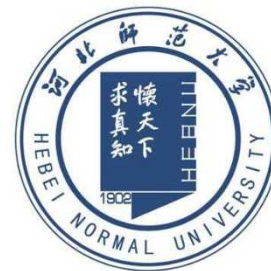


GIS设计与开发

资源与环境科学学院 B521室 刘强

邮箱: liuqiangjp@163.com

手机号: 15733175965

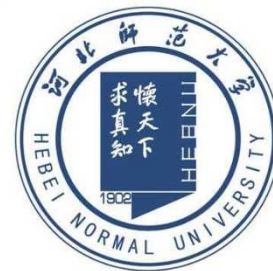


ArcGIS

- ArcGIS是ESRI在全面整合了GIS与数据库、软件工程、人工智能、网络技术及其它多方面的计算机主流技术之后，成功地推出了代表GIS最高技术水平的全系列GIS产品。ArcGIS是一个全面的，可伸缩的GIS平台，为用户构建一个完善的GIS系统提供完整的解决方案。ArcGIS的基本体系能够让用户在任何需要的地方部署GIS功能和业务逻辑，无论是在桌面、服务器、还是在野外。

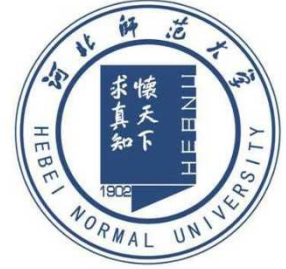
ArcGIS软件体系结构





基于 ArcGIS产品的开发方式

- 配置/ 定制ArcGIS Desktop产品；
- 扩展ArcGIS结构与数据模型；
- 使用ArcGIS Engine在其他应用程序中嵌入GIS功能；
- 使用ArcGIS Engine构建自定义桌面应用程序；
- 使用ArcGIS Server构建网络服务和网络应用程序。

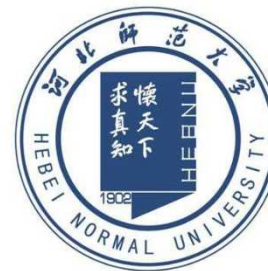


ArcGIS Engine

- ArcGIS Engine 是一组完备的并且打包的嵌入式GIS 组件库和工具库，开发人员可用来创建新的或扩展已有的桌面应用程序。使用ArcGIS Engine，开发人员可以将GIS 功能嵌入到已有的应用软件中，如自定义行业专用产品；或嵌入到商业生产应用软件中，如Microsoft Word 和Excel；还可以创建集中式自定义应用软件，并将其发送给机构内的多个用户。
 - 在ArcObjects基础之上封装的一组高级接口
 - 核心构建在ArcObjects之上，完全可扩展
- Developers use ArcGIS Engine to deploy GIS data, maps, and geoprocessing scripts in desktop or mobile applications using application programming interfaces (APIs) for COM, .NET, Java, and C++.



GIS客户端可以从简单的浏览器访问过渡到专业的GIS桌面端，ArcGIS Engine为需要访问有丰富GIS功能的最终用户提供了能够将GIS逻辑集成到中间件应用的能力。



ArcObjects

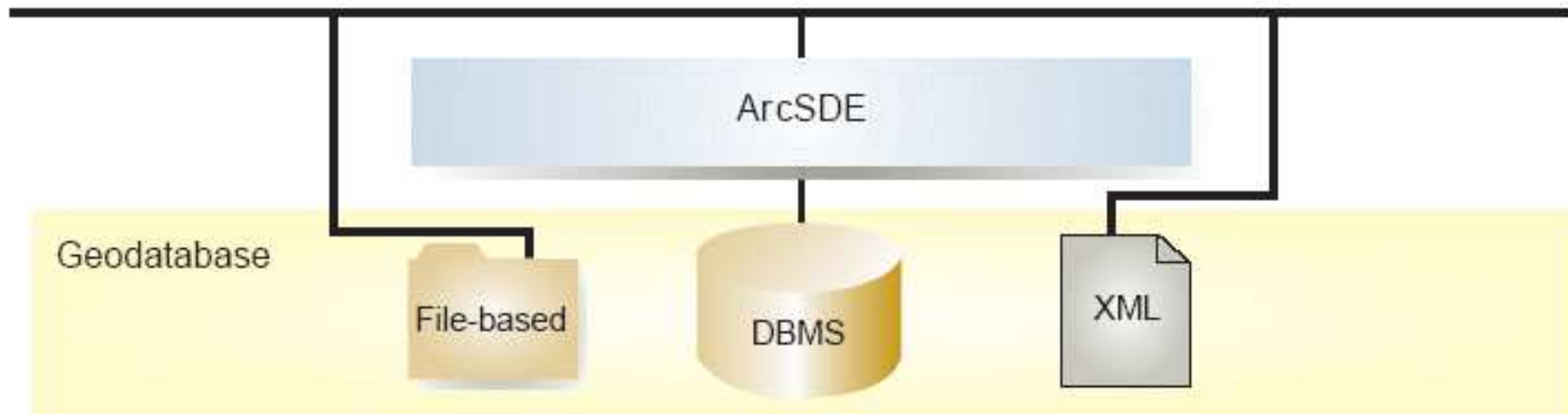
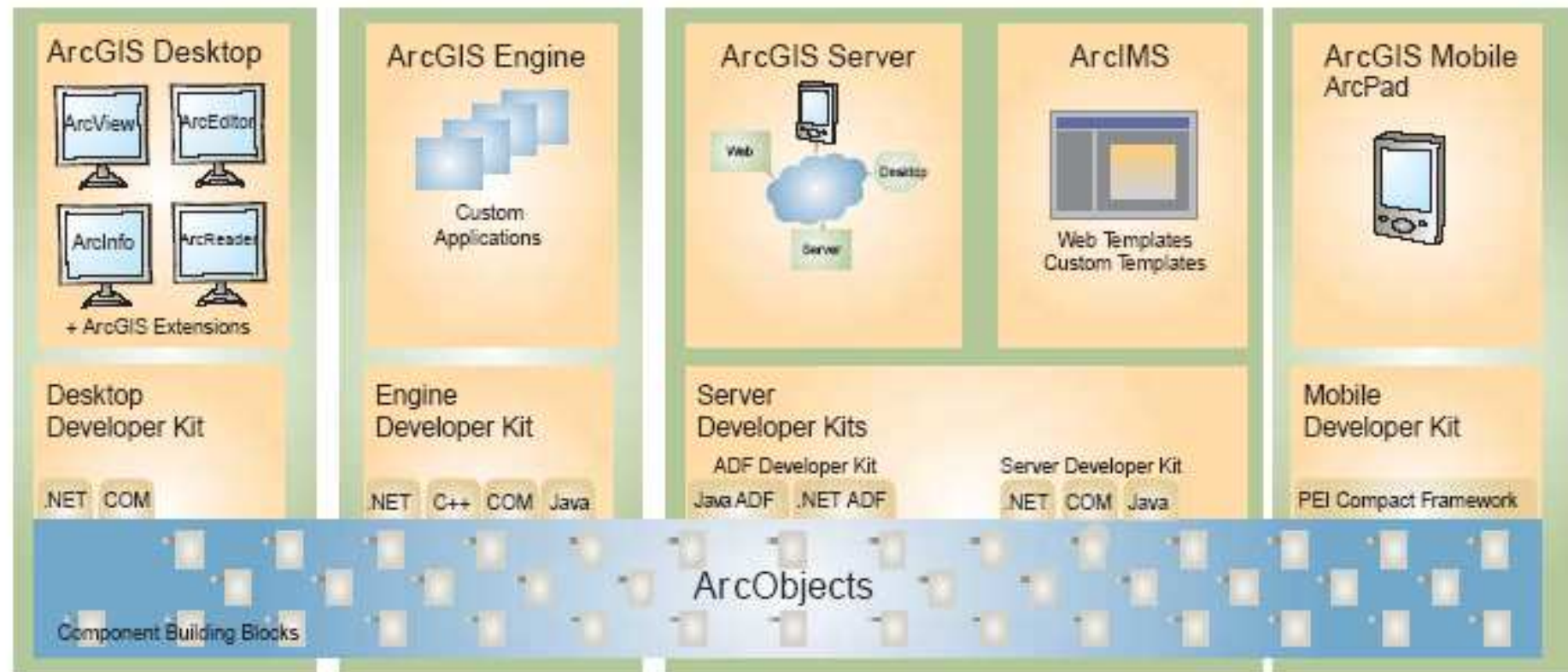
- ArcObjects是 ESRI公司ArcGIS系列产品的开发平台，它是基于Microsoft COM技术所构建的GIS组件产品，是一套可重用的通用的二次开发组件产品，它可以用于大量开发框架中，包括流行的像 .NET、Visual C++、Java等开发环境。
- ArcObjects不是为最终用户而是专门为开发人员提供的二次开发组件。
- ArcObjects是ArcGIS系列产品的基础，大部分ArcGIS产品都是由ArcObjects构建的。

