

В начале стоит определиться что такое snippet и для чего они нужны.

Фрагменты кода IntelliSense (они же snippet-ы) — это предварительно созданные готовые части кода, которые можно вставить в приложение в Visual Studio. Фрагменты кода повышают производительность за счет сокращения времени, затрачиваемого на ввод повторяющихся частей кода и поиск примеров. С помощью схемы XML фрагментов кода IntelliSense вы можете создавать собственные фрагменты и добавлять их к фрагментам, которые уже есть в Visual Studio.

Проще говоря, snippet – это шаблон (заготовка) кода, например определение свойства, определение метода, определение класса и т.д. При вставке фрагмента кода, например Exception, IntelliSense позволяет не только получить шаблон, но и автоматически выполняет изменение однотипных элементов, например имя класса.

Как работать со сниппетами? Для этого доступно несколько вариантов:

1. набрать известное имя сниппета и нажать tab;
2. нажать сочетание клавиш Ctrl+k+x для вывода списка всех зарегистрированных сниппетов;
3. использовать контекстное меню, выбрав вставить фрагмент или разметить фрагмент.

Первый вариант наиболее удобен (при наличии знаний о snippet-ax IntelliSense): можно набрать, к примеру «prop», нажать tab и мгновенно получить шаблонное определение свойства класса. Вполне естественно, что такой механизм чрезвычайно полезен. Он экономит время, позволяя определять заранее отформатированный код с возможностью удобной кастомизации. Возможно, это никак не повлияет на работу вашей программы, но добавит вам некоторую порцию удовольствия от процесса кодирования. Еще больше пользы можно получить делая свои сниппеты.

Где расположены snippet-ы?

Расположение сниппетов для языка C#, поставляемых в составе Visual Studio:

```
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC#Snippets\1049\Visual C#
```

Расположение сниппетов, разрабатываемых самостоятельно, для языка C# :

%USERPROFILE%\Documents\Visual Studio 2015\Code Snippets\Visual C#\My Code Snippets

Создание снипетов

В составе Visual Studio при разработке консольного приложения есть snippet для формирования строки Console.WriteLine(), но при этом отсутствует snippet например для публикации Console.ReadLine(). Чтобы исправить это досадное недоразумение, в качестве примера, создадим файл cr.snippet со следующим кодом:

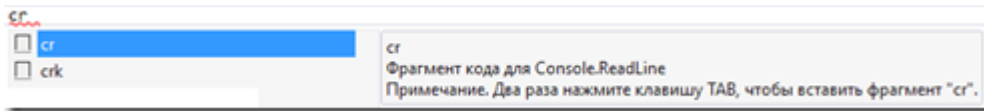
```
1  <CodeSnippets xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
2    <CodeSnippet Format="1.0.0">
3      <Header>
4        <Title>cr</Title>
5        <Shortcut>cr</Shortcut>
6        <Description>Фрагмент кода для Console.ReadLine</Description>
7        <Author>Сергей</Author>
8        <SnippetTypes>
9          <SnippetType>Expansion</SnippetType>
10       </SnippetTypes>
11     </Header>
12     <Snippet>
13       <Declarations>
14         <Literal Editable="false">
15           <ID>SystemConsole</ID>
16 <Function>SimpleTypeName(global::System.Console)</Function>
17         </Literal>
18       </Declarations>
19       <Code
20 Language="csharp"><![CDATA[\$SystemConsole\$.ReadLine\(\$end\$\);]]></Code>
21     </Snippet>
22   </CodeSnippet>
23 </CodeSnippets>
```

Интеграция снипетов

Для интеграции разработанного фрагмента кода в среду Visual Studio можно поступить одним из следующих способов.

1. Поместить созданный файл файл в папку My Code Snippets и Visual Studio мгновенно отреагирует на его появление.
2. Перейти в главное меню Сервис -> Диспетчер фрагментов кода, в открывшемся

окне указать в какой раздел сохранить snippet и нажав кнопку Импорт. После этого можно набрать сочетание `cr` в редакторе Visual Studio и нажать `tab`:



Таким же образом был создан snippet для `Console.ReadKey()` который на скриншоте выше называется `crk`.

Немного опишем структуру снипетов

Итак, что же представляет собой описание данного снипета. Основные тэги следующие: `Header`, который содержит описательную часть, и `Snippet`, который содержит логику.

Более детальное описание дочерних тэгов `Header`:

Наименование	Описание
<code><Title></code>	Задаёт заголовок фрагмента кода. Заголовок, отображается в меню выбора фрагментов кода и в описании фрагмента в Диспетчере фрагментов кода.
<code><Shortcut></code>	Задаёт краткий текст, используемый для вставки фрагмента кода. Текстовое значение элемента <code>Shortcut</code> может содержать только буквенно-цифровые знаки, дефисы (-) и символы подчеркивания (_).
<code><Description></code>	Задаёт описательные сведения о содержимом фрагмента кода IntelliSense
<code><Author></code>	Задаёт имя автора фрагмента.
<code><SnippetType></code>	Задаёт способ вставки фрагмента кода в Visual Studio. <ul style="list-style-type: none"> • SurroundsWith - позволяет расположить фрагмент кода вокруг выбранного отрывка кода, подобно <code>#region</code>. • Expansion - позволяет вставить фрагмент кода в то место, где находится курсор. • Refactoring - указывает, что фрагмент кода используется во время рефакторинга Visual C#.

