

Автоопределяемые свойства. Приведите пример задания автоопределяемого свойства.

В языке C# в классах не принято использовать `public` переменные, поэтому вместо них употребляются автоопределяемые свойства.

В C# 3.0 и более поздних версиях автоопределяемые свойства делают объявление свойств более лаконичным, когда в методах доступа к свойствам не требуется дополнительная логика. Они также позволяют клиентскому коду создавать объекты. При объявлении свойства компилятор создает закрытое анонимное резервное поле, которое может быть доступно только через методы доступа `get` и `set` свойства.

Свойства позволяют придавать программам на языке C# дополнительную гибкость, особенно при модификации кода. Например, если изменились правила вычисления какой-либо переменной (допустим, нужно возвращать значение переменной, умноженной на 2), то можно модифицировать `get`-аксессор не внося изменений в вызывающий код. Если же при присвоении переменной нужно производить дополнительный контроль (допустим, целочисленной переменной можно присваивать значения только от 1 до 1000), его можно поместить в `set`-аксессор (если условие не выполняется, то генерируется исключение) не внося изменений в вызывающий код. Поэтому в языке C# принято объявлять `public`-переменные класса как автоопределяемые свойства, а при необходимости свойство может быть переписано в полной форме с расширением кода аксессоров.

В версии C# 6 появились новые возможности работы со свойствами. При объявлении свойства возможна его непосредственная инициализация:

```
public string String1 { get; set; } = "строка 1";
```

Возможно объявление автоопределяемых свойств, предназначенных только для чтения. Такие свойства могут быть инициализированы в конструкторе класса или непосредственно при объявлении свойства.

```
public int Int1 { get; } = 333;  
public int Int2 { get; }  
/// Конструктор класса ///  
public PropertyExample()  
{ this.Int2 = 45; }
```

Попытка изменить такое свойство, например, в методе класса приводит к ошибке:

```
public void Method1()  
{  
    //Ошибка  
    this.Int1 = 123;  
}
```

Компилятор выдает следующий текст ошибки: «Невозможно присвоить значение свойству или индексатору "PropertyExample.Int1" — доступ только для чтения».