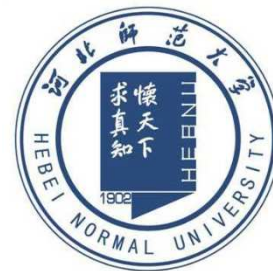
Desktop GIS

Embedded GIS

Server GIS

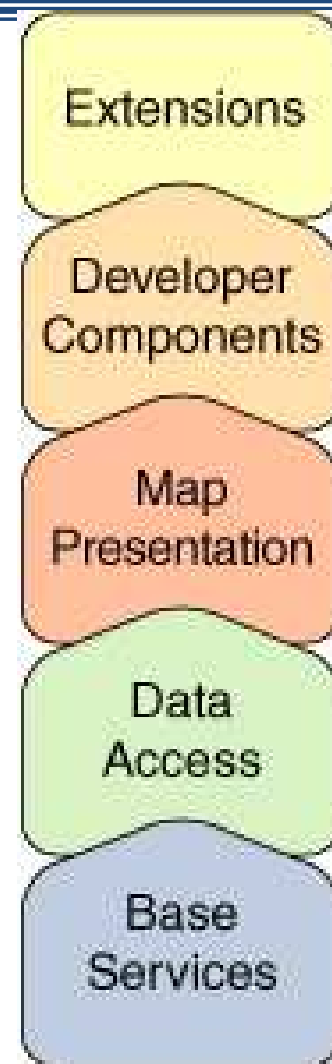
Mobile GIS

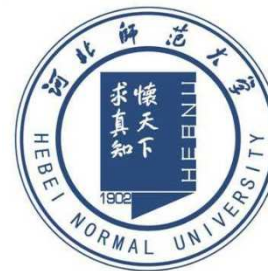




ArcGIS Engine的逻辑体系结构

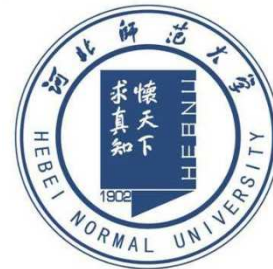
- 基本服务
 - 核心的功能，几何图形的操作、 图形编辑
- 数据访问
 - 访问矢量和栅格数据
- 地图分析
 - 高级的空间和SQL查询、空间分析
- 地图展现
 - 标注、符号化、专题图渲染、动态地图
- 开发组件
 - 开发包，地图控件，制图控件和3D控件，工具条，TOC





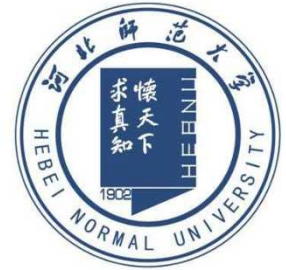
你能构建哪些类型的应用程序?

- 可视化独立应用程序
 - **Mapping, drawing, selecting, editing, ...**
- 非可视化独立应用程序
 - **Utility, editing, updating, ...**
- 嵌入式



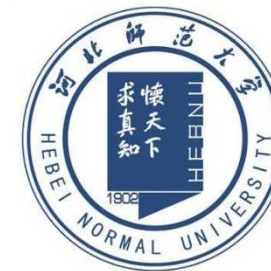
ArcGIS Engine 开发过程

- 开发步骤
 - 1) 准备数据
 - 2) 确定应用程序的类型
 - 3) 实现软件授权编码
 - 4) 应用程序功能的开发实现
 - 5) 配置和发布



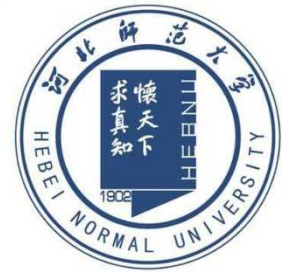
AE VS AO

- ArcGIS Engine 是一组完备的并且打包的嵌入式GIS 组件库和工具库，开发人员可用来创建新的或扩展已有的桌面应用程序。
 - 在ArcObjects基础之上封装的一组高级接口
 - 核心构建在ArcObjects之上，完全可扩展
- ArcObjects是 ESRI公司ArcGIS系列产品的开发平台，它是基于Microsoft COM技术所构建的GIS组件产品，是一套可重用的通用的二次开发组件产品，它可以用于大量开发框架中，包括流行的像 .NET、Visual C++、Java等开发环境。
- 利用ArcObjects开发的软件产品，必须要运行在安装好了ArcGIS Desktop 软件的环境中；利用ArcGIS Engine开发的软件，只需ArcGIS Engine（Runtime）。



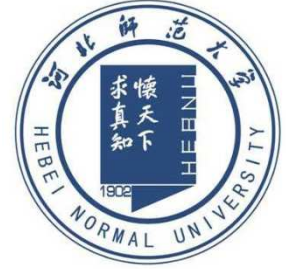
ArcObjects的组件库

	Enums	Structs	Interfaces	Classes	Total
ArcGIS 9.0	628	35	3029	2322	6014
ArcGIS 9.1	933	36	3918	3043	7930
ArcGIS 9.2	1100	40	4798	3832	9770
ArcGIS 9.3	1195	51	5206	4050	10502
ArcGIS 10.0	1053	93	5555	4081	10782
ArcGIS 10.1	1132	94	5957	4419	11602
ArcGIS 10.2	1140	94	5997	4445	11676



