**Что такое сниппеты и почему это крутая фича**

**Сниппетом** (или, по-английски, code snippet) называют короткий фрагмент текста, который можно вставить в основной код программы, и, затем, настроить в соответствии с потребностью разработчика. С помощью сниппетов не получится сгенерировать законченный проект или что-то в этом роде, зато их можно использовать для быстрой вставки часто используемых конструкций или трудно запоминающихся блоков программ. Основным достоинством сниппетов является значительное увеличение **скорости** и эффективности написания кода.

Как вообще устроены сниппеты? Сниппет – некоторый код, записанный в отдельный XML-файл, вместе с переменными параметрами, упрощающими его настройку в рамках некоторой конкретной задачи. Сниппеты также являются частью технологии **IntelliSense**.

**Сниппеты можно вставлять в следующие части программы:**

* **Объявление класса/структуры**. В данном случае сниппет, по сути, генерирует целый класс/структуру.

Например:

**struct MyStruct**

**{**

**struct+TAB+TAB**

**}**

* **Объявление члена класса/структуры.** В этом случае сниппет содержит определение членов, в частности, методов, свойств и обработчиков событий, поэтому его следует вставлять за пределами существующих членов класса/структуры.

К примеру, вот так можно создать автоматически реализуемое свойство:

**prop+TAB+TAB public int MyProperty { get; set; }**

* **Тело члена класса.** Эта часть кода предназначена для сниппетов, которые вставляются в уже определенный член класса.

Например, если необходимо добавить цикл **for** в метод, это будет выглядеть так:

**for (int i = 0; i<length; i++)**

**{**

**for+TAB+TAB**

**}**

Из приведённых выше примеров, очевидно, что для того, чтобы код волшебным образом дописался за вас, необходимо написать **название сниппета**, а затем **два раза нажать на TAB**.

**Часто используемые сниппеты:**

**I. Классы, структуры и все такое прочее**

* **class** – создает новый класс
* **struct** – создает новую структуру
* **ctor** – создает конструктор без параметров для данного класса/структуры
* **exception** – создает класс пользовательского исключения, унаследованного от **System.Exception**

**II. Циклы**

* **while** – создает конструкцию цикла вида **while(true) {}**
* **do** – создает конструкцию цикла вида **do {} while(true);**
* **for** – создает конструкцию цикла вида **for (int i = 0; i < length; i++) {}**
* **forr** – создает конструкцию цикла такого же вида как и **for**, только проход осуществляется в обратном порядке
* **foreach** – создает конструкцию цикла вида **foreach (var item in collection) {}**

**III. Свойства и методы**

* **prop** – создает автореализуемое свойство
* **propg** – создает свойство с открытым геттером и закрытым сеттером
* **propfull** – создает свойство со стандартным описанием:

**private int myVar;**

**public int MyProperty**

**{**

**get { return myVar; }**

**set { myVar = value; }**

**}**

* **svm** – создает метод **Main** с возвращаемым значением типа **void**
* **indexer** – создает индексатор

**IV. Блоки try**

* **try** – создает блок **try {} catch (Exception) {}**
* **tryf** – создает блок **try {} finally {}**

**V. Условия**

* **if** – создает блок **if(true) {}**
* **else** – создает пустой блок **else**
* **switch** – создает блок **switch** с **default**

**VI. Консольный вывод**

* **cw** – создает **Console.WriteLine()**

Помимо встроенных сниппетов можно также создавать и свои собственные.

Добавление сниппетов в Visual Studio:

**Средства Диспетчер фрагментов кода Импорт**

**Tools Code Snippets Manager Import**