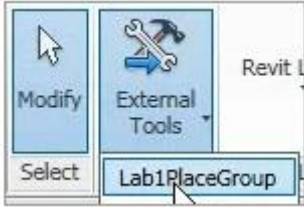


Урок 1: Общее представление о программе

На этом занятии мы создадим первый базовый плагин для Autodesk Revit, который копирует выбранную пользователем группу в указанное место.

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



Видео: Порядок создания вашего первого плагина.

Скачать материалы для урока 1

 [lesson1_revit_2013_projects.zip](#) (zip - 28884Kb)

 [lesson1_revit_2012_and_earlier_project_files.zip](#) (zip - 7283Kb)

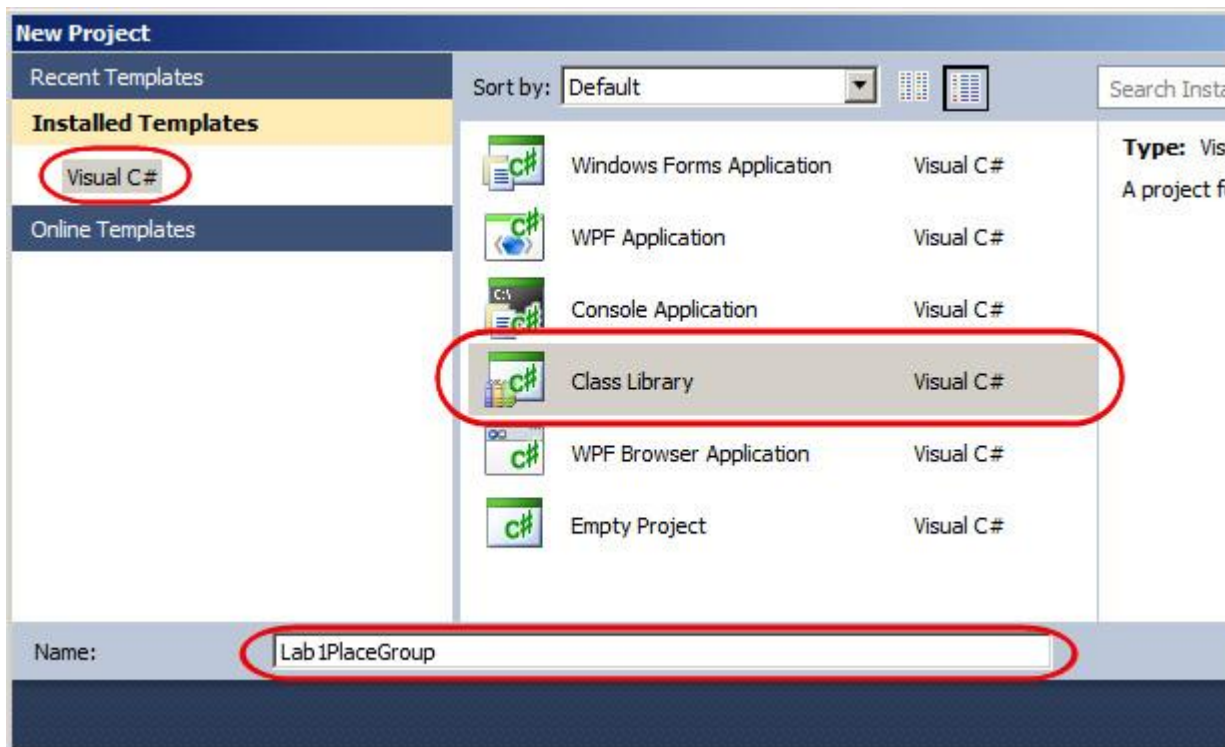
Этапы создания вашей первой программы

1. Запустите Visual C# Express.

Чтобы открыть Visual C# 2010 Express, из меню **Пуск** выберите **Все программы** » **Microsoft Visual Studio 2010 Express** » **Microsoft Visual C# 2010 Express**. Вы также можете использовать Visual Basic 2008 Express. К учебному пособию прилагаются файлы данных для обеих сред разработки приложений.

2. Создайте проект библиотеки классов.

В среде Visual C# Express из меню **Файл (File)** выберите **Создать проект (New Project)**. Слева на вкладке **Установленные шаблоны (Installed Templates)** выберите **Visual C#**. Затем в средней части укажите пункт **Библиотека классов (Class Library)**. В текстовом поле **Имя (Name)** введите **Lab1PlaceGroup**. Нажмите **ОК**.



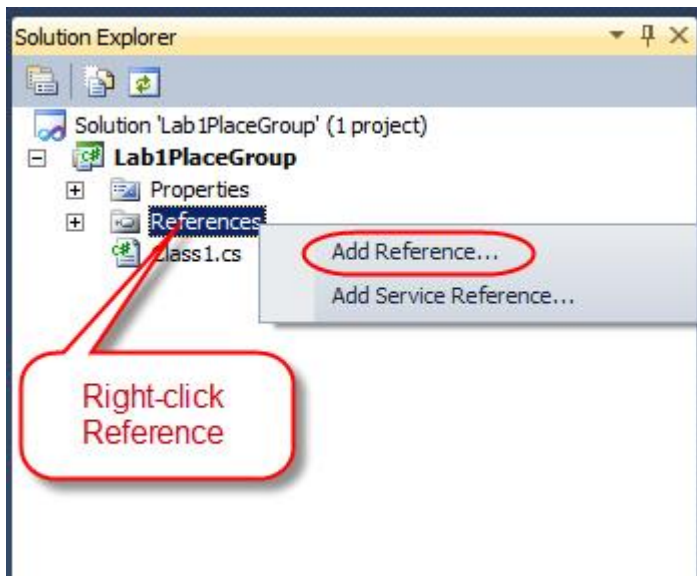
Visual C# Express создает проект с кодом по шаблону и открывает его.