ArcGIS Engine的组件库

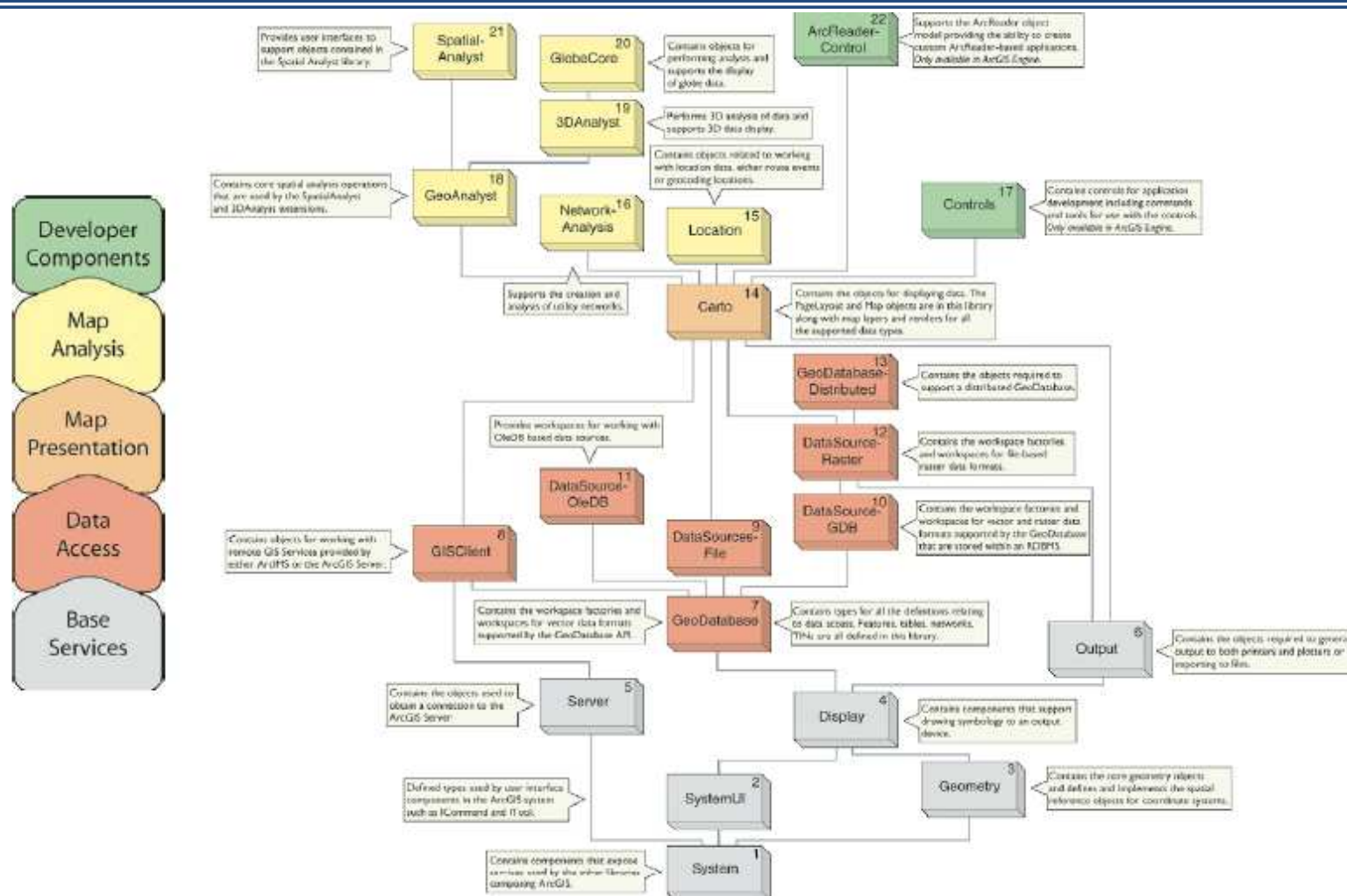
- ArcGIS Engine开发中，为了更好地管理COM对象，ESRI将这些COM对象放在不同的组件库中
- 在.NET环境中，一个组件库对应一个assembly（程序集），assembly可以有多个不同的命名空间。
 - ESRI.ArcGIS.ADF Assembly 的Namespaces:
 - ESRI.ArcGIS.ADF
 - ESRI.ArcGIS.ADF.BaseClasses
 - ESRI.ArcGIS.ADF.CATIDs
 - ESRI.ArcGIS.ADF.COMSupport
 - ESRI.ArcGIS.ADF.Resources

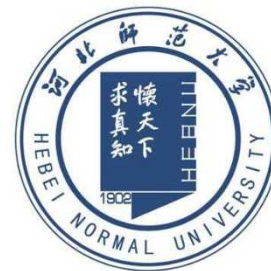


ArcObjects Assembly

- ArcObjects 10.2的组件库共75个
(75 assemblies)
- Shared with ArcGIS for Desktop, ArcGIS Engine for Windows, and ArcGIS for Server (AE, 35)
- Only available in ArcGIS for Desktop (30)

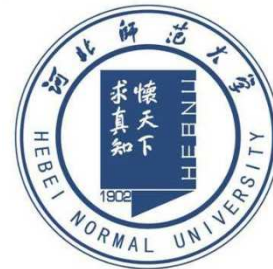
ArcGIS Engine的主要共享程序集





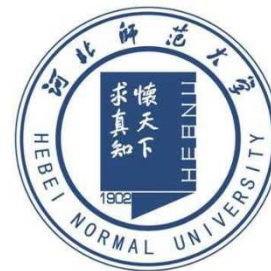
ArcGIS Engine的共享程序集介绍

- **Version程序集**
 - Version是ArcGIS 10新出来的一个程序集，该程序集包含了将独立应用程序绑定到特定的ArcGIS 系列产品的函数和方法，该程序集是在运行Engine的应用程序的时候。



系统、框架及控件程序集

- **System**
 - System程序集是ArcGIS框架中最底层的一个程序集，它包含了一些被其他程序集使用的一些基础组件。
- **SystemUI**
 - SystemUI程序集定义了一些在ArcGIS中被用户界面组件使用的类型。象ICommand和ITool这样的接口就在这里定义。
- **ADF (Application Developer Framework)**
 - ADF including [ADF](#), [ADF.Local](#), [ADF.Core](#), [ADF.Connection](#), [ADF.Connection.Local](#), and [ADF.Connection.Core](#)。提供了.Net环境下基础类、组件服务、组件ID以及一些实用的类，如Convertor、ComReleaser等。
- **Controls**
 - 控件程序集包含了应用程序开发中用到的控件，包括在控件中使用的命令和工具。



要素及要素可视化程序集

- **Carto**

- 包含了为数据显示服务的对象。**PageLayout**和**Map**对象在这个程序集中，以及支持各种数据类型的图层，渲染。这个程序集中也包括了**MxdServer**和**MapServer**对象，它们通常被**Server**用来显示地图数据。

- **Geometry**

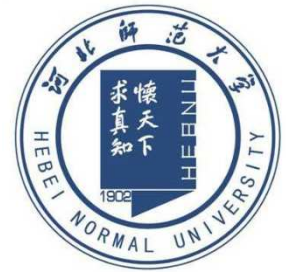
- 几何程序集包含了核心的几何对象，比如点，面，线等。在几何程序集中还定义和实现了空间参考的对象，包括投影坐标系和几何坐标系。

- **Display**

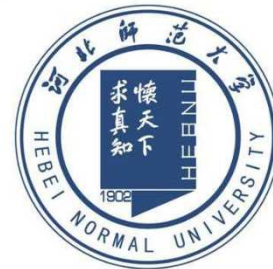
- 显示程序集包含了支持在输出设备上显示图形的组件，屏幕显示、符号、颜色、显示交互（**Feedback**）等都在这个程序集中定义。

- **Output**

- 提供了将**MapControl**和**PagelayoutControl**中的地图输出到打印机以及其他设备文件中。

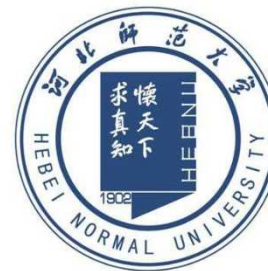


- Animation
 - ArcGIS动画框架，提供了在Map、Scene以及Globe中支持动画功能。
- Maplex
 - 提供了在Map中高级标注功能扩展。



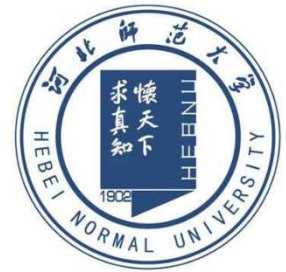
Web程序集

- **GISClient**
 - 包含了作用于远程**GIS**服务的对象。这些远程服务可以由**ArcGIS Server**和**ArcIMS**提供。
- **Server**
 - 提供了连接、管理**ArcGIS for Server**的核心类，如**IServerObjectManager**、**IServerObject**等，其他程序集的对象可以通过**ArcGIS Server**的**IServerContext**进行远程创建。



