

cad二次开发

所需要的知识准备

1. C#基本语法入门
 2. 面向对象的编程概念
-

环境搭建

软件准备

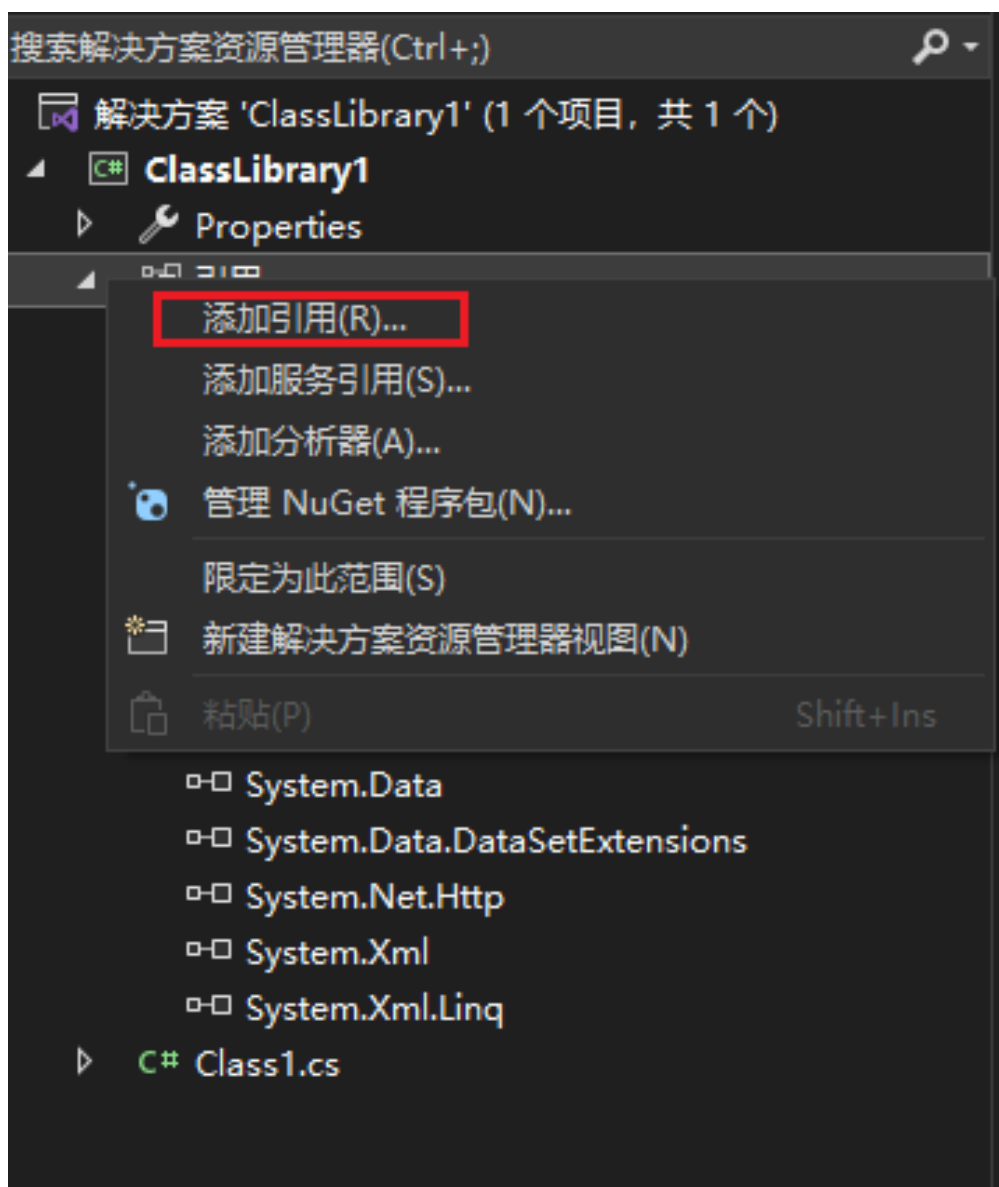
1. 下载Visual Studio(VS)并安装：[Visual Studio下载](#)
2. 安装.net framework
 - 注意要运行插件的cad软件版本和.net的版本对应关系：[cad .net版本对应](#)

建立项目

1. VS新建库项目



2. 根据程序所需要调用的API，引入cad提供的.net API文件，它们为cad程序的文件根目录下的DLL文件



以下4个常用：

名称	作用	备注
AcDbMgd.dll	处理图形文件中存储的对象	
AcMgd.dll	处理AutoCAD应用程序和用户接口	
AcCui.dll	处理自定义文件	
AcCoreMgd.dll	处理编辑器、发布与打印、定义AutoLISP命令和函数	AutoCAD 2014 版