3. Сохраните проект.

Из меню **Файл (File)** выберите **Сохранить все (Save All)**. В диалоговом окне, в поле **Расположение (Location)** наберите **C:\test**, а затем нажмите **Сохранить (Save)**.

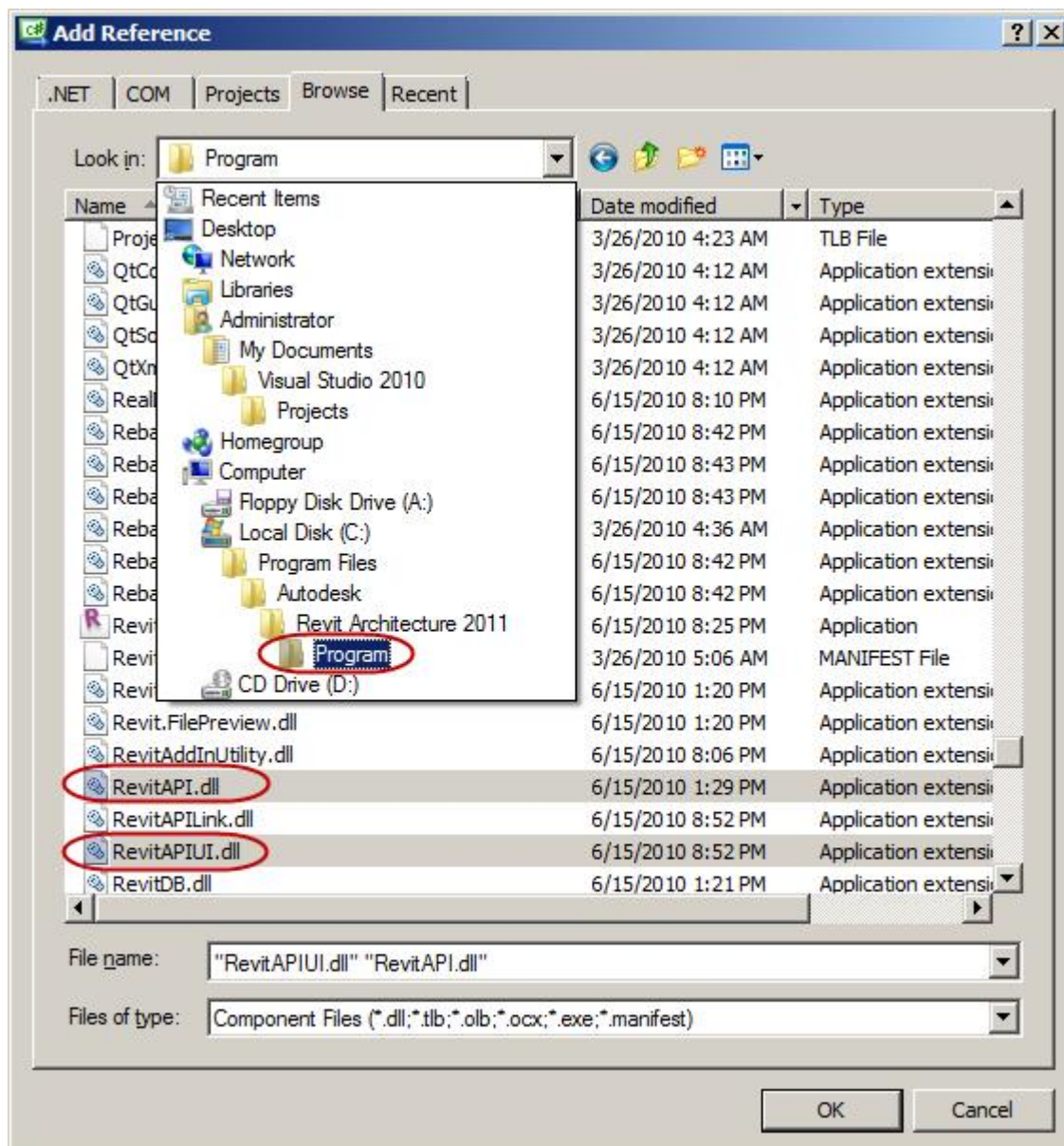
4. Добавьте ссылки.

В окне **Обозреватель решений (Solution Explorer)**, которое находится в правой части окна Visual C# Express, щелкните правой кнопкой мыши на элементе **Ссылки (References)** и из контекстного меню выберите **Добавить ссылку (Add Reference...)**.



