

## Урок 4. Отладка кода

На этом уроке мы научимся использовать отладчик Visual Studio для пошагового исполнения плагина и отслеживания изменения переменных.

**Обратная связь:** напишите нам об этом уроке или обо всем курсе «Моя первая программа»: [myfirstplugin@autodesk.com](mailto:myfirstplugin@autodesk.com)  
(Пожалуйста, пишите на английском языке)

Редко удается сразу написать идеально работающую программу. Даже если в вашей программе нет ошибок, препятствующих компиляции плагина в Visual Studio, вы могли допустить ошибку в программной логике, из-за чего программа может выдавать результаты, отличающиеся от ожидаемых. Процесс пошагового исполнения кода для изучения хода работы программы и отслеживания значений переменных называется отладкой.

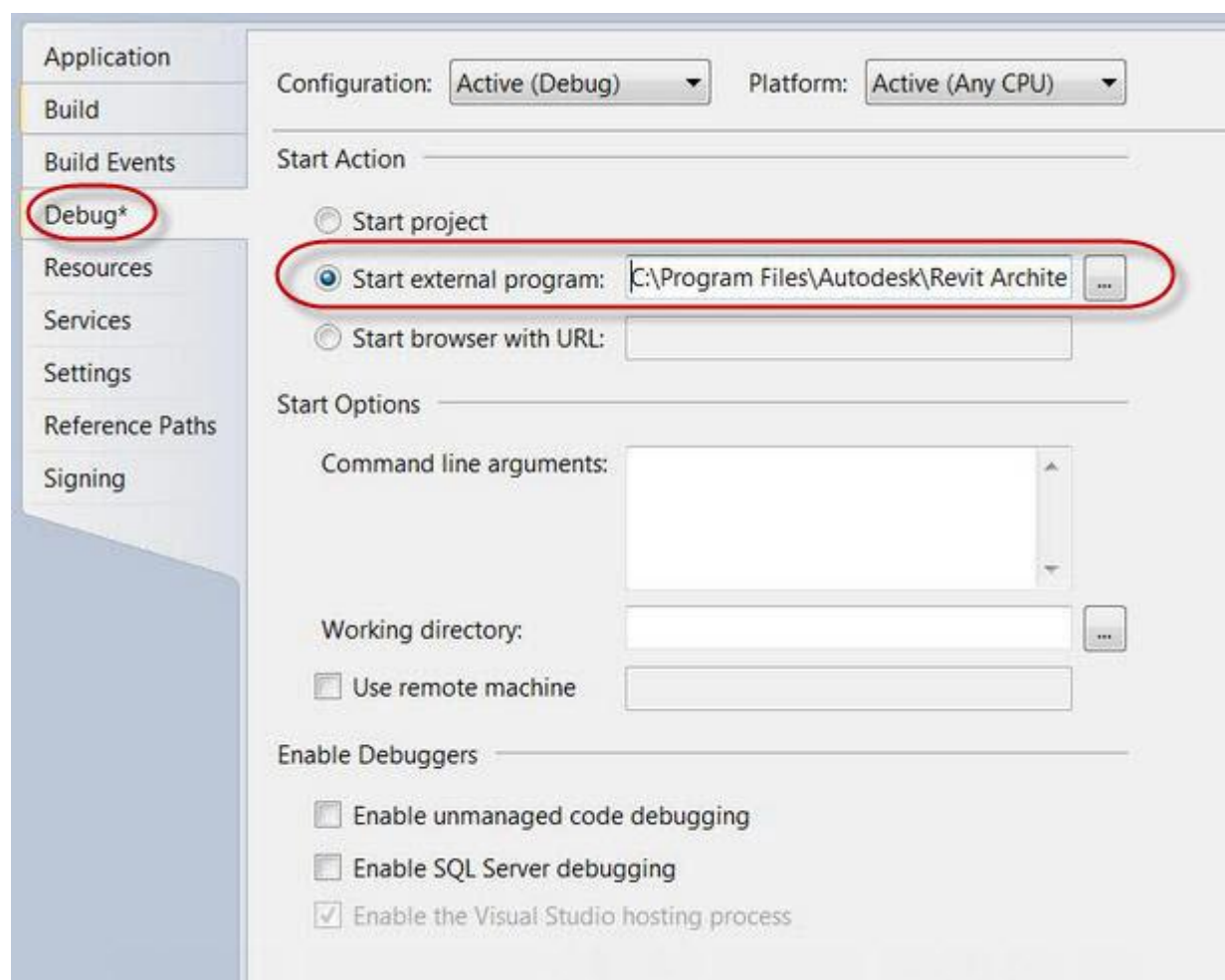
Для отладки кода плагина вы должны запустить Revit из отладчика Visual Studio Express и загрузить в Revit свой плагин. Обратите внимание, что Express-версии Visual C# и Visual VB.NET не позволяют настроить проект на отладку DLL-файлов, как нам требуется. Поэтому для отладки с помощью Visual C# Express необходимо внести некоторые изменения в файлы проекта вручную, выполнив следующие действия:

1. Найдите файл проекта C# (\*.csproj) в папке проекта. В данном случае это – файл Lab1PlaceGroup.csproj.
2. **Откройте файл:**  
Запустите приложение **Блокнот** из меню **Пуск**. В Блокноте из меню **Файл** выберите **Открыть**. Перейдите к файлу и откройте его.
3. Найдите первый тэг <PropertyGroup> (этот тэг встречается в файле несколько раз).
4. Сразу перед закрывающим тэгом </PropertyGroup> вставьте две строки с тэгами <StartAction> и <StartProgram> (отмечены полужирным шрифтом). Теперь файл Lab1PlaceGroup.csproj должен выглядеть следующим образом:

```
<PropertyGroup>
    . . .
    <StartAction>Program</StartAction>
    <StartProgram>C:\Program Files\Autodesk\Revit Architecture 201x\Program\Revit.exe
    </StartProgram>
</PropertyGroup>
```

Если Revit Architecture установлен в папку, отличающуюся от предлагаемой по умолчанию, отредактируйте путь к Revit.exe в тэге <StartProgram>.

В профессиональных версиях Visual Studio это делается проще: в окне **Properties** можно задать исполняемый файл, который будет запускаться вместе с отладчиком:



Эта возможность недоступна в Express IDE, поэтому мы и продемонстрировали другой способ добавления параметров в файл .csproj.

### Запуск отладчика

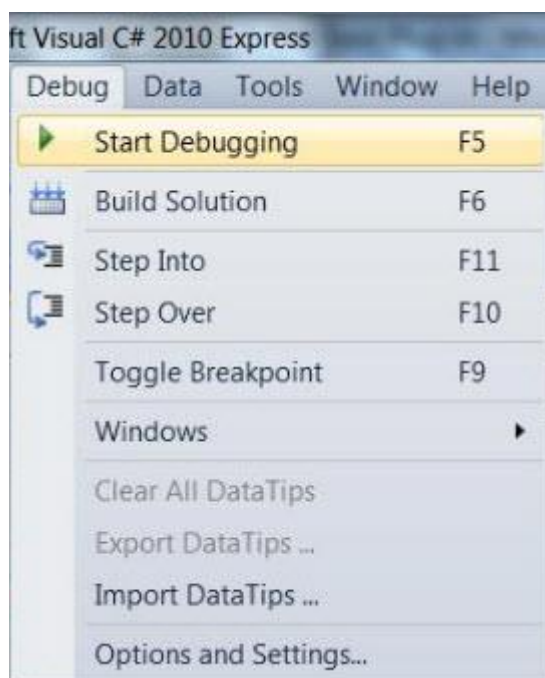
Если вы закрыли среду Visual C# 2010 Express, запустите ее снова и откройте проект, который был сохранен на уроке 1 (Lab1PlaceGroup).

