

Всплывающие подсказки - это то, что появляется, когда вы подводите мышку к написанному коду C# (или когда пишете свой, но об этом чуть ниже).

```
project.SendInfoToLog("test", true);
```

void IZennoPosterProjectModel.SendInfoToLog(string message, bool showInPoster) (+1 overload)  
Send an information to the log of ZennoPoster.

Если подсказок нет, но код при этом рабочий, то скорее всего у вас нет файла с этими подсказками, а возможно просто разработчик не позаботился об этом. Подсказки нужны для облегчения чтения/написания кода с использованием различных dll библиотек. Эти dll библиотеки используются как часть какой-либо программы (например ZennoPoster) или дополнение по расширению функционала и повышают удобство чтения/написания кода.

В большинстве случаев они на английском, но могут быть и на других языках, в том числе русском.

```
project.SendInfoToLog("test", true);
```

void IZennoPosterProjectModel.SendInfoToLog(string message, bool showInPoster) (+1 overload)  
Отправка информационного сообщения (information) в лог ZennoPoster.

Согласитесь, удобно при написании/чтении кода знать описание и какой тип и сколько параметров принимает выбранный метод, какие из них обязательные и что в них нужно указывать и в каком формате.

При использовании методов, подсказки появляются, когда вы ставите открывающую скобку, то сразу появляется подсказка для первого параметра (он выделяется жирным шрифтом)

```
project.SendInfoToLog(
```

2 of 2 void IZennoPosterProjectModel.SendInfoToLog(string message, string way, bool showInPoster)  
Send an information to the log of ZennoPoster.  
message: Type: System.String An information for send.

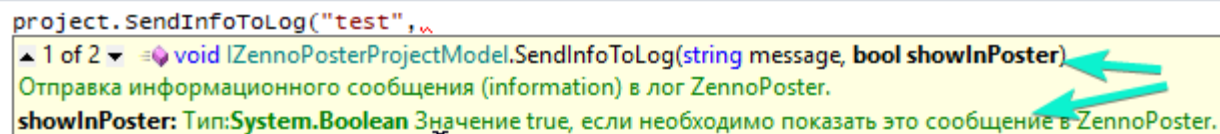
Конечно, лучше, когда эти подсказки на понятном языке (русском)

```
project.SendInfoToLog(
```

2 of 2 void IZennoPosterProjectModel.SendInfoToLog(string message, string way, bool showInPoster)  
Отправка информационного сообщения (information) в лог ZennoPoster.  
message: Тип: System.String Информации для отправки.

Если параметров несколько, то после того, как поставите запятую после предыдущего параметра, следующий параметр выделяется жирным шрифтом и в подсказке появляется его описание.

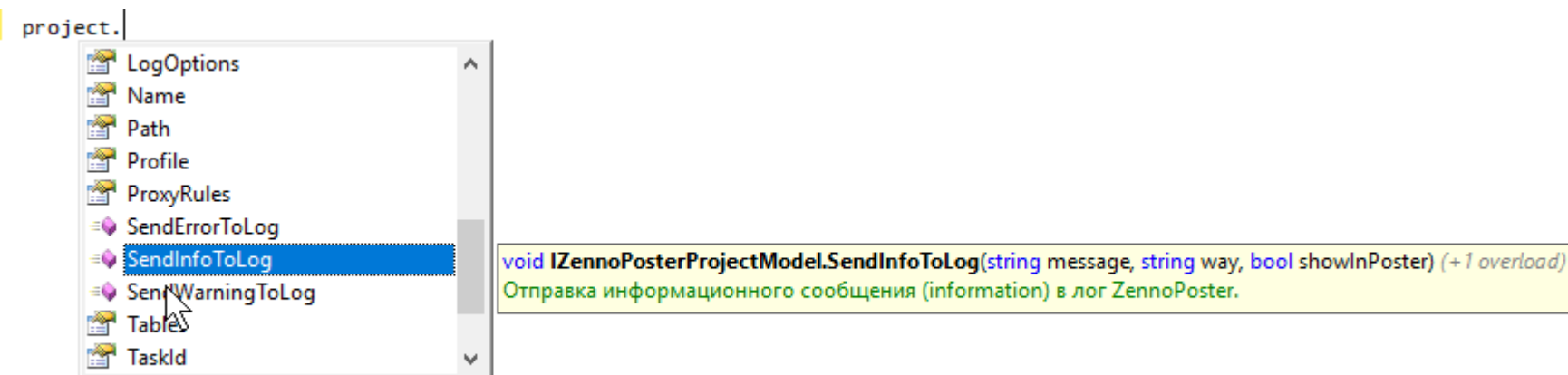
```
project.SendInfoToLog("test",
```



▲ 1 of 2 ▾ void IZennoPosterProjectModel.SendInfoToLog(string message, bool showInPoster)  
Отправка информационного сообщения (information) в лог ZennoPoster.  
showInPoster: Тип: System.Boolean. Значение true, если необходимо показать это сообщение в ZennoPoster.