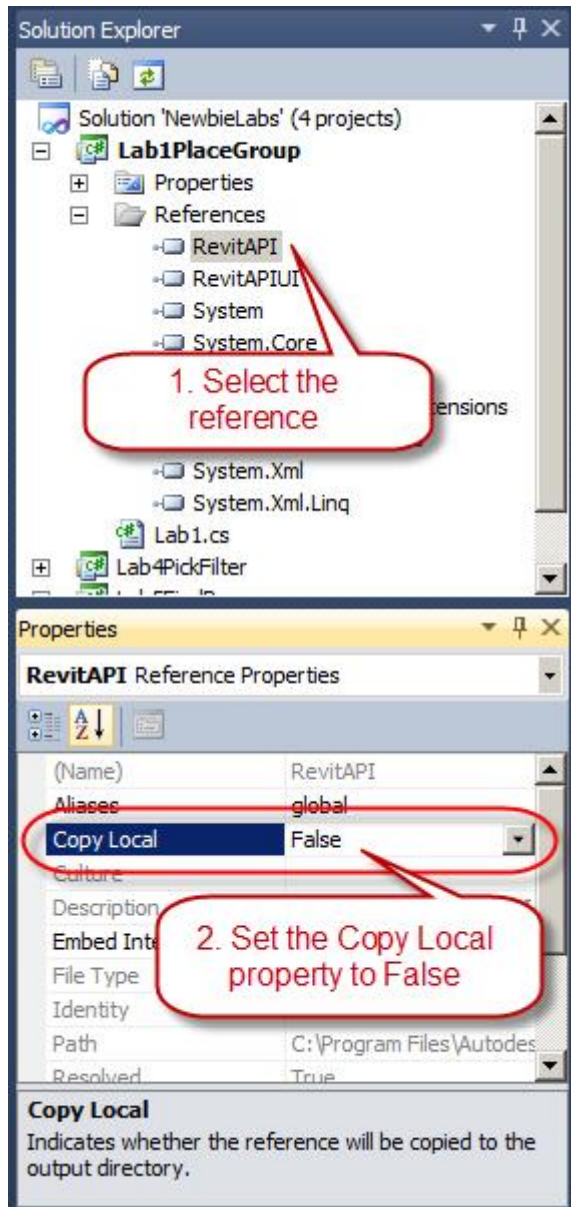
5. В диалоговом окне **Добавление ссылки (Add Reference)** перейдите на вкладку **Обзор (Browse)**. Найдите папку установки Revit. (Расположение этой папки зависит от того, куда был установлен Revit Architecture 201x. По умолчанию это – C:\Program Files\Autodesk\Revit Architecture 201x\Program*.)

* Путь зависит от используемой вами версии Autodesk Revit.



Добавим две ссылки из этой папки. Нажав и удерживая клавишу Ctrl, выберите **RevitAPI.dll** и **RevitAPIUI.dll**. Нажмите **OK**. Две ссылки на dll-библиотеки, в которые заложен весь функционал Revit API, добавляются в проект.

6. Измените значение параметра Copy Local.



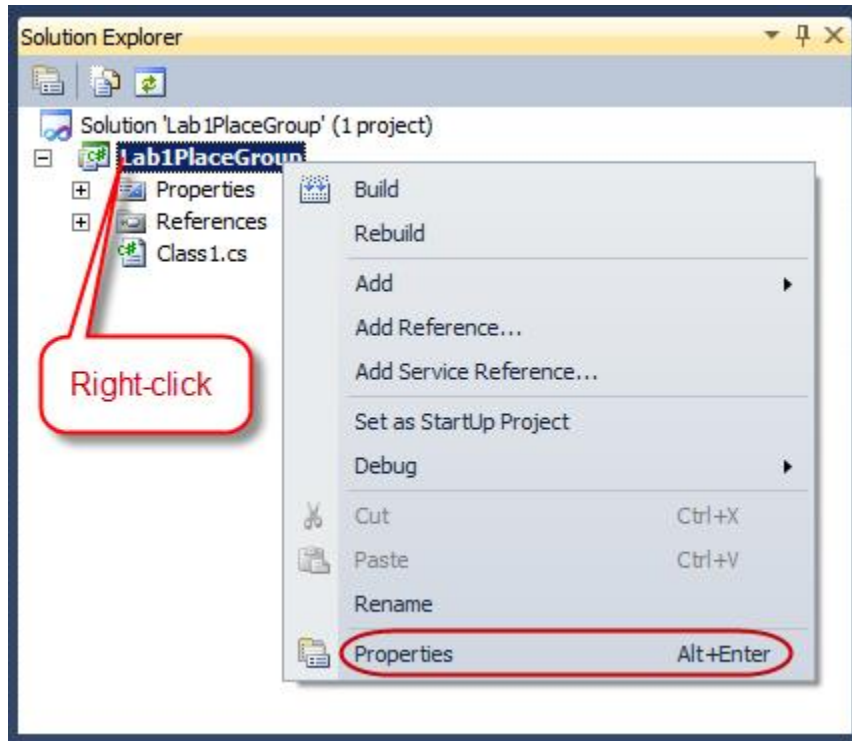
В окне **Обозревателя решений (Solution Explorer)**, с которым мы уже работали в пункте 5, щелкните на элементе **RevitAPI** в разделе **Ссылки (Reference)**. В окне **Свойства (Properties)** выберите параметр **Копировать Локально (Copy Local)**, разверните список значений и укажите **False**. Повторите те же действия с элементом **RevitAPIUI**.

7. Укажите версию .NET Framework.

Внимание! Autodesk Revit 2011 поддерживает .NET Framework 3.5, а Autodesk Revit 2012 и более поздние версии – .NET Framework 4.0, которую по умолчанию использует Visual C# 2010 Express. В этом пункте описывается порядок указания нужной версии .NET.