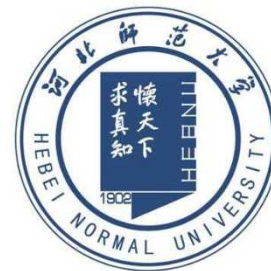
数据及数据访问程序集

- **DataInterop**
 - 数据互操作模块，提供了在Engine中直接访问70多种数据的接口。
- **Geodatabase**
 - 包含了所有相关数据组织的定义类型。要素、表、网络、TIN、栅格数据集、栅格目录、要素子类、拓扑模型都在这个程序集中定义。其他一些类型的实现在各自的数据来源的程序集中。
- **DataSourcesFile**
 - DataSourcesFile程序集包含了为支持的矢量数据格式提供的工作空间工厂和工作空间。
- **DataSourcesGDB**
 - DataSourcesGDB 程序集包含了为存储在RDBMS（ Relational database management system ）中的矢量和栅格数据提供的工作空间工厂和工作空间。
- **DataSourcesOleDb**
 - DataSourcesOleDb 程序集为通过OleDb（对象链接与嵌入）方式提供的数据库提供空组空间。



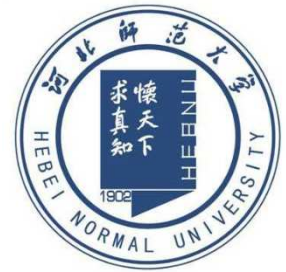
数据及数据访问程序集

- **DataSourcesRaster**
 - DataSourcesRaster 包含了为基于文件方式的栅格数据提供的工作空间工厂和工作空间。
- [DataSourcesNetCDF](#)
 - [DataSourcesNetCDF](#) 程序集 contains objects used for accessing network Common Data Form (netCDF) files and for creating rasters, features, or tables from netCDF variables in ArcGIS.
- **GeoDatabaseDistributed**
 - GeoDatabaseDistributed 程序集包含了需要执行导入导出 (checkout/checkin) 的离线GeoDatabase的对象。
- [GeoDatabaseExtensions](#)
 - [GeoDatabaseExtensions](#) 程序集 contains objects to aid in the creation, management, and use of terrain datasets and cadastral fabrics.



数据分析程序集

- [DefenseSolutions](#)
 - [DefenseSolutions](#) 程序集包含为制作军事防御应用，有军事符号、坐标转换、制作军用地理几何对象和其他国防工具。
- [NetworkAnalysis](#)
 - [NetworkAnalysis](#) 程序集用于网络分析。
- [NetworkAnalyst](#)
 - [NetworkAnalyst](#) 程序集提供网络分析的数据集对象，以支持网络分析。
- [Location](#)
 - [Location](#) 包含位置相关的数据对象，可以是路径时间或者地理编码。



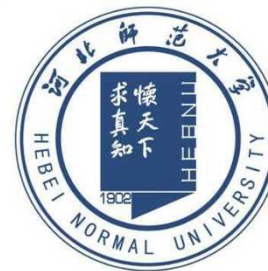
数据分析程序集

- **GeoAnalyst**
 - GeoAnalyst程序集包含了核心的空间分析功能，这些功能是在spatialAnalyst和3DAnalyst扩展都会用到的功能。
- **3DAnalyst**
 - 3DAnalyst程序集包含了数据的3d分析对象，也包括显示3d数据，在这个程序集中有一个控件SceneControl可用。
- **GlobeCore**
 - GlobeCore程序集包含了globe数据分析的对象，也包含了显示globe数据，在这个程序集中有一个控件GlobeControl可用。
- **SpatialAnalyst**
 - SpatialAnalyst程序集包含了在栅格和矢量数据上执行空间分析的对象。



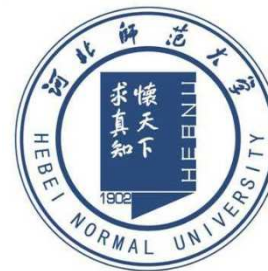
数据分析程序集

- Geoprocessing
 - 提供了在Engine中访问GP功能的核心类，.Net用户请使用Geoprocessor程序集，它提供了访问GP的简单工作流。
- The [TrackingAnalyst](#) assembly implements the non-user interface functionality of the ArcGIS Tracking Analyst extension to ArcMap. The Tracking Analyst extension supports the display, analysis, and manipulation of temporal data within ArcGIS.
- The [EngineCore](#) assembly contains objects that allow developers to build and manage collections of graphics in a map or globe.



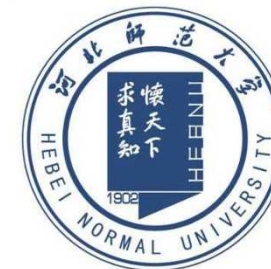
数据分析程序集

- Schematic
 - 提供了在Engine中访问ArcGIS Schematic扩展的核心类，构建逻辑拓扑图。
- The [SchematicControls](#) assembly provides a set of commands that work with the ArcGIS Engine controls to enable building Schematic applications.



ArcGIS Resources

- 学习 ArcGIS系列产品开发方式的最直接准确的资源：<http://resources.arcgis.com/zh-cn/help/>
 - 应用程序帮助
 - 开发者帮助
 - 其他帮助



resources.arcgis.com/zh-cn/help/ 百度 <Ctrl+K>

访问最多 火狐官方网站 新手上路

ArcGIS Resources

ArcGIS Online Sign In English

Home Communities **Help** Blog Forums Videos

搜索 ArcGIS R

帮助

ArcGIS 10.2 帮助

(包括10.2.1和10.2.2帮助)

- 入门
- 10.2.2 | 10.2.1 | 10.2 的新特性
- 扩展模块
- 地理数据
- Desktop
- Server
- Mobile
- Online*
- Portal

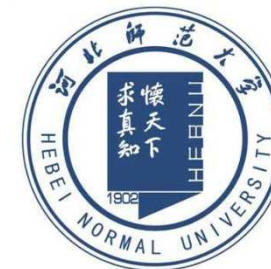
应用程序帮助

Web 应用程序 移动应用程序

查找 Arc

ArcGIS 入门

观看视频、执行简单任务，了解 ArcGIS 平台入门。



开发者帮助*

本地 SDK (Runtime)

- [Android*](#)
- [iOS*](#)
- [Java*](#)
- [Mac OS X*](#)
- [.NET*](#)
- [Qt*](#)
- [Windows Mobile*](#)
- [Windows Phone*](#)
- [Windows \(WPF\)*](#)

Web API

- [ArcGIS API for JavaScript*](#)
- [ArcGIS API for Flex*](#)
- [ArcGIS API for Silverlight*](#)

其他 SDK

- [Web ADF \(.NET\) : Web 应用程序帮助](#)
- [Web ADF \(Java\) : Web 应用程序帮助](#)
- [Web ADF \(.NET\) : 开发者帮助*](#)

服务 API

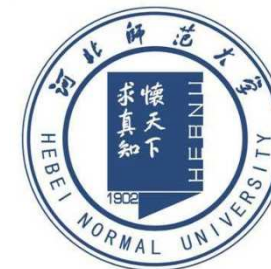
ArcGIS REST API*

- [来自 Esri 的服务*](#)
- [您自己的服务*](#)
- [管理您的组织*](#)
- [管理您的服务器*](#)
- [管理 ArcGIS Online 托管服务*](#)
- [Web 地图 JSON 格式*](#)

ArcGIS SOAP API*

ArcObjects

- [ArcObjects .NET \(包括 VC++\) *](#)
- [ArcObjects Cross-platform C++*](#)
- [ArcObjects Java*](#)



ArcObjects Help for .NET

resources.arcgis.com/en/help/arcobjects-net/conceptualhelp/

ArcGIS Resources

ArcGIS Online

esri

Home Communities Help Blog Forums Videos

Search ArcObjects .NET

ArcObjects Help for .NET

Resource Center

FEEDBACK | PRINT | EMAIL

Concepts & Samples API Reference

ArcObjects Help for .NET developers

ArcObjects Help for .NET developers

Tips for using the ArcObjects .NET SD

What's new for developers at 10.x

What's new for developers at 10.x

What's new for developers at 10.x

What's new at 10?