Так же удобно посмотреть какие вообще доступны действия с текущим объектом. Список этих действий появляется после того, как поставили точку после названия текущего объекта. После выбора нужного пункта в списке появляется всплывающая подсказка к выбранному действию.

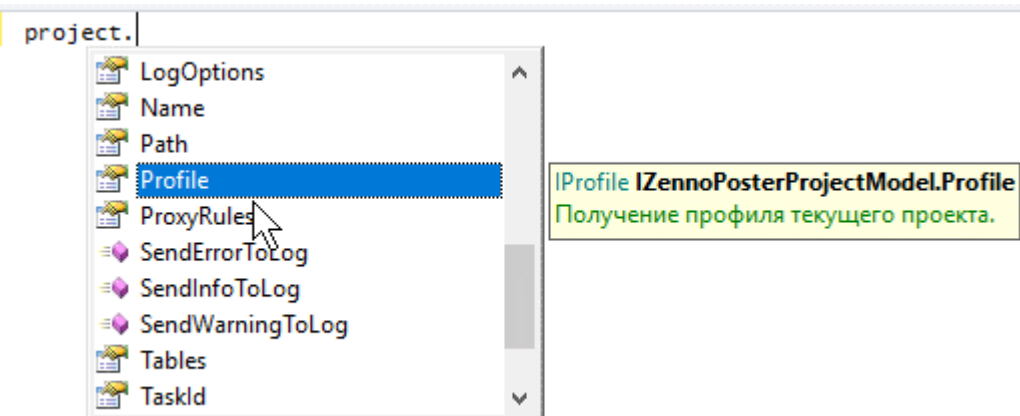
```
project.
```



LogOptions  
Name  
Path  
Profile  
ProxyRules  
SendErrorToLog  
SendInfoToLog  
SendWarningToLog  
Tables  
TaskId

void IZennoPosterProjectModel.SendInfoToLog(string message, string way, bool showInPoster) (+ 1 overload)  
Отправка информационного сообщения (information) в лог ZennoPoster.

```
project.
```




LogOptions  
Name  
Path  
Profile  
ProxyRules  
SendErrorToLog  
SendInfoToLog  
SendWarningToLog  
Tables  
TaskId

IProfile IZennoPosterProjectModel.Profile  
Получение профиля текущего проекта.

Из подсказки сразу видно описание действия, тип параметра, их количество, есть ли возвращаемое значение и выбрать нужное.


При наведении мышки на любой из объектов, можно лучше понять какие взаимодействия с ним происходят.

```
var test = project.Variables["Test1"].Value;
```

(parameter)  **ZennoLab.InterfacesLibrary.ProjectModel.IZennoPosterProjectModel** project


Представляет собой модель текущего проекта. Выполняет доступ в глобальные и локальные переменные, таблиц, профилей, листов. Содержит методы для отправки сообщения о предупреждении программы и журнала.

```
var test = project.Variables["Test1"].Value;
```

 **ILocalVariables IZennoPosterProjectModel.Variables**

Получение набора (collection) локальных переменных, доступ к которым может быть получен по имени.

```
var test = project.Variables["Test1"].Value;
```

 **string ILocalVariable.Value**

Получение или установка значения локальной переменной.


Так же подсказки могут правильно писать код, ведь иногда возникают ошибки из-за большой/маленькой буквы в названии объекта/действия и при ошибке подсказки не появляются, правильное написание можно посмотреть в выпадающем списке доступных действий.

Всплывающие подсказки доступны не только в ZennoPoster, но и для большинства других (в том числе бесплатных) dll библиотек.

С подсказками на русском языке всегда легче писать код или изучать новое. В своем коде, конечно, можно комментарии писать.

```
// Подключаемся к базе данных указав путь к ней
```

```
using (SQLiteConnection connect = new SQLiteConnection(Strin
```

```
//  class System.Data.SQLite.SQLiteConnection
```

```
con SQLite implementation of DbConnection.
```

Но, если в чужом коде таких комментариев нет, а есть всплывающие подсказки для используемых в коде библиотек, то они очень помогают при анализе чужого кода или написании своего.