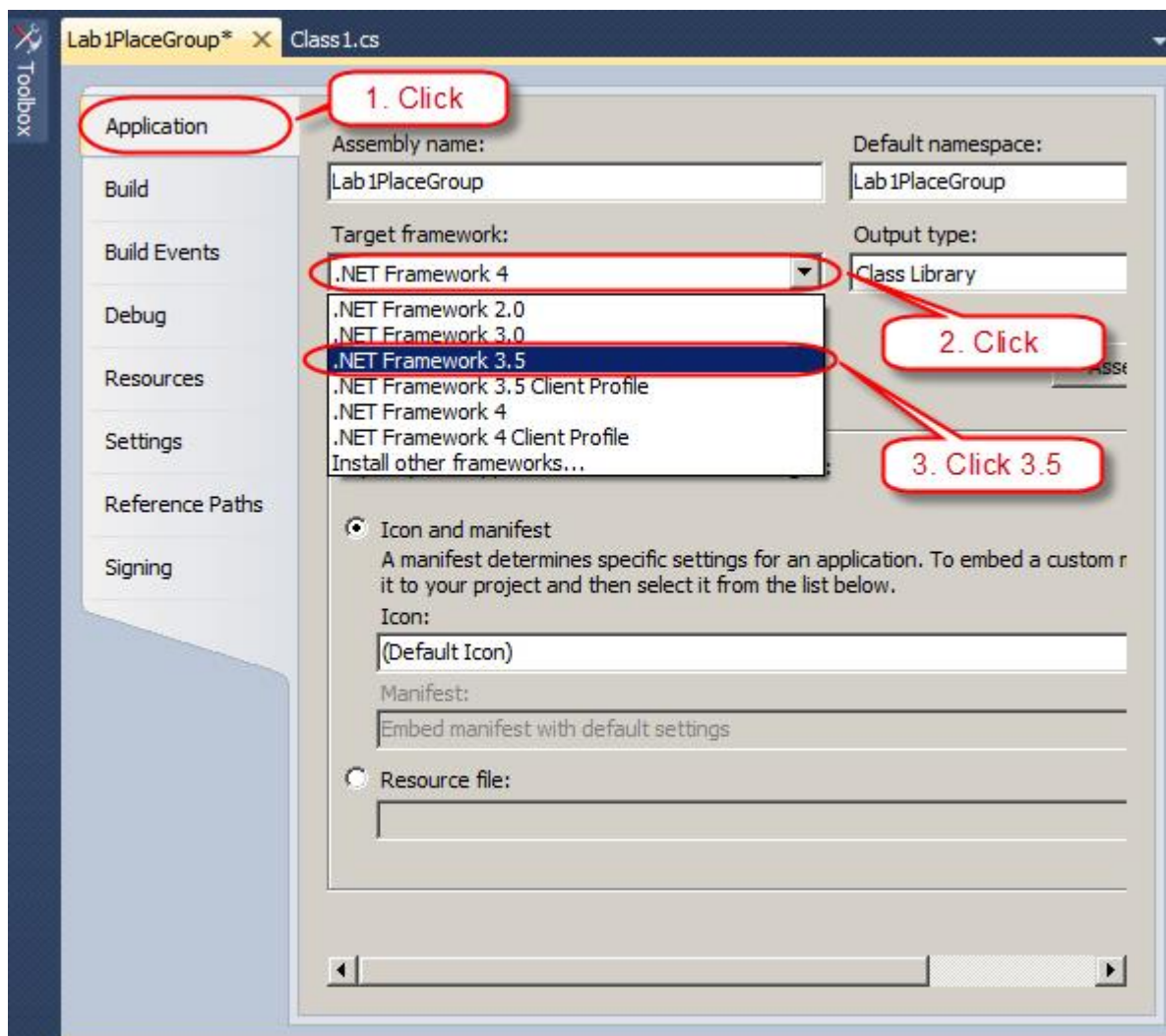
Если Revit Architecture, который вы используете, поддерживает .NET Framework 4.0, то пункты 7, 8 и 9 можно пропустить.

В окне **Обозревателя решений (Solution Explorer)** щелкните правой кнопкой мыши на проекте **Lab1PlaceGroup** и из контекстного меню выберите **Свойства (Properties)**.

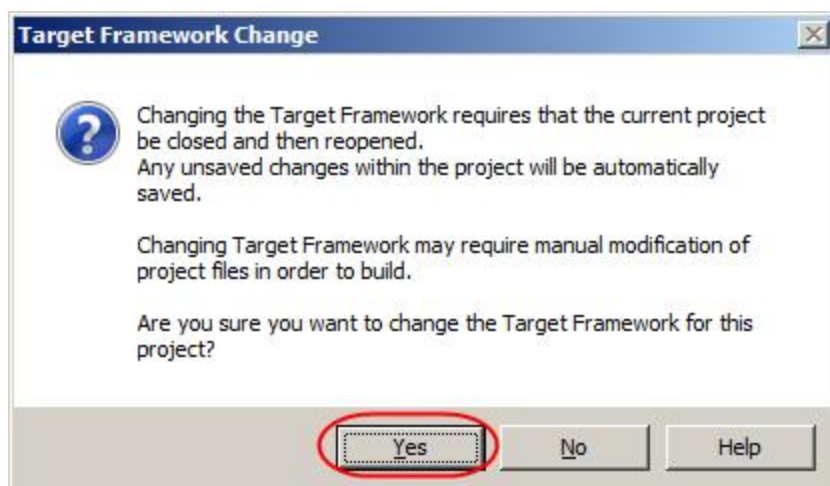


- В окне свойств (с левой стороны) перейдите на вкладку **Приложение (Application)** и из выпадающего списка **Требуемая версия .NET Framework (Target framework)** выберите **.NET Framework 3.5**.