Перед запуском отладчика убедитесь, что файл манифеста для плагина создан и находится в нужной папке, как описано в разделе *Написание файла манифеста* в [Уроке 1: Общее представление о программе](#). Поскольку теперь мы начнем отладку кода плагина, Visual C# Express создает отладочную версию DLL-файла плагина .NET. Для того чтобы Revit прочитал отладочную версию DLL-файла вместо

финальной версии, созданной на первом уроке, необходимо изменить путь к DLL-файлу в манифесте следующим образом (изменения отмечены полужирным шрифтом):

```
<?xml version="1.0" encoding="utf-8"?>
<RevitAddIns>
  <AddIn Type="Command">
    <Assembly>
      C:\test\Lab1PlaceGroup\Lab1PlaceGroup\bin\ReleaseDebug\Lab1.dll
    </Assembly>
    <ClientId>502fe383-2648-4e98-adf8-5e6047f9dc34</ClientId>
    <FullClassName>Lab1PlaceGroup</FullClassName>
    <Text>Lab1PlaceGroup</Text>
    <VendorId>ADSK</VendorId>
    <VisibilityMode>AlwaysVisible</VisibilityMode>
  </AddIn>
</RevitAddIns>
```

Чтобы запустить сеанс отладки, из меню **Debug** выберите **Start Debugging** или просто нажмите F5. Из отладчика запускается Revit. Разновидность Revit, которая будет запускаться, зависит от того, какой путь указан в отредактированном файле .csproj.



Revit следует запускать именно таким способом, чтобы отладчик мог подключиться к процессу Revit для отслеживания исполнения кода DLL-плагина. Поскольку мы поместили манифест в нужную папку, Revit автоматически загружает наш плагин.

*Если при этом открыто окно Visual C# Express Immediate, в нем будет появляться много текста. Некоторые сообщения могут напугать вас, так как в них содержится слово «error» (ошибка). Не беспокойтесь. Этот текст выводится только в информационных целях, он не указывает на проблемы с Revit или с плагином.*

*Вспомним, что DLL-файл, который загружался на уроке 1, находился в папке bin\Release. При сборке окончательной версии DLL-файла плагина, которую можно будет предоставлять пользователям и клиентам, Visual Studio создает **финальную** версию. При сборке финальной версии Visual Studio выполняет некоторую оптимизацию скомпилированного кода, что ускоряет исполнение программы и сокращает затраты оперативной памяти. Если вы хотите отладить свой код, Visual Studio создает **отладочную** версию DLL-файла плагина. Отладочная версия не оптимизирована с точки зрения скорости работы и потребления памяти; кроме того, в нее включается дополнительная информация, которую использует отладчик, чтобы информировать вас о ходе работы программы.*

**Примечание.** Если вы пользуетесь Visual Studio 2010 Professional (а не Express Edition), у вас есть еще один способ настроить запуск Revit из отладчика Visual Studio. Подробная информация представлена в этой статье: [How to: Attach to a Running Process](#).

Теперь откройте файл проекта [Hotel.rvt](#).

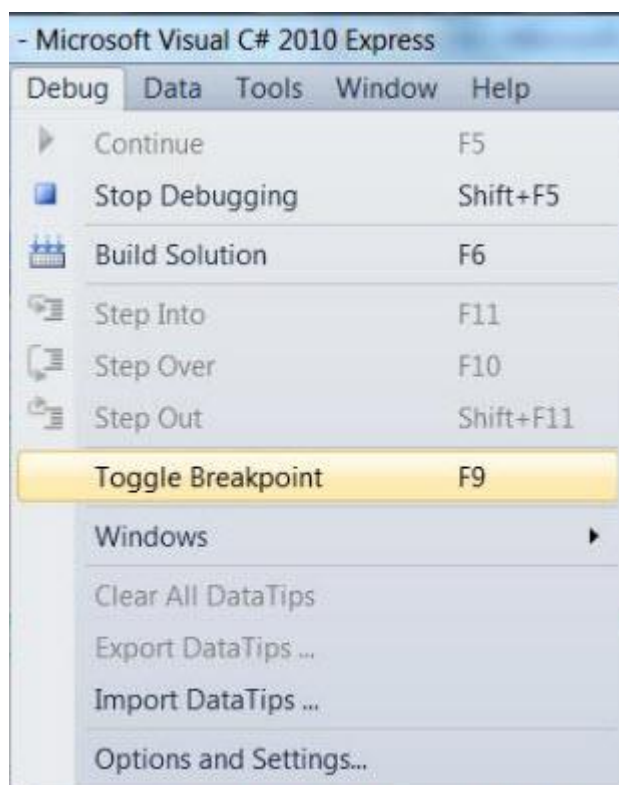
## Контрольные точки

Ваш DLL-плагин готов к отладке. Однако перед тем, как запускать команду **Lab1PlaceGroup**, необходимо указать, в каких местах кода его исполнение будет останавливаться. Для этого используются **контрольные точки**.

