

В языке C# класс может содержать методы с одинаковыми именами, но различными сигнатурами. Вот общая структура объявления класса, затем идут примеры.

“мод. видимости” “параметры” “возвращаемое значение” “название функции” (“параметры”);

```
public int MethodReturn(int a) { return i; }  
public string MethodReturn(string a) { return i.ToString(); }
```

В C# модификаторы видимости указываются перед каждым элементом класса, не надо путать с C++, где они задаются в виде секций с двоеточием, например «public:».

В C# используются следующие модификаторы видимости:

- `public` – элемент виден и в классе и снаружи класса;
- `private` – элемент виден только в классе;
- `protected` – элемент виден только в классе и наследуемых классах;
- `internal` – элемент виден в текущей сборке;
- `protected internal` – элемент виден в наследуемых классах текущей сборки.

Для реализации абстрактного (`abstract`) или виртуального (`virtual`) метода в наследуемом классе используется параметр `override`. Если метод наследуется из интерфейса, этот параметр не нужен.