Примечание: Если вы работаете с версией Revit 2013 API то вам потребуется выбрать .NET Framework 4.0



- Открывается окно подтверждения выполненных действий. Нажмите в нем Да (Yes).



- Добавьте программный код.**

Двойным щелчком на элементе **Class1.cs** в окне **Обозревателя решений (Solution Explorer)** откройте окно редактирования кода. Удалите весь шаблонный код и добавьте приведенный в этом пункте. Если у вас недостаточно времени, вы можете воспользоваться функциями копирования и вставки кода в Visual C# Express, но мы настоятельно рекомендуем набирать текст в редакторе кода вручную, чтобы оценить такие полезные возможности Visual C# Express, как IntelliSense.

```

using System;
using System.Collections.Generic;
using System.Linq;

using Autodesk.Revit.DB;
using Autodesk.Revit.DB.Architecture;
using Autodesk.Revit.UI;
using Autodesk.Revit.UI.Selection;
using Autodesk.Revit.ApplicationServices;
using Autodesk.Revit.Attributes;

[TransactionAttribute(TransactionMode.Manual)]
[RegenerationAttribute(RegenerationOption.Manual)]
public class Lab1PlaceGroup : IExternalCommand
{
    public Result Execute(
        ExternalCommandData commandData,
        ref string message,
        ElementSet elements)
    {
        //Получение объектов приложения и документа
        UIApplication uiApp = commandData.Application;
        Document doc = uiApp.ActiveUIDocument.Document;

        //Определение объекта-ссылки для занесения результата указания
        Reference pickedRef = null;

        //Указание группы
        Selection sel = uiApp.ActiveUIDocument.Selection;
        pickedRef = sel.PickObject(ObjectType.Element,
            "Выберите группу");
        Element elem = pickedRef.Element;
        Group group = elem as Group;

        //Указание точки
        XYZ point = sel.PickPoint("Укажите точку для размещения группы");

        //Размещение группы
        Transaction trans = new Transaction(doc);
        trans.Start("Lab");
        doc.Create.PlaceGroup(point, group.GroupType);
        trans.Commit();

        return Result.Succeeded;
    }
}

```

Не будем пока останавливаться на работе самого кода. Мы вернемся к нему позже, на занятии 3.

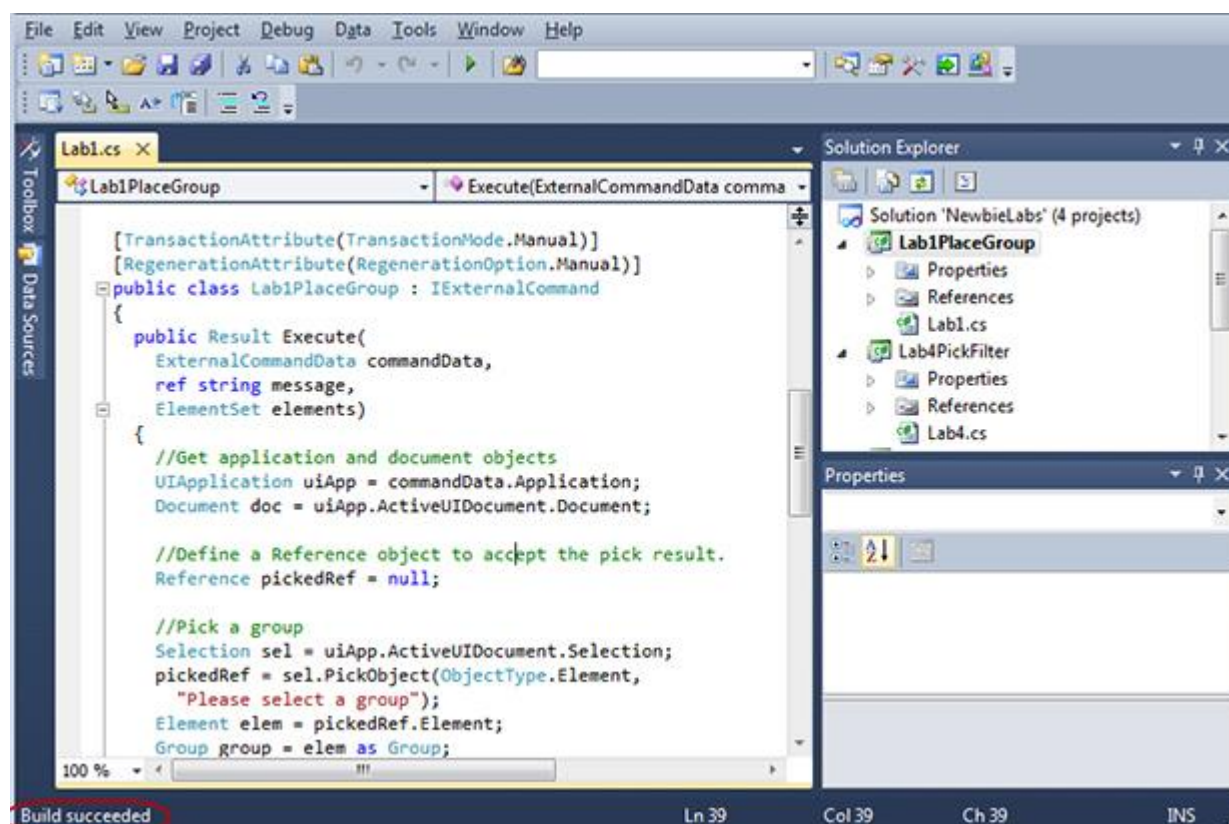
11. Сохраните файл.