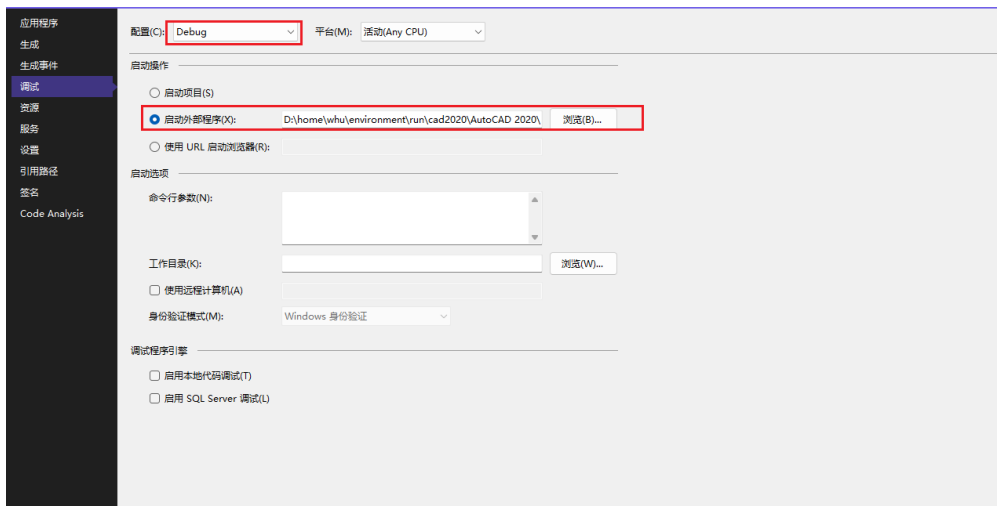
- [主要API的官方介绍](#)

3. 还可根据其他需要引入相应的库，如常用的文件选择窗口，需引入

`System.Windows.Forms` 库

4. 双击此处更改项目属性，在 `调试` 中启动外部cad程序





5. 在源代码(.cs 文件)中引入Cad相关API

```
using Autodesk.AutoCAD.EditorInput;
using Autodesk.AutoCAD.Runtime;
using Autodesk.AutoCAD.DatabaseServices;
using Autodesk.AutoCAD.ApplicationServices;
using Application =
Autodesk.AutoCAD.ApplicationServices.Application;
```

