**Классы (Руководство по программированию на C#)**

Класс — это логическая структура, позволяющая создавать свои собственные пользовательские типы путем группирования переменных других типов, методов и событий. Класс подобен чертежу. Он определяет данные и поведение типа. Если класс не объявлен статическим, то клиентский код может его использовать, создав *объекты* или *экземпляры*, назначенные переменной. Переменная остается в памяти, пока все ссылки на нее не выйдут из области видимости. В это время среда CLR помечает ее пригодной для сборщика мусора. Если класс объявляется [статическим](http://msdn.microsoft.com/ru-ru/library/98f28cdx.aspx), то в памяти остается только одна копия и клиентский код может получить к ней доступ только посредством самого класса, а не *переменной экземпляра*. Для получения дополнительной информации см. [Статические классы и члены статических классов (Руководство по программированию в C#)](http://msdn.microsoft.com/ru-ru/library/79b3xss3.aspx).

В отличие от структур классы поддерживают *наследование*, фундаментальную характеристику объектно-ориентированного программирования. Для получения дополнительной информации см. [Наследование (Руководство по программированию на C#)](http://msdn.microsoft.com/ru-ru/library/ms173149.aspx).

[Объявление классов](javascript:void(0))

Классы объявляются с помощью ключевого слова [class](http://msdn.microsoft.com/ru-ru/library/0b0thckt.aspx), как показано в следующем примере.

public class Customer

{

//Fields, properties, methods and events go here...

}

Ключевому слову **class** предшествует уровень доступа. Поскольку в данном случае используется [public](http://msdn.microsoft.com/ru-ru/library/yzh058ae.aspx), любой может создавать объекты из этого класса.Имя класса указывается после ключевого слова **class**. Оставшаяся часть определения является телом класса, в котором задаются данные и поведение.Поля, свойства, методы и события в классе обозначаются термином *члены класса*.

[Создание объектов](javascript:void(0))

Класс и объект — это разные вещи, хотя в некоторых случаях они взаимозаменяемы. Класс определяет тип объекта, но не сам объект. Объект — это конкретная сущность, основанная на классе и иногда называемая экземпляром класса.

Объекты можно создавать с помощью ключевого слова [new](http://msdn.microsoft.com/ru-ru/library/51y09td4.aspx), за которым следует имя класса, на котором будет основан объект:

Customer object1 = new Customer();

При создании экземпляра класса ссылка на этот объект передается программисту. В предыдущем примере object1 является ссылкой на объект, основанный на Customer. Эта ссылка указывает на новый объект, но не содержит данные этого объекта. Фактически, можно создать ссылку на объект без создания самого объекта:

Customer object2;

Создание таких ссылок, которые не указывают на объект, не рекомендуется, так как попытка доступа к объекту по такой ссылке приведет к сбою во время выполнения. Однако такую ссылку можно сделать указывающей на объект, создав новый объект или назначив ее существующему объекту:

Customer object3 = new Customer();

Customer object4 = object3;

В данном коде создаются две ссылки на объекты, которые указывают на один объект. Поэтому любые изменения объекта, выполненные посредствомobject3, будут видны при последующем использовании object4. Поскольку на объекты, основанные на классах, указывают ссылки, классы называют ссылочными типами.

**Объект. Теоретические сведения** ключевое слово new отвечает за вычисление правильного количества байт для указанного объекта и выделение досточного объема памяти из управляемой кучи. Объектные переменные представляют собой ссылки на объекты в памяти, а не сами объекты.  
Принцип разделения ответственности: один тип работает как объект приложения (определяет метод Main()) и множество других типов, составляющих все приложение.  
При определении ЛОКАЛЬНЫХ переменных присвоение начальных значений перед использованием ОБЯЗАТЕЛЬНО, т.к. они не получают значений по умолчанию.  
Static методы могут быть вызваны на уравне класса, а не на уравне экземпляра. Могут использовать только статические данные. Адреса объектов определяются во время выполнения, а значение const полей устанавливается на этапе компиляции.

|  |  |
| --- | --- |
| **Элемент** | **Описание** |
| [Поля](https://msdn.microsoft.com/ru-ru/library/ms173118.aspx) | Поля являются переменными, объявленными в области класса. Поле может иметь встроенный числовой тип или быть экземпляром другого класса. Например, в классе календаря может быть поле, содержащее текущую дату. |
| [Константы](https://msdn.microsoft.com/ru-ru/library/ms173119.aspx) | Константы — это поля или свойства, значения которых устанавливаются во время компиляции и не изменяются. |
| [Свойства](https://msdn.microsoft.com/ru-ru/library/x9fsa0sw.aspx) | Свойства — это методы класса. Доступ к ним осуществляется так же, как если бы они были полями этого класса. Свойство может защитить поле класса от изменений (независимо от объекта). |
| [Методы](https://msdn.microsoft.com/ru-ru/library/ms173114.aspx) | Методы определяют действия, которые может выполнить класс. Методы могут получать параметры, предоставляющие входные данные, и возвращать выходные данные посредством параметров. Также методы могут возвращать значения напрямую, без использования параметров. |
| [События](https://msdn.microsoft.com/ru-ru/library/awbftdfh.aspx) | События предоставляют другим объектам уведомления о различных случаях, таких как нажатие кнопки или успешное выполнение метода. События определяются и переключаются с помощью делегатов. |
| [Операторы](https://msdn.microsoft.com/ru-ru/library/ms173145.aspx) | Перегруженные операторы рассматриваются как члены класса. При перегрузке оператора его следует определить как открытый статический метод в классе. Предопределенные операторы (**+**, **\***, **<** и т. д.) не считаются членами. Для получения дополнительной информации см. [Перегружаемые операторы (Руководство по программированию в C#)](https://msdn.microsoft.com/ru-ru/library/8edha89s.aspx). |
| [Индексаторы](https://msdn.microsoft.com/ru-ru/library/6x16t2tx.aspx) | Индексаторы позволяют индексировать объекты аналогично массивам. |
| [Конструкторы](https://msdn.microsoft.com/ru-ru/library/ace5hbzh.aspx) | Конструкторы — это методы классов, вызываемые при создании объекта заданного типа. Зачастую они используются для инициализации данных объекта. |
| [Деструкторы](https://msdn.microsoft.com/ru-ru/library/66x5fx1b.aspx) | Деструкторы очень редко используются в C#. Они являются методами, вызываемыми средой выполнения, когда объект нужно удалить из памяти. Деструкторы обычно применяются для правильной обработке ресурсов, которые должны быть высвобождены. |
| [Вложенные типы](https://msdn.microsoft.com/ru-ru/library/ms173120.aspx) | Вложенными типами являются типы, объявленные в другом типе. Вложенные типы часто применяются для описания объектов, использующихся только типами, в которых эти объекты находятся. |