В Visual C# Express дважды щелкните на Lab1.cs в Solution Explorer, чтобы отобразить код, а затем установите курсор в любую позицию следующей строки:

```
UIApplication uiApp = commandData.Application;
```

Затем из меню **Debug** выберите **Toggle Breakpoint** (или нажмите **F9**).



Строка, указанная в качестве контрольной точки, выделяется красным цветом и отмечается красным кружком в поле слева:

```
[TransactionAttribute(TransactionMode.Manual)]
[RegenerationAttribute(RegenerationOption.Manual)]
public class Lab1PlaceGroup : IExternalCommand
{
    public Result Execute(
        ExternalCommandData commandData,
        ref string message,
        ElementSet elements)
    {
        //Get application and document objects
        UIApplication uiApp = commandData.Application;
        Document doc = uiApp.ActiveUIDocument.Document;
    }
}
```

Таким же способом зададим контрольную точку для функции **PickPoint**:

```
XYZ point = sel.PickPoint("Укажите точку для размещения группы");
```

Когда Revit вызывает эти методы в вашем коде, отладчик останавливает исполнение программы в соответствующих строках и ожидает указаний пользователя.

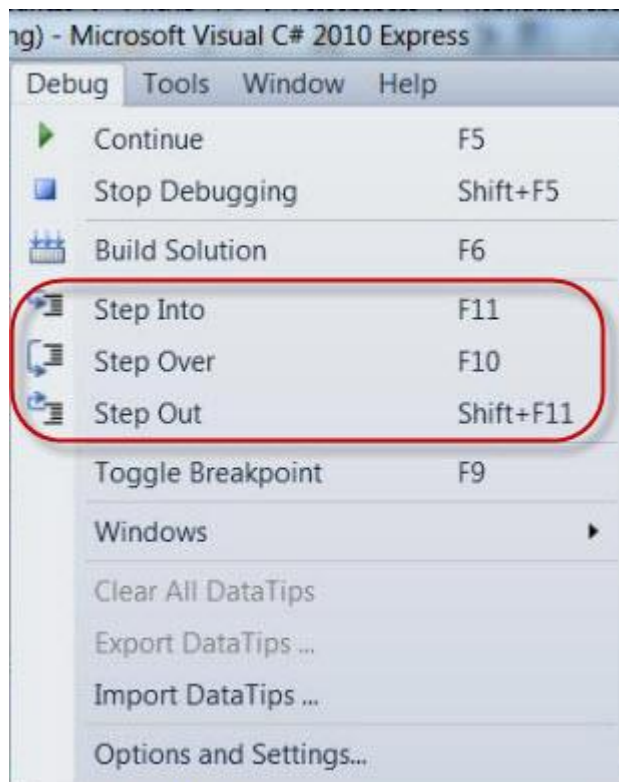
## Пошаговое исполнение кода

Пора запустить нашу команду. В Revit на вкладке ленты **Add-Ins** из выпадающего списка **External Tools** выберите **Lab1PlaceGroup**. Запускается плагин Revit, а Visual C# Express перехватывает управление и становится активным приложением (если этого не происходит, щелкните на значке приложения в панели задач Windows.) Отладчик останавливает выполнение программы, ожидая действий пользователя. Строка кода, которая будет исполнена следующей, выделяется желтым цветом и отмечается желтой стрелкой в поле слева.

```
[TransactionAttribute(TransactionMode.Manual)]
[RegenerationAttribute(RegenerationOption.Manual)]
public class Lab1PlaceGroup : IExternalCommand
{
    public Result Execute(
        ExternalCommandData commandData,
        ref string message,
        ElementSet elements)
    {
        //Get application and document objects
        UIApplication uiApp = commandData.Application;
        Document doc = uiApp.ActiveUIDocument.Document;

        //Define a Reference object to accept the pick result.
        Reference pickedRef = null;
    }
}
```