Из меню **Файл (File)** выберите **Сохранить все (Save All)**.

12. Скомпилируйте проект.

Код, который мы ранее набрали, удобен для восприятия человеком; теперь же нам требуется сделать его понятным для компьютера. Этот процесс называется компиляцией.

Для компиляции вашего проекта из меню **Отладка (Debug)** Visual C# Express выберите **Построить решение (Build Solution)**. Если процесс завершается успешно, в строке состояния Visual C# Express появляется сообщение **Построение успешно завершено (Build Succeeded)**.



Примечание. Если вы работаете в Revit 2012 API, то увидите предупреждение, что Autodesk.Revit.DB.Reference.Element устарел. Не волнуйтесь пока по поводу этого предупреждения. Мы укажем, что нужно изменить в коде, на занятии 3.

При работе с API Revit 2013 и API более старших версий, вам будет необходимо заменить следующую строку в коде:

```
Element elem = pickedRef.Element;
```

На строку представленную ниже

```
Element elem = doc.GetElement(pickedRef);
```

Мы только что создали первый плагин для Autodesk Revit.

Для того чтобы с плагином можно было работать, нужен еще один шаг – создать файл манифеста.

Написание файла манифеста

Файл манифеста располагается в определенной папке, которую Revit просматривает при запуске. В этом файле содержится вся информация, необходимая для запуска плагина.

1. Добавьте код файла манифеста.

Запустите редактор **Блокнот** через меню **Пуск Windows**. Скопируйте в него приведенный ниже код:

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

В зависимости от версии Autodesk Revit, вам может понадобиться изменить строку, указывающую расположение файла Lab1PlaceGroup.dll:

```
C:\test\Lab1PlaceGroup\Lab1PlaceGroup\bin\Release\Lab1PlaceGroup.dll
```

2. Сохраните файл.

Из меню **Файл (File)** Блокнота выберите **Сохранить (Save)**. Введите **MyFirstPlugin.addin** в поле **Имя файла (File name)**. Измените **Тип файла (Save as type)** на значение **Все файлы (All Files)**. (Имя файла может быть любым, однако расширение всегда должно быть «.addin»)

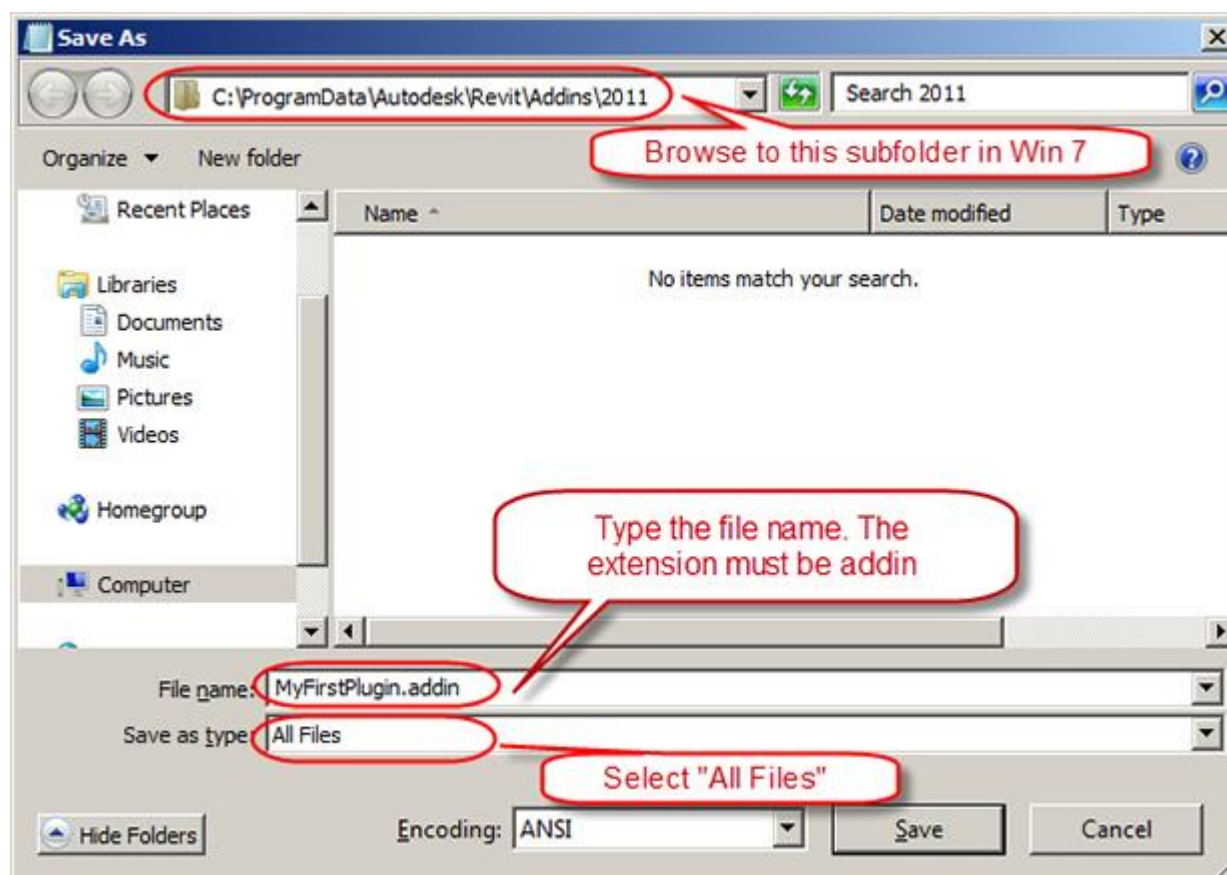
Откройте соответствующую папку (см. ниже) и нажмите **Сохранить (Save)**.

