

- значимые типы

1) числовые

1.1 числа со знаком (sbyte, short, int, long)

1.2 числа без знака (byte, ushort, uint, ulong)

1.3 дробные числа (float, double, decimal)

2) логические (bool)

3) символьные (char)

- ссылочные типы

1) строки (string)

2) объекты (object)

-пользовательские

Класс является основным пользовательским типом. Синтаксис объявления класса в С# следующий:

```
<модификаторы> class <имя класса>
{
    [<элементы класса>]
}
```

Допустимые элементы класса

1. Поле. Синтаксис объявления поля класса совпадает с синтаксисом оператора объявления переменной¹ (тип поля должен всегда быть указан явно, использование **var** не допускается).

2. Константа. Синтаксис объявления константы в классе аналогичен синтаксису, применяемому при объявлении константы в теле метода.

Следующие элементы класса будут подробно рассмотрены в дальнейшем.

3. Метод. Методы описывают функциональность класса.

4. Свойство. Свойства класса призваны предоставить защищенный доступ к полям.

5. Индексатор. Индексатор – это свойство-коллекция, отдельный элемент которого доступен по индексу.

6. Конструктор. Задача конструктора – начальная инициализация объекта или класса.

7. Финализатор. Финализатор автоматически вызывается сборщиком мусора и содержит завершающий код для объекта.

8. Событие. События представляют собой механизм рассылки уведомлений различным объектам.

9. Операция. Язык С# допускает перегрузку некоторых операций для объектов класса.

10. Вложенный пользовательский тип. Описание класса может содержать описание другого пользовательского типа – класса, структуры, перечисления, интерфейса, делегата. Обычно вложенные типы выполняют вспомогательные функции и явно вне основного типа не используются.

модификаторы доступа:

- **private.** Элемент с данным модификатором доступен только в том типе, в котором определен. Например, поле доступно только в содержащем его классе.
- **protected.** Элемент виден в типе, в котором определен, и в наследниках этого типа (даже если наследники расположены в других сборках).
- **internal.** Элемент доступен без ограничений, но только в той сборке, где описан.
- **protected internal.** Комбинация модификаторов **protected** и **internal**. Элемент виден в содержащей его сборке без ограничений, а вне сборки – только в наследниках типа (т.е. **protected** или **internal**²).
- **public.** Элемент доступен без ограничений как в той сборке, где описан, так и в других сборках, к которым подключается сборка с элементом.

¹ Как правило, идентификаторы полей снабжаются неким оговоренным префиксом.

² В CLR имеется модификатор доступа, соответствующий **protected** и **internal**. При помощи языка С# такой уровень доступа описать нельзя.