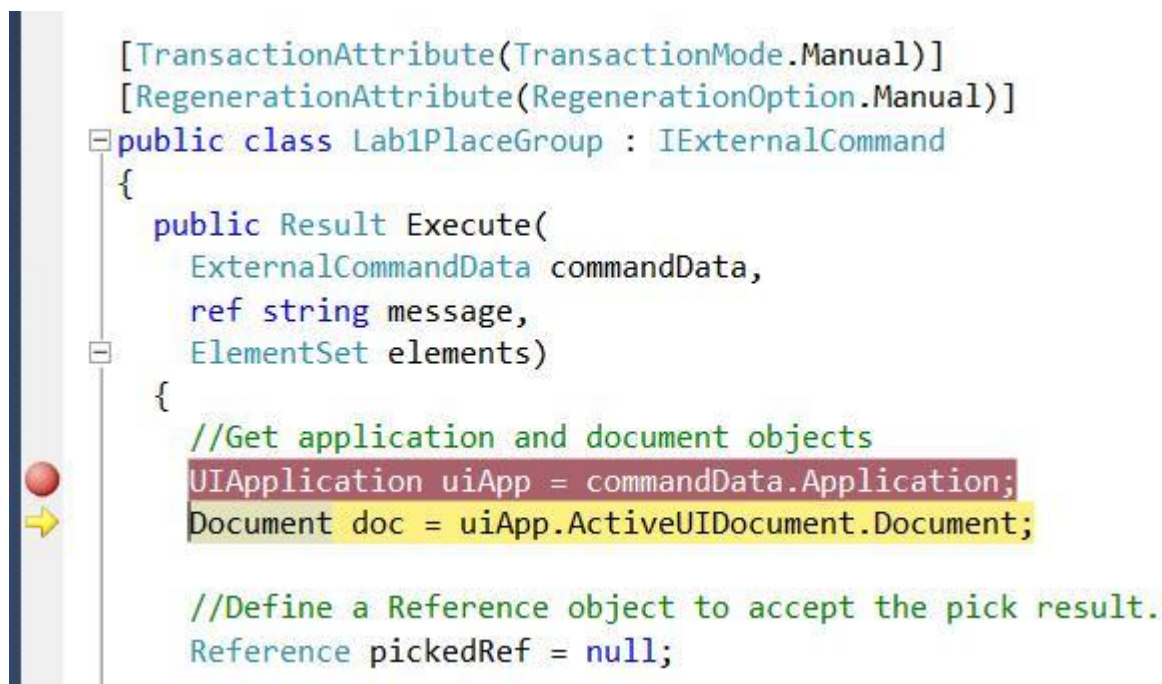
Теперь можно исполнять код строку за строкой. В меню отладки Visual C# Express имеются три функции пошагового исполнения: **Step Into**, **Step Over** и **Step Out**. Чаще всего вы будете пользоваться функцией *Step Over* — она исполняет текущую строку кода (выделенную желтым) в отладчике и переходит к следующей строке. Если в исполняемой строке содержится вызов метода, то функция *Step Over* исполняет весь метод. Если вы хотите исполнять вызываемый метод также построчно, то можете воспользоваться функцией *Step Into*. Функция *Step Out* позволяет вернуться из метода к основному коду.



Функции *Step Into*, *Step Over* и *Step Out* можно вызвать не только из меню *Debug*, но и с помощью соответствующих кнопок на панели инструментов, а также с помощью сочетаний клавиш (**F8**, **Shift+F8** и **Ctrl+Shift+F8** соответственно).