Для Windows XP* – C:\Documents and Settings\All Users\Application Data\Autodesk\Revit\Addins\2011\

Для Windows Vista/Windows 7* – C:\ProgramData\Autodesk\Revit\Addins\2011\ (папка ProgramData по умолчанию скрыта)

* Путь может различаться в зависимости от версии Autodesk Revit, которую вы используете.

Ниже приведен образец диалогового окна «Сохранить как» (Save As) в среде Windows 7 для Revit Architecture 2011.



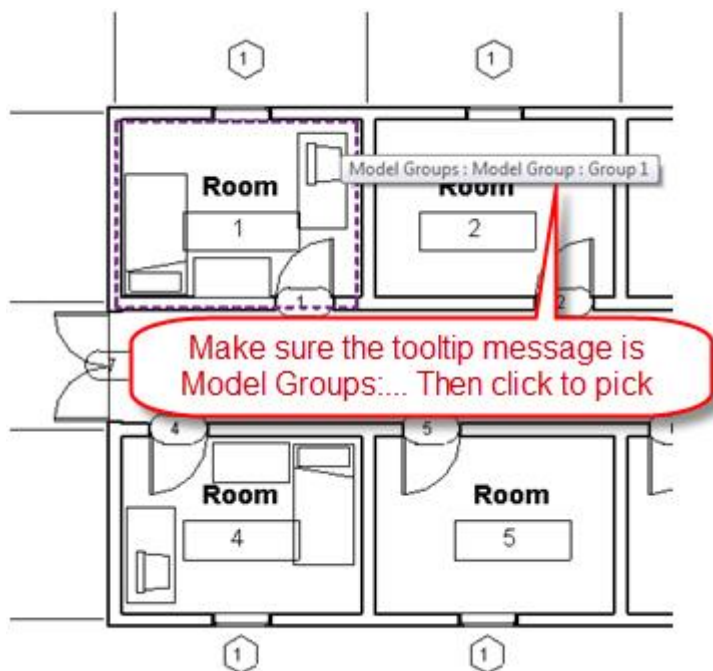
Чтобы увидеть, как работает наш плагин, давайте запустим его.

Запуск плагина

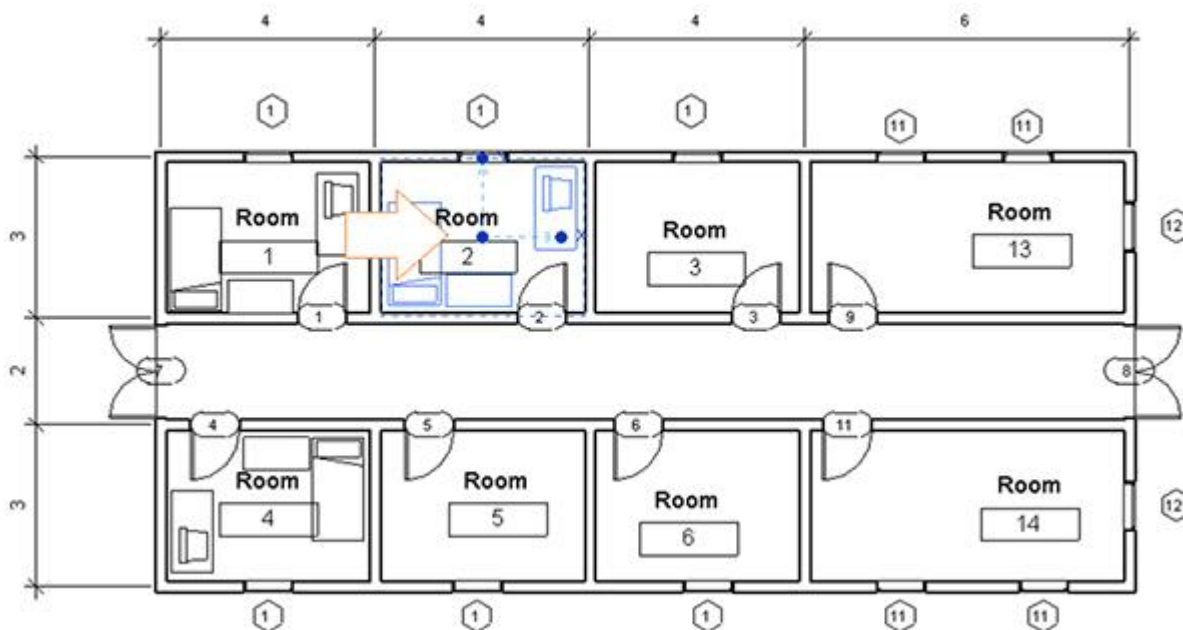
1. Запустите Autodesk Revit Architecture.
2. Откройте проект [Hotel.rvt](#).
3. **Загрузите плагин в Revit и запустите его:**
В Revit, на вкладке **Надстройки (Add-Ins)** ленты щелкните на кнопке **Внешние инструменты (External Tools)**. Из выпадающего списка выберите **Lab1PlaceGroup**. Это действие запускает наш плагин.