Developing with ArcGIS

Developing with ArcGIS

Building add-ins for ArcGIS for D

Building stand-alone application

Extending ArcGIS Server

Learning ArcObjects

Visual Studio integration

Migrating your code

Migrating your code

Migrating your code from 9.3 to 1

Migrating from VB6 to VB.NET

How to migrate from COM comp

ArcGIS developer products

ArcGIS developer products

ArcGIS Engine

ArcGIS for Desktop (Basic, Stand

ArcGIS for Server

ArcObjects namespaces

Appendix: ArcObjects in Visual C++

Copyright information

ArcObjects Help for .NET developers

About the ArcObjects Help for .NET developers

ArcObjects is a library of Component Object Model (COM) components that make up the foundation of ArcGIS. To work with ArcObjects in the .NET development environment, a collection of .NET primary interop assemblies (PIAs) and COM object libraries have been generated to manage .NET-COM interaction. ArcObjects components are installed with ArcGIS for Desktop, ArcGIS Engine, and ArcGIS for Server and can be used in the following ways:

- To customize ArcGIS for Desktop applications
- To build stand-alone mapping applications
- To develop Web applications

The software development kit (SDK) for developing with ArcObjects in .NET assists you in writing your applications by providing best practices, conceptual documentation, code sections, and samples.

If you are new to ArcObjects development, first determine what type of application you need to build to address your problem. This topic presents the different development options and points you in the right direction. Once you know what type of application you are going to create, the [Learning ArcObjects](#) topic provides a foundation as you begin to build ArcObjects solutions.

Developers with existing solutions built on earlier releases of ArcGIS should read [What's new for developers at 10.2?](#), [What's new for developers at 10.1?](#), [What's new for developers at 10?](#), and the section on [Migrating your code](#).

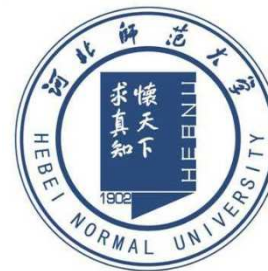
Although this SDK focuses on C# and VB.NET development, it also contains resources for the VC++ developer. To access those resources, select them as a part of your SDK installation. If you did not do so, you can modify your installation and select the VC++ option to get additional Help topics and samples.

Customizing ArcGIS for Desktop

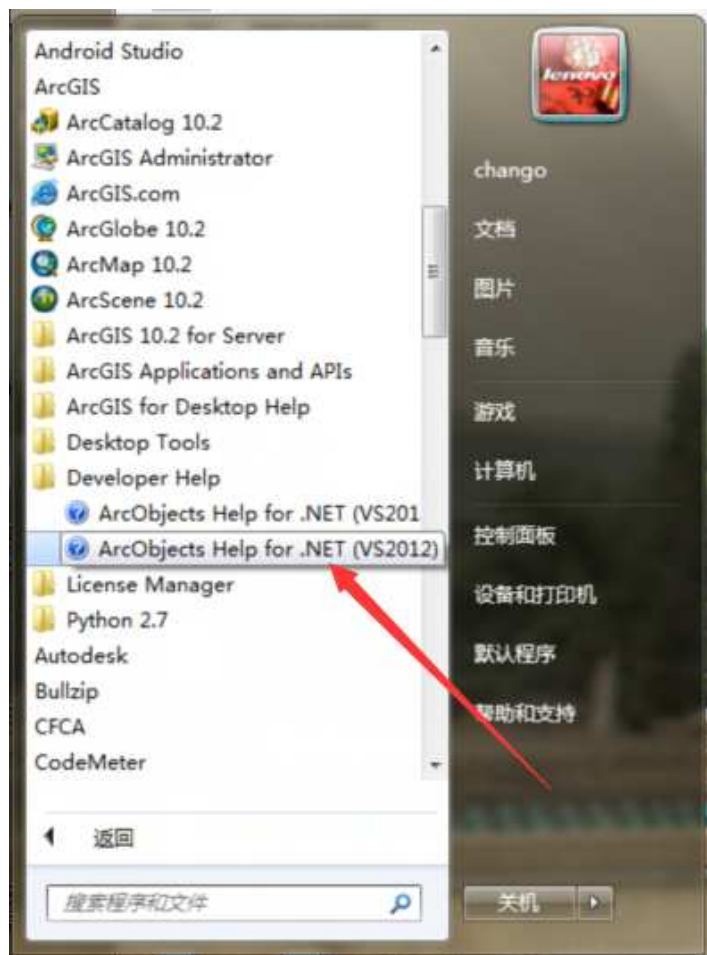
ArcGIS for Desktop is a suite of integrated applications including ArcMap, ArcGlobe, ArcScene, and ArcCatalog. ArcGIS for Desktop applications allow you to perform geographic information system (GIS) tasks from simple to advanced, including mapping, geographic analysis, data compilation, data management, visualization, and geoprocessing. The ArcGIS for Desktop Help system provides topics about these applications.

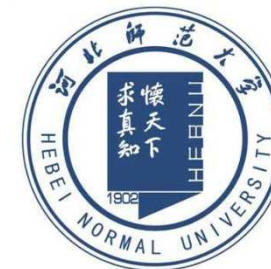
ArcGIS for Desktop applications can be customized to automate GIS tasks or create unique workflows and environments suitable for your organization. The following are the ways you can customize ArcGIS for Desktop applications:

- Configure the application user interface (UI), such as rearranging toolbars, tools, commands, and dockable windows



本地帮助





Microsoft Help Viewer 2.0 - Visual Studio 2012 文档

目录(C)

筛选内容

查看器主页

ET Framework 4.5

end for Visual Studio 入门

sual Studio 2012

sual Basic 和 Visual C#

用程序生命周期管理与 Visual Studio 和 Team

sual Studio 中的 Office 开发

icrosoft Help Viewer 2.0 帮助

cObjects SDK for ArcGIS 10.2 - 英语

ArcObjects Help for .NET developers

Tips for using the ArcObjects .NET SDK

▸ What's new for developers at 10.x

▾ Developing with ArcGIS

▸ Building add-ins for ArcGIS for Des

▸ Building stand-alone applications

▸ Server object extensions

▸ Learning ArcObjects

▾ Visual Studio integration

ArcGIS Visual Studio IDE Integration

▸ ArcGIS Visual Studio IDE Integration

Add ArcGIS Reference dialog box

Snippets for ArcGIS .NET develop

▸ ArcGIS Snippet Finder

▸ ArcGIS Snippet Editor

Common issues with IntelliSense and

▾ Migrating your code

▸ Migrating your code from 9.3 to 10.

▸ Migrating from VB6 to VB.NET

How to migrate from COM compon

▾ ArcGIS developer products

ArcGIS Engine

ArcGIS for Desktop (Basic, Standard

ArcGIS for Server

Copyright information

ArcObjects reference

目录(C) 索引(I) 收藏夹(F) 搜索(S)

Developing with ArcGIS

管理内容(M)

Developing with ArcGIS

Summary

As an ArcGIS developer, you can develop several different customizations; that is, add-ins, stand-alone applications, and custom components. This topic helps you determine the type of development that is right for your customization.