6. 在程序中，命令所执行的程序以函数的形式编写，命令在函数上方以

[CommandMethod("commandName")] 的形式定义

调试和编译

1. 调试：保存代码后，点击启动，即可打开Cad程序，之后可用 netload 加载 项目根目录/bin/debug 下的dll文件，然后运行命令进行测试



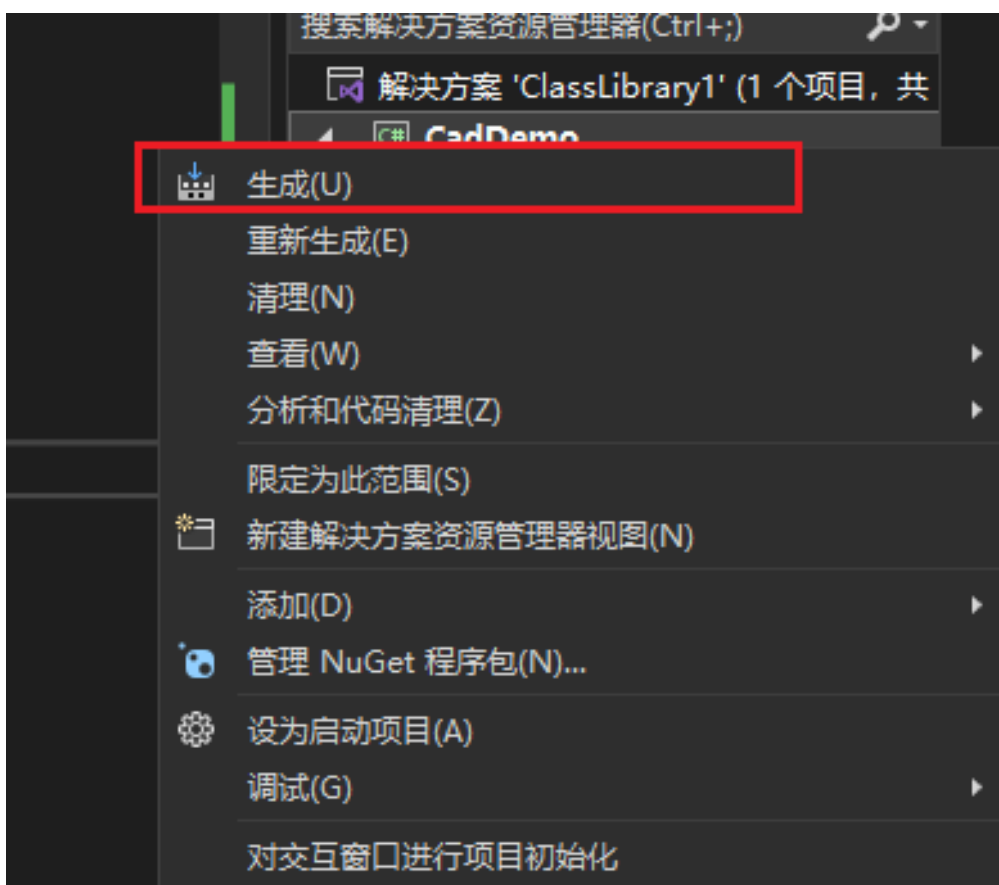
2. 修改代码后，可使用热部署，将代码更改应用于程序，避免反复启动Cad



3. 调试完成后，点击此处结束程序



4. 编写完成后，可使用 生成 构建程序文件



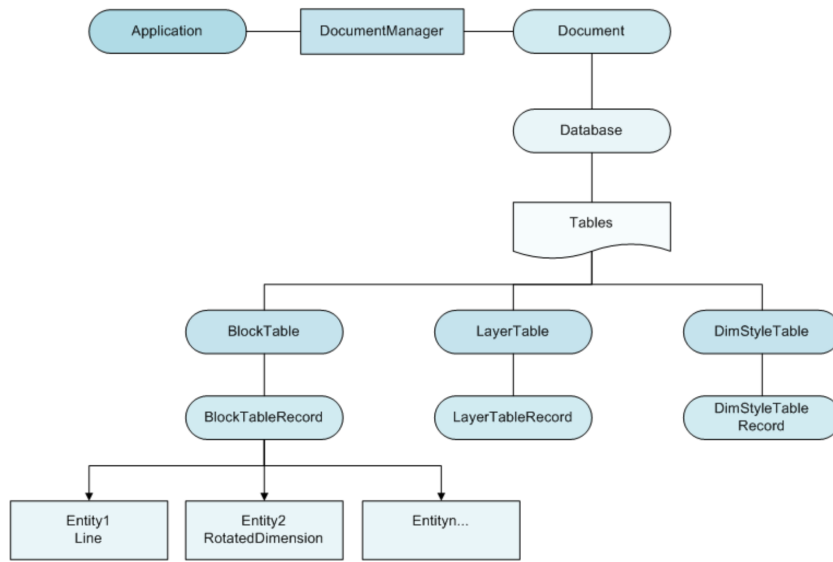
理解Cad二次开发的API

Cad对象模型

Cad API将所有内容均抽象成对象，可理解为：cad中的数据 and 操作均包裹在“对象（Object）”这一容器中，通过对应API访问对象的属性和方法，实现对应的功能

- 例如，整个Cad应用程序就是一个对象，其中的一个文档也是一个对象，绘图区中的一个长方形、一条线等各种图形也都是一个个对象
- 对象有层级关系，整体对象包含局部对象，如Cad应用程序对象中有文档对象、菜单对象、打印机对象等，一个文档对象中有（若干个）图形对象

对象层级关系总览：



了解Cad对象层级

程序案例

实现一个最简单的Cad功能，加载此脚本的dll后，在命令行输入 `demo`，会打印 `hello cad`

```

// 引入依赖
using Autodesk.AutoCAD.EditorInput;
using Autodesk.AutoCAD.Runtime;
using Application = Autodesk.AutoCAD.ApplicationServices.Application;

namespace CadDemo
{

```

```
public class CadDemo
{
    // 功能函数
    [CommandMethod("demo")] // 通过`demo`命令调用这个功能
    public static void Demo()
    {
        string str = "hello cad"; // 创建字符串变量，值为`hello cad`
        Editor ed =
        Application.DocumentManager.MdiActiveDocument.Editor; // 通过Cad提供的
        API拿到活动文档的命令编辑器对象
        ed.WriteMessage(str); // 调用命令编辑器的WriteMessage方法，
        向其写入字符串
    }
}
```

参考资料

相关资源

[cad .net版本对应](#)

[cad2020 .net api](#)

[cad二次开发步骤](#)

[官方教程资源及代码](#)

写入文件的代码

```
Class1.cs ×
_16与外部文件_txt交互.Class1 DataToTxtDemo()

11 namespace _16与外部文件_txt交互
12 {
13     0 个引用
14     public class Class1
15     {
16         [CommandMethod("DataToTxtDemo")]
17         0 个引用
18         public void DataToTxtDemo()
19         {
20             Database db = HostApplicationServices.WorkingDatabase;
21             Wnd.SaveFileDialog saveDlg = new Wnd.SaveFileDialog();
22             saveDlg.Title = "保存图形数据";
23             saveDlg.Filter = "文本文件(*.txt)|*.txt";
24             saveDlg.InitialDirectory = Path.GetDirectoryName(db.FileName);
25             string fileName = Path.GetFileName(db.FileName);
26             saveDlg.FileName = fileName.Substring(0, fileName.IndexOf('.'));
27             Wnd.DialogResult saveDlgRes = saveDlg.ShowDialog();
28             if (saveDlgRes == Wnd.DialogResult.OK)
29             {
30                 string[] contents = new string[] { "1111111", "22222" };
31                 File.WriteAllLines(saveDlg.FileName, contents);
32             }
33         }
34     }
35 }
36
```