4. **Поработайте с плагином.**
Поместите курсор в пределы комнаты **Room1** в рабочей области Revit. Когда курсор окажется наведен на группу с мебелью, ограничивающая рамка подсветится (как показано на рисунке ниже) и появится всплывающая подсказка **Model Groups: Model Group: Group 1**. Щелкните левой кнопкой мыши, чтобы выбрать эту группу. (Имейте в виду, что подсвеченные комнаты выглядят так же, как и группы, поэтому внимательно проверяйте содержимое всплывающей подсказки. Если будет выбрана комната, то вы не добьетесь нужного результата)



5. Укажите точку в другой комнате, например **Room 2**. Группа должна скопироваться в эту комнату. Центр новой группы размещается в точке, которую вы указали.



Поздравляем! Вы только что написали свой первый плагин для Autodesk Revit и поработали с ним. Программный код плагина будет подробно рассмотрен в [уроке 3](#).

Прежде чем перейти к следующим занятиям, давайте коснемся ряда дополнительных тем. Мы обратимся к общим принципам программирования и узнаем какую пользу это может принести проектировщику.

Дополнительные темы

Введение в программирование

Код C#, который мы написали, скомпилировали и выполнили, состоит всего из 30 строк. Этот небольшой программный фрагмент работает по тому же принципу, что и встроенная команда Revit Создать аналог (Create Similar). Программирование позволяет, поняв однажды логику работы конкретной функции, снова и снова обращаться к ней, когда возникает необходимость.

Что такое программирование?

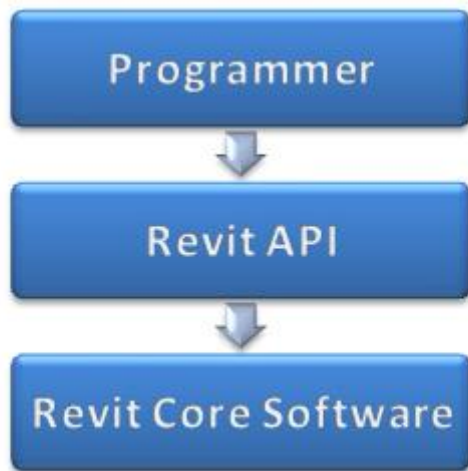
Самый простой ответ на этот вопрос: программирование – это процесс создания цепочки инструкций, заставляющих компьютер сделать что-либо. Воспринимайте свои программы как набор таких инструкций. На следующих занятиях вы столкнетесь с различными строками и целыми блоками кода. Смотрите на них в контексте того, что они являются инструкциями для компьютера.

Если бы вам понадобилось объяснить, что такое компьютеры, маленькому ребенку, вы могли сказать так: компьютер – это инструмент, которое следует твоим инструкциям. Программирование – это один из способов инструктирования компьютера. На самом деле, компьютер видит эти кодированные инструкции как серии цифр (так называемый машинный код). Понятные для человека инструкции, которые вы видели в начале этого занятия, называются исходным кодом; компьютер конвертирует эти инструкции в машинный код, который он может читать и выполнять. Последовательность таких инструкций (или кода), предназначенная для выполнения определенной задачи, называется программой, а совокупность таких программ и связанных с ними данных – программным продуктом. Одним из реальных примеров программного продукта является Autodesk Revit.

Исходный код может быть написан на разных языках, по аналогии с тем, как люди используют разные языки для общения между собой. Язык, который вы будете использовать на занятиях этой серии, называется C# (читается как «Си-шарп»).

Что такое API?

API – это сокращение от Application Programming Interface (интерфейс программирования приложений). API является средством взаимодействия программиста с программным продуктом. Revit API, например, предоставляет программистам доступ к функциональным возможностям Revit и позволяет писать инструкции, которые Revit будет выполнять по порядку.



Взглянем на процесс немного с другой стороны. Коммерческие компании-разработчики программного обеспечения, такие как Autodesk, часто предоставляют наборы библиотек, которые можно применять в пользовательских приложениях для взаимодействия с программными продуктами (Autodesk Revit и др.), расширяя таким образом их функциональность. Такие наборы библиотек называются API программных продуктов.

Техника написания приложений, создаваемых для расширения функциональности программного продукта, зависит от структуры API и от перечня функций, доступ к которым он предоставляет.

Что такое плагины?

Плагины – это программные модули или файлы, которые добавляют дополнительные функциональные возможности в целевой продукт (обычно в виде команд, автоматизирующих некоторые действия). Когда мы говорим о плагинах для Revit (для них применяются также термины «надстройка», «дополнение» и «подключаемый модуль»), мы имеем в виду модули, содержащие код, который взаимодействует с Revit API. Revit загружает плагины и использует их в определенных ситуациях – например, когда пользователь вызывает ту или иную команду.