In this topic

- [About developing with ArcGIS](#)
- [Determining what type of customization to create](#)

About developing with ArcGIS

ArcGIS development, like any development, should follow a plan. The following steps will help you be successful as an ArcGIS developer:

- Define the problem that the programming task will solve**—Geographic information system (GIS) technology is used to answer questions such as "What is the problem that your development is going to solve?"
- Decide on the kind of customization to create**—ArcGIS has various extension points. Before you begin writing your customization, you need to know which of these points you will be targeting. For more information, see the [Determining what type of customization to create](#) section in this topic.
- Locate the correct ArcObjects for the functionality your customization requires**—ArcObjects are the components that provide all the GIS functionality to your customization. There are many ArcObjects, but they are grouped into functional areas. To help determine your starting point and where in the ArcObjects object model you need to focus, see [Learning ArcObjects](#).
Some customizations require you to use multiple areas of ArcObjects. Treat each of these as a sub-task of your customization and focus on one task at a time.
- Develop your customization**—The following steps provide a basic outline for many customizations:
 - Create a Visual Studio project.
 - Reference the required ArcObjects assemblies, identified in [Step 3](#).
 - Create the customization framework by creating your stand-alone application, setting up your add-in and the associated .esriaddin file, or implementing the required ArcObjects interfaces. Use the Visual Studio integrated development environment (IDE) integration utilities. The templates and wizards will help you stub out the basis of your customization type. They are documented in the "Developer utilities" section of this Help system in the "Visual Studio integration" area.
 - Add code, including the GIS functionality through ArcObjects. You will need to license your application to provide access to ArcObjects. Topics on licensing are included in the topics about the different types of customizations. If you are using an extension, see [How to use extensions](#).
 - Compile the source into an executable (EXE) for a stand-alone application, .esriaddin for an add-in, or a dynamic-link library (DLL) for an extension to ArcObjects.
- Test your customization**—Verify that your application works as expected, and make sure that you handle any common user errors appropriately.
- Deploy your customization**—Your completed customization is more beneficial if it can be shared and used by others. Topics on deploying customizations are included in the sections of this Help system about each type of customization. For example, the "Building stand-alone applications" section contains a "Deploying stand-alone applications" subsection.

Determining what type of customization to create

As a developer, you have several development entry points. Once you know the problem that your development is going to solve, you can determine what type of customization to build.

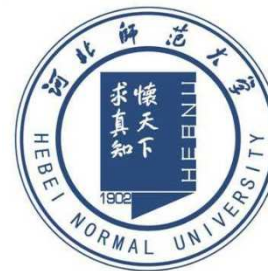
Before you develop an ArcGIS customization, be sure this is the right development path for your project requirements. Other customizations can be developed by customizing the ArcGIS application user interface (UI) or by writing Python scripts and applications. For an introduction to the available functionality through those entry points, and references to more information about them, see [ArcObjects Help for .NET developers](#).

If you want to make your collection of tools on a toolbar, or change the layout of tools and dockable windows in an ArcGIS for Desktop application (ArcMap, ArcCatalog, ArcGlobe, or ArcScene), you can configure the UI to match your requirements. To do this, you can drag and drop within the applications. Since ArcObjects programming is not required, this software development kit (SDK) does not cover this type of customization. For more information, see [About configuring the user interface](#).

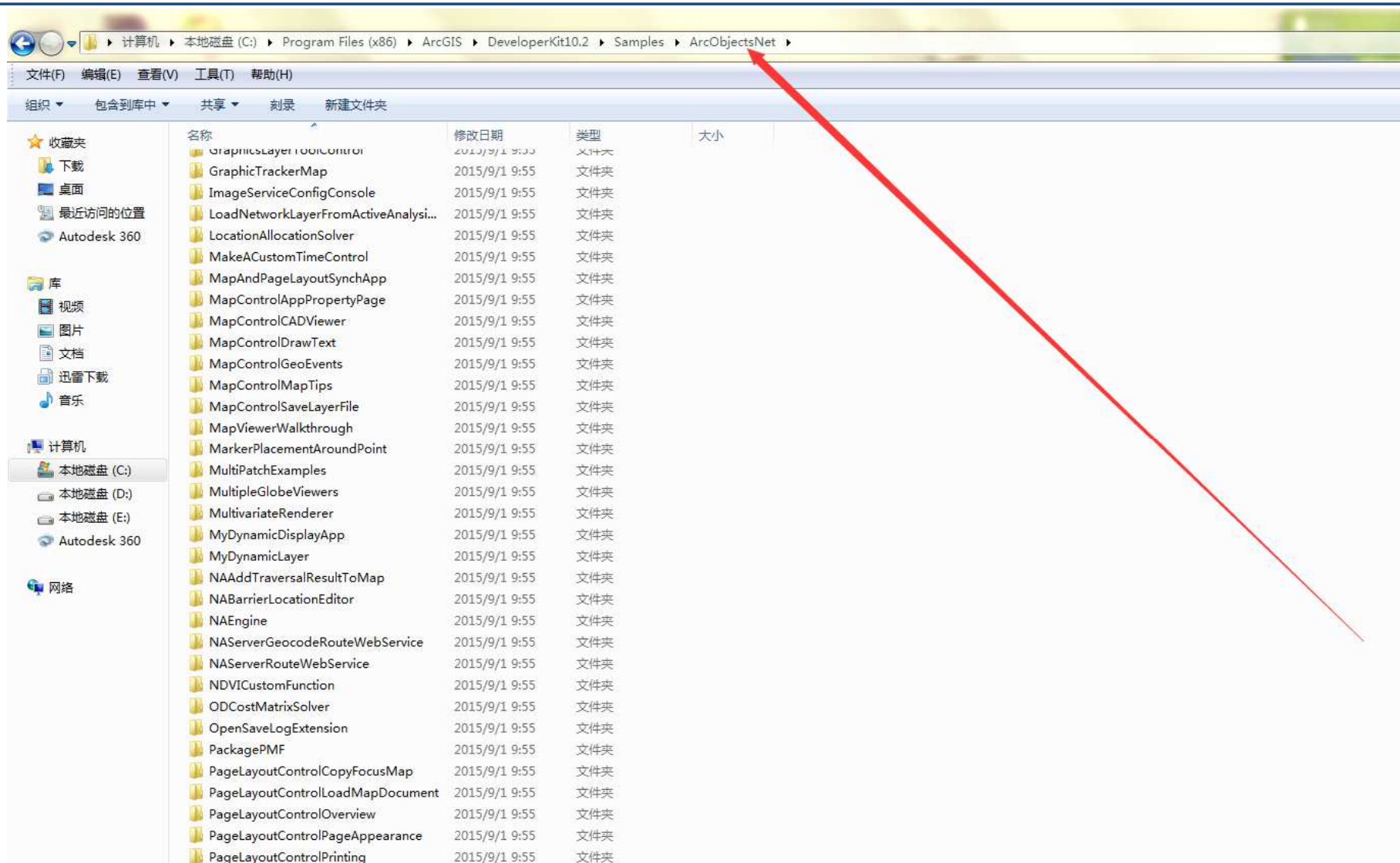
If you have a set of geoprocessing tools to run, or a series of maps to produce, consider Python scripting. Since .NET ArcObjects programming is not required, this SDK does not cover this type of customization. For more information, see [A quick tour of ArcPy](#) and [Geoprocessing scripts for map document management and output](#).

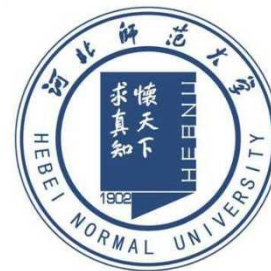
If you have a user that wants a focused application that addresses a particular task or workflow, you can create a stand-alone application. The application can do the following:

- Use the ArcGIS Engine controls to provide an interactive experience, either with a look and feel similar to that available in the ArcGIS for Desktop applications, or one with a completely different look

