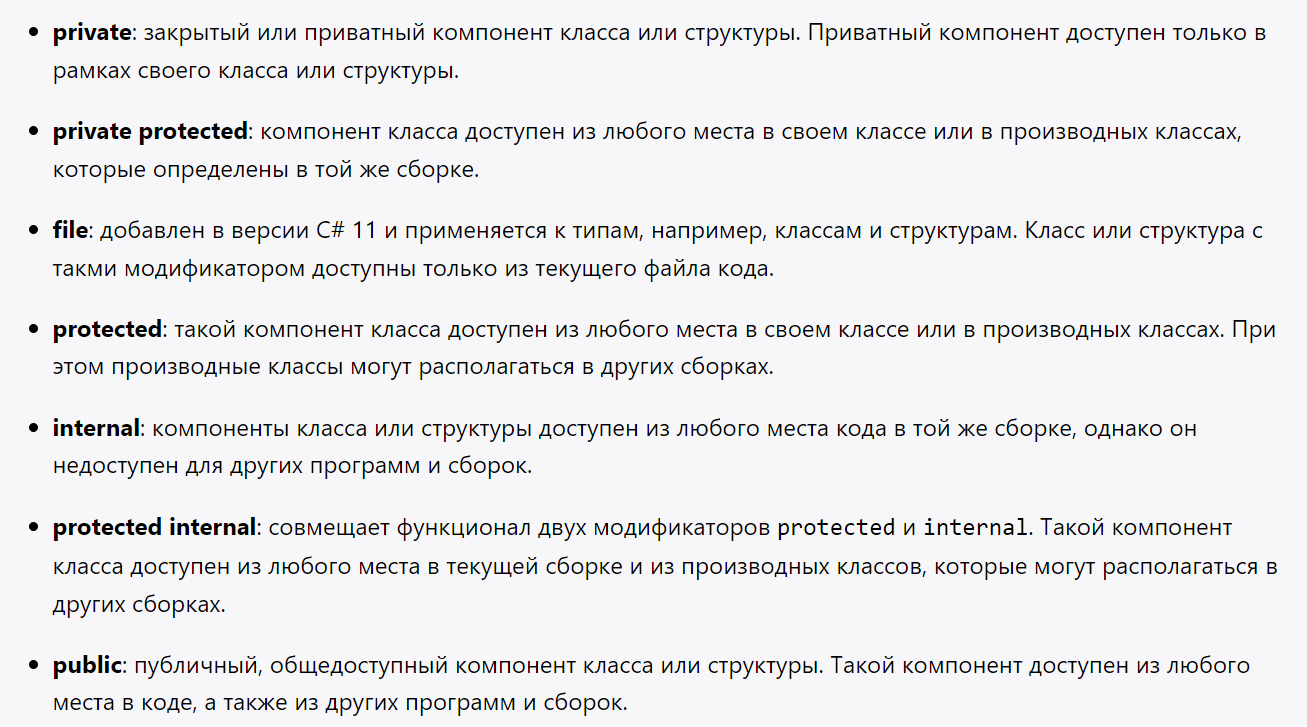
Кортеж

Кортеж позволяет хранить до 8 различных типов данных

Объявляются **var – неявное определение**

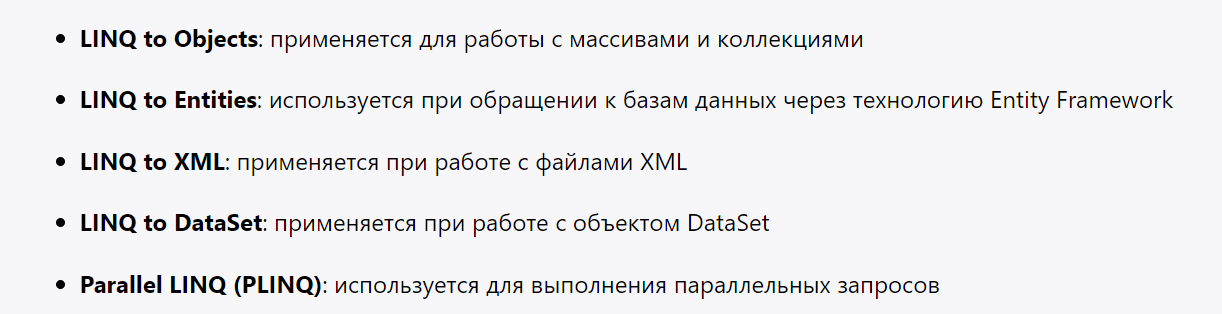
(string, int, double) person = ("Tom", 25, 81.23); -- явное определения

В абстрактном классе можно писать не абстрактные члены и от абстрактного класса наследуемся единожды а интерфейс поддерживают множественное наследование.



Sealead запрещает наследование

При отложенном выполнении LINQ-выражение не выполняется, пока не будет произведена итерация или перебор по выборке, например, в цикле foreach.



Анонимная функция -- не именованный блок кода. Лямда-варажение -- краткая запись анонимной функции.

IEnumerator позволяет перебирать элементы коллекции и предоставляет методы для перемещения по коллекции и доступа к текущему элементу.

IEnumerable определяет метод GetEnumerator(), который возвращает объект типа IEnumerator для возможности итерации по коллекции.

Необобщенные коллекции Специальные коллекции Поразрядная коллекция Обобщенные коллекции Параллельные коллекции