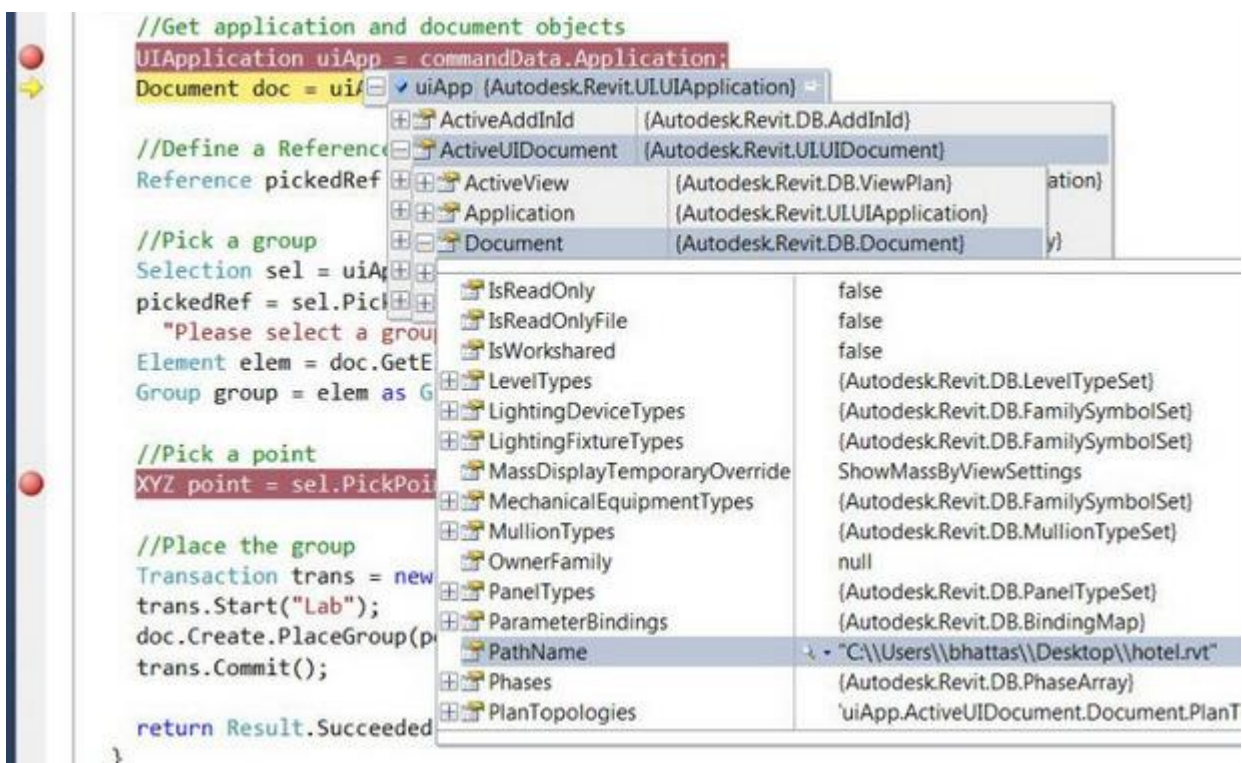
Нажмите кнопку *Step Over* на панели инструментов. Отладчик переходит к следующей строке кода (строки с комментариями игнорируются).



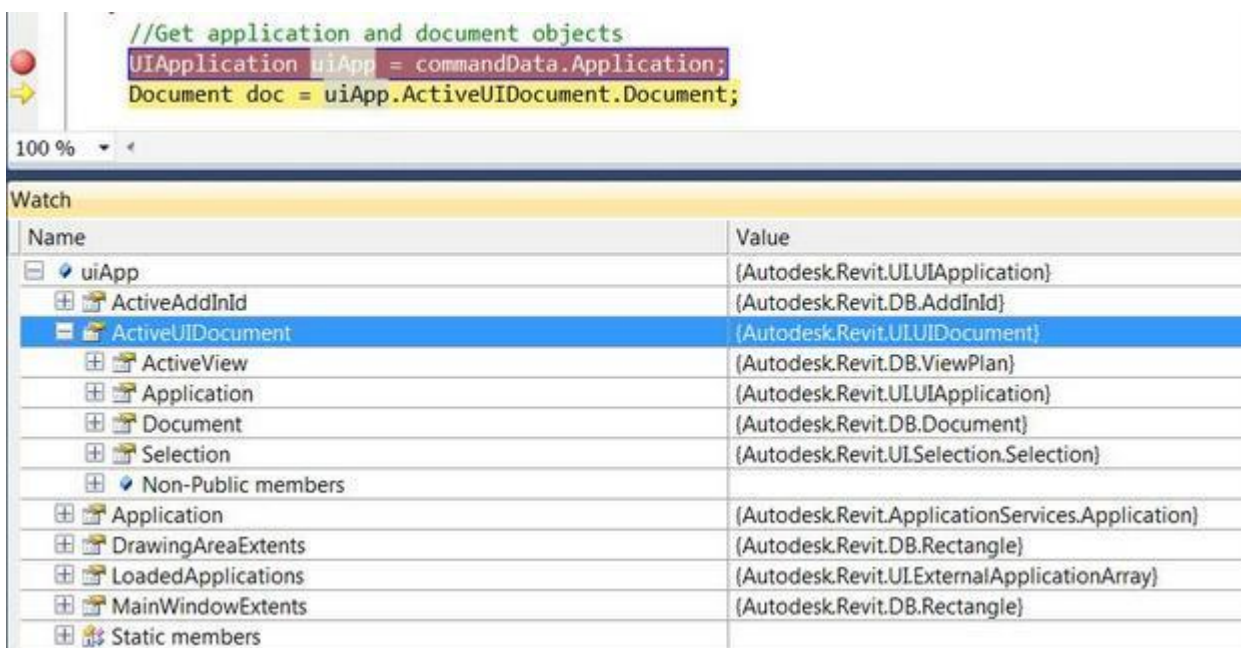
Если навести курсор на текст **uiApp**, во всплывающей подсказке отобразится текущее значение этой переменной.