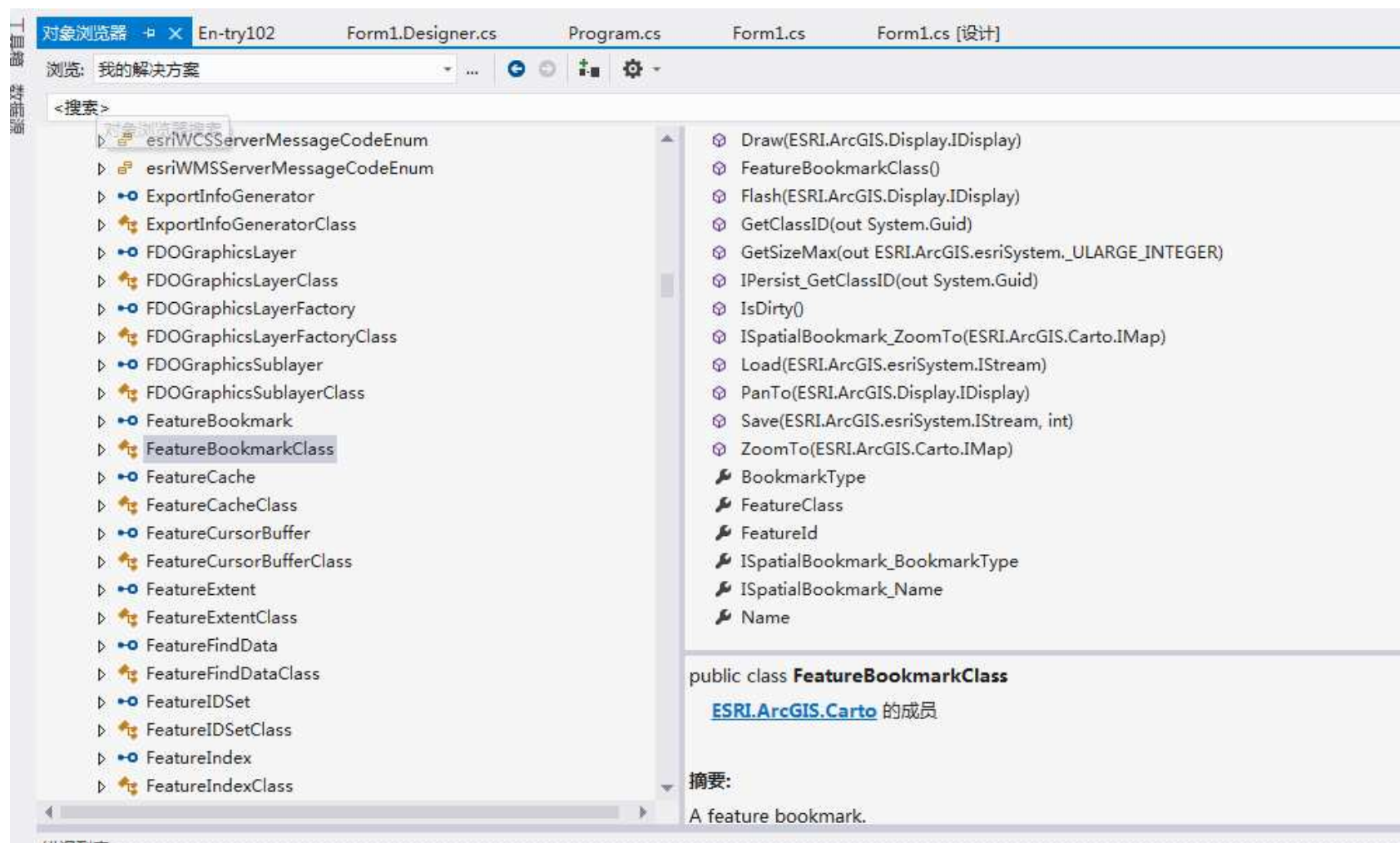


实例代码



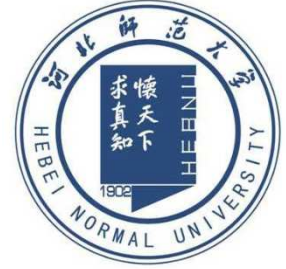


VS 中的对象浏览器

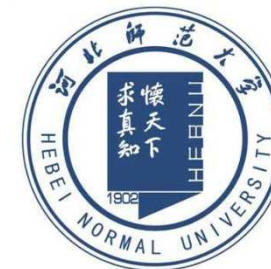


OMD (Object model diagrams)

对象模型图

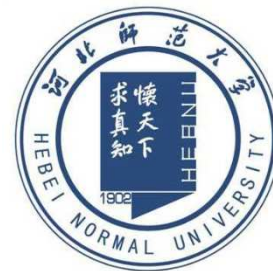


- OMD使用UML(Unified Modeling Language, 统一建模语言)描述, 它补充了在对象浏览器中看不到的信息, 可以帮助程序员了解类之间的关系, 如何从一个类到另一个类, 选择正确的接口, 获取需要的属性和方法等。



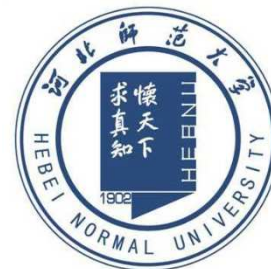
OMD 的物理存储地址

名称	修改日期	类型	大小
3DAnalystObjectModel	2012/2/22 12:31	Adobe Acrobat ...	100 KB
3DAnalystUIObjectModel	2012/2/22 12:34	Adobe Acrobat ...	31 KB
AddInsObjectModel	2012/2/27 12:30	Adobe Acrobat ...	1,045 KB
AnimationObjectModel	2012/2/23 11:38	Adobe Acrobat ...	52 KB
AnimationUIObjectModel	2012/2/23 11:39	Adobe Acrobat ...	29 KB
ArcCatalogObjectModel	2012/2/22 13:01	Adobe Acrobat ...	20 KB
ArcCatalogUIObjectModel	2012/2/22 13:11	Adobe Acrobat ...	13 KB
ArcGlobeObjectModel	2012/2/22 12:36	Adobe Acrobat ...	47 KB
ArcMapObjectModel	2012/2/23 11:41	Adobe Acrobat ...	23 KB
ArcMapUIObjectModel	2012/2/23 11:42	Adobe Acrobat ...	114 KB
ArcScanObjectModel	2012/2/22 13:27	Adobe Acrobat ...	48 KB
ArcSceneObjectModel	2012/2/22 12:38	Adobe Acrobat ...	29 KB
CadastralObjectModel	2012/2/22 12:51	Adobe Acrobat ...	29 KB
CadastralUIObjectModel	2012/2/22 12:56	Adobe Acrobat ...	120 KB
CartoObjectModel	2012/2/23 11:52	Adobe Acrobat ...	1,267 KB
CartoUIObjectModel	2012/2/23 11:55	Adobe Acrobat ...	97 KB
CatalogObjectModel	2012/2/22 13:13	Adobe Acrobat ...	154 KB
CatalogUIObjectModel	2012/2/22 13:14	Adobe Acrobat ...	42 KB
ControlsObjectModel	2012/2/22 13:25	Adobe Acrobat ...	177 KB
DataSourcesFileObjectModel	2012/2/22 16:44	Adobe Acrobat ...	84 KB
DataSourcesGDBObjectModel	2012/2/22 16:46	Adobe Acrobat ...	29 KB
DataSourcesNetCDFObjectModel	2012/2/22 16:47	Adobe Acrobat ...	21 KB
DataSourcesOLEDBObjectModel	2012/2/22 16:48	Adobe Acrobat ...	31 KB
DataSourcesRasterObjectModel	2012/3/19 14:10	Adobe Acrobat ...	420 KB
DataSourcesRasterUIObjectModel	2012/2/23 12:41	Adobe Acrobat ...	14 KB
DefenseSolutionsObjectModel	2011/9/20 8:51	Adobe Acrobat ...	232 KB
DisplayObjectModel	2012/2/23 11:29	Adobe Acrobat ...	353 KB
DisplayUIObjectModel	2012/2/23 11:31	Adobe Acrobat ...	32 KB
DotNetConnectionObjectModel	2008/5/5 11:34	Adobe Acrobat ...	82 KB
DotNetGeoprocessorObjectModel	2008/1/23 14:04	Adobe Acrobat ...	67 KB
EditorExtObjectModel	2012/2/22 13:28	Adobe Acrobat ...	75 KB
EditorObjectModel	2012/2/22 13:31	Adobe Acrobat ...	72 KB
EngineCoreObjectModel	2012/2/22 16:38	Adobe Acrobat ...	33 KB
FrameworkObjectModel	2012/2/27 12:34	Adobe Acrobat ...	71 KB