Более детальное описание дочерних тэгов `Snippet`:

Наименование	Описание
<code><Declarations></code>	Задаёт литералы и объекты, которые составляют части фрагмента кода, доступные для изменения.

<Literal>	Задаёт литералы фрагмента кода, доступные для редактирования. Элемент Literal может содержать любое число элементов Declarations, включая ноль.
<ID>	Задаёт уникальный идентификатор элемента Literal или Object. Никакие два литерала или объекта в одном фрагменте кода не могут иметь одинаковое текстовое значение в элементах ID.
<Function>	Задаёт функцию для выполнения при получении литералом фокуса в Visual Studio. Элемент Function поддерживается только фрагментами кода Visual C#.
Code	Является контейнером для коротких блоков кода. Language - отражает язык фрагмента кода.

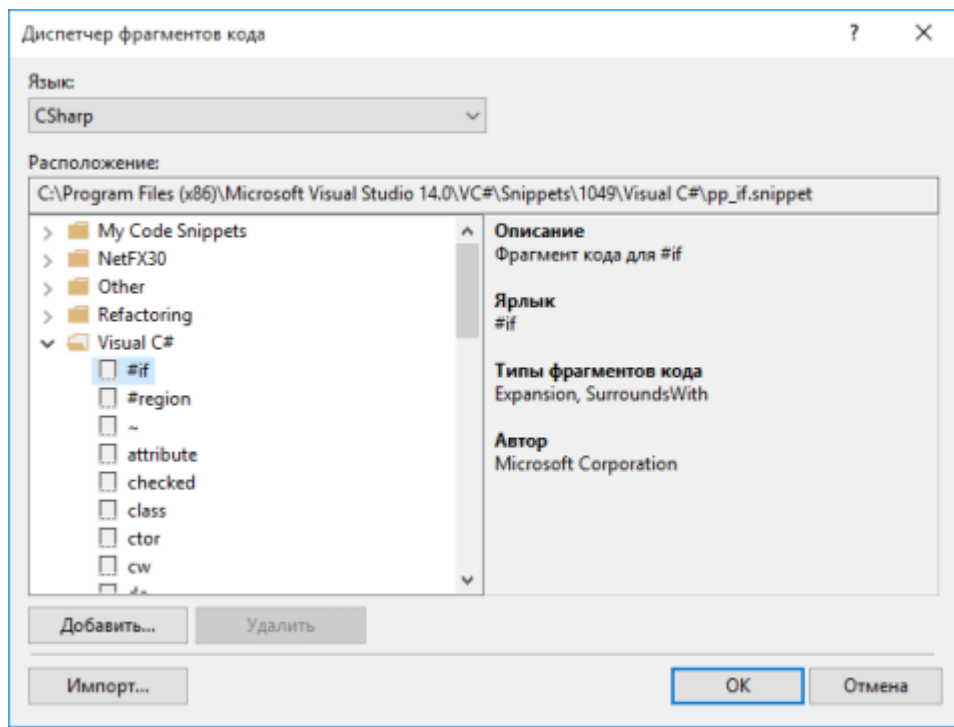
В подразделе Snippet декларируются литералы, значение которых можно будет редактировать при использовании сниппета, и собственно код сниппета. Обратите внимание на то, как определяется литерал: указывается его идентификатор, подсказка и значение по умолчанию. Идентификатор литерала затем используется в теле кода сниппета. Для выделения литерала, он заключается между двумя знаками доллара «\$».

Тэг Code содержит кроме определения кода сниппета еще и несколько атрибутов: Language, который указывает на целевой язык сниппета и может принимать значения VB, CSharp, VJSharp, или XML; Delimiter, который позволяет переопределить знак доллара на другой символ; и Kind, который позволяет указать сферу применения сниппета и принимает значения «method body», «method decl», «type decl», file, или any.

В заключение

Для того чтобы ознакомиться со всеми доступными фрагментами кода Visual Studio, необходимо перейти в главное меню Сервис -> Диспетчер фрагментов кода. Необходимо выбрать язык программирования, в моем случае CSharp. В строке расположение будут указаны пути к файлам snippet, поэтому всегда можно открыть нужный файл и взять его за основу для разрабатываемого фрагмента кода (snippet-a).

Visual Studio C#: создание фрагментов кода IntelliSense (snippets)



Подробнее почитать можно здесь:

<https://msdn.microsoft.com/ru-ru/library/ms171418.aspx>

<https://msdn.microsoft.com/ru-ru/library/ms165394.aspx>

<https://msdn.microsoft.com/en-US/library/ms379562?f=255&MSPPErr=-2147217396>