При щелчке на символе «+» разворачивается список свойств, где можно просмотреть их значения.



Для отображения значения переменной можно также щелкнуть правой кнопкой мыши на имени переменной в коде и выбрать **Add Watch**. Значение переменной отображается в окне Watch в нижней части среды разработки.



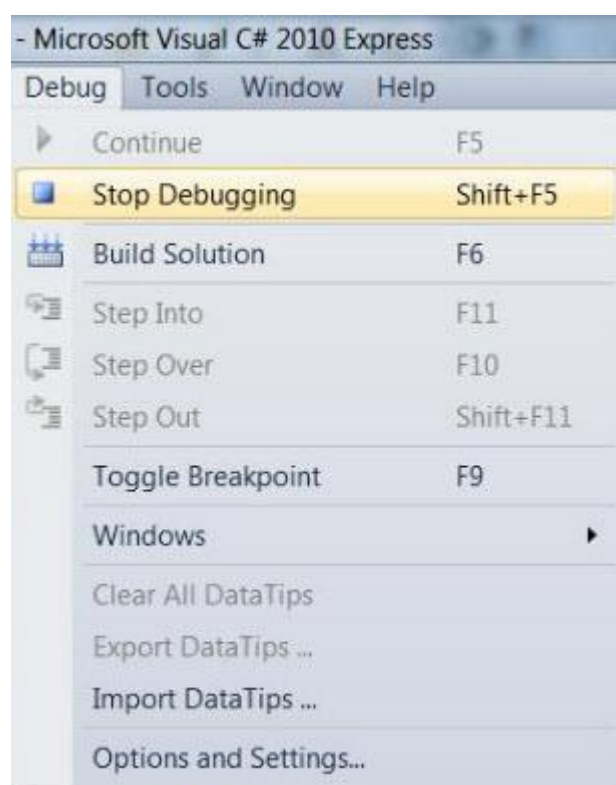
Продолжайте пошаговое исполнение кода, наблюдая за изменением значений переменных и свойств. Нажатие клавиши *F5* останавливает построчную отладку, и исполнение кода продолжается до следующей контрольной точки; в нашем случае — до строки **sel.PickPoint()**.

Пройдите эту строку, нажав *F10*. Revit перемещается на передний план и предлагает указать, куда необходимо скопировать группу Furniture. Щелкните в центре смежной комнаты. После выбора точки активным приложением опять становится Visual C# Express; следующая строка выделяется желтым цветом, а отладчик дает возможность проверить координаты указанной точки, продолжить построчную отладку или перейти к следующей контрольной точке путем нажатия *F5*. Проверка координат осуществляется путем наведения курсора на переменную-контейнер, которую мы создали для хранения точки назначения при копировании группы мебели.



Если теперь нажать *F5*, отладчик исполнит остальной код и снова выведет на передний план Revit, в котором отображается результат успешно выполненного копирования.

Закончив экспериментировать, из меню **Debug** выберите **Stop Debugging**, чтобы закрыть Revit и завершить сеанс отладки.



Теперь вы знаете, как отладить проект в Visual C# Express.