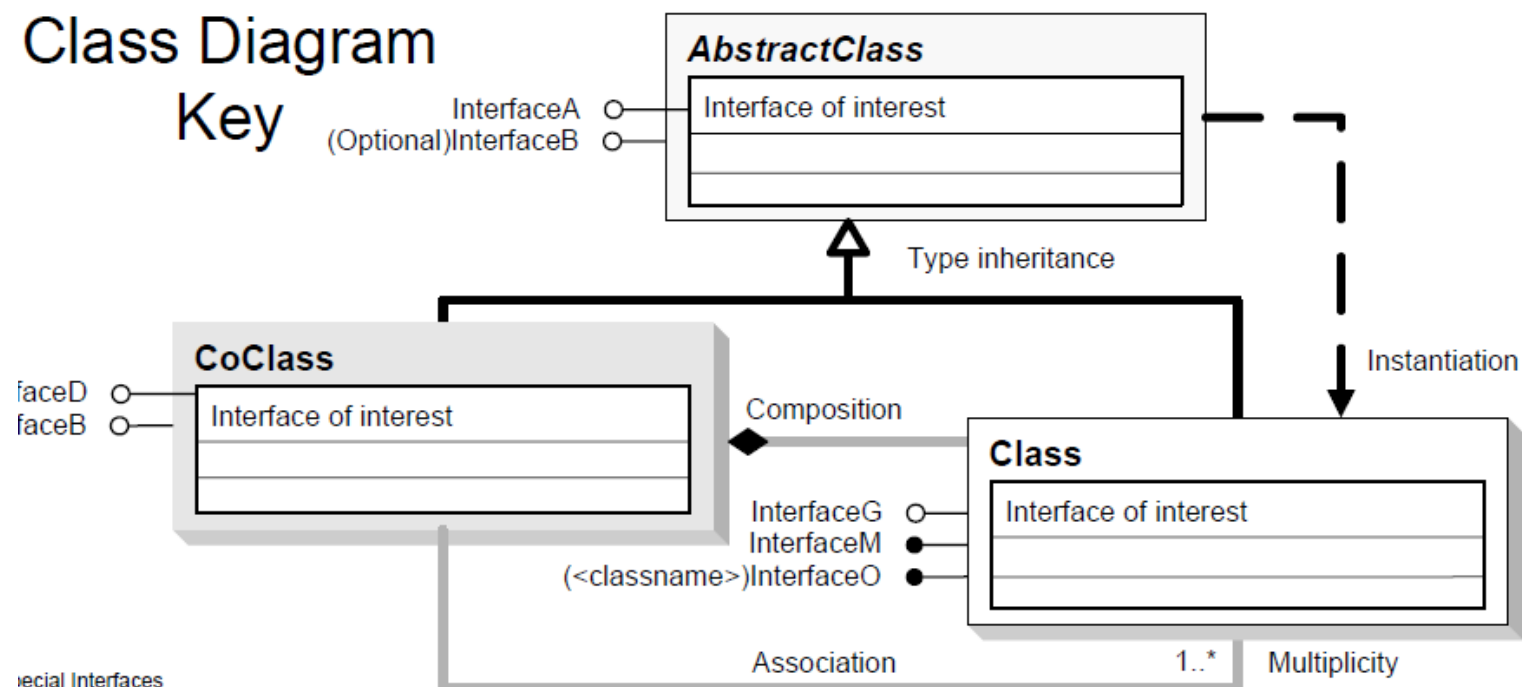
OMD功能明细

1. 该类支持哪些接口；
2. 完成任务需要哪些对象；
3. 如何使用该类的对象；
4. 是否可以直接实例化类；
5. 接口有哪些方法和属性；
6. 是否有其它类也支持该接口；
7. 对象间的关系。



对象模型图的钥匙

Class Diagram



Special Interfaces

(Optional) represents interfaces that are inherited by some subclasses but not all. The subclasses list the optional interfaces they implement.

(Instance) represents interfaces that are only on specific instances of the class.

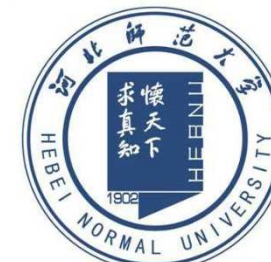
(<classname>) indicates the name of the class required to support this interface in Visual Basic.

- Inbound Interface
- Outbound Interface

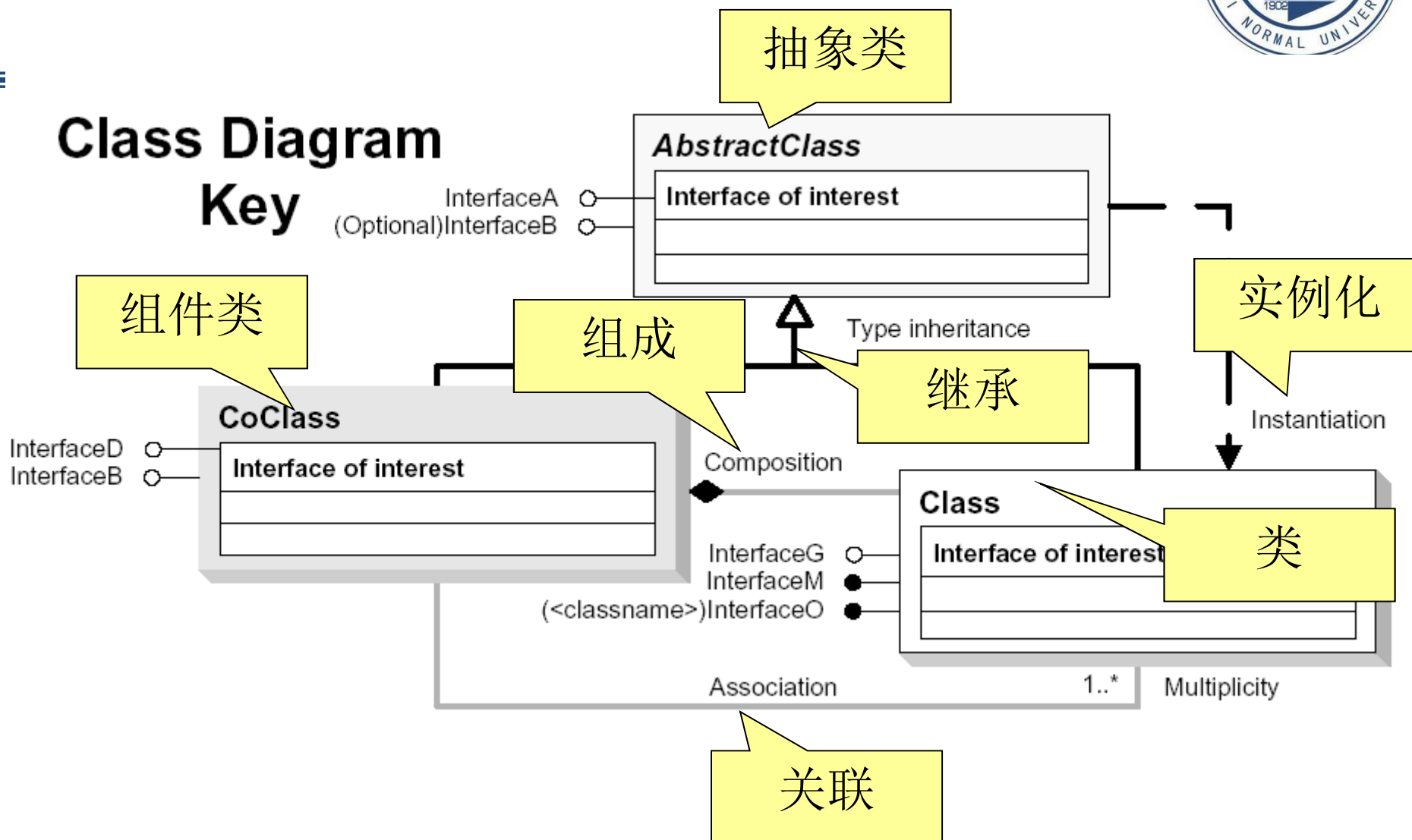
Interface key	
■	Property Get
■	Property Put
■	Property Get/Put
□	Property Put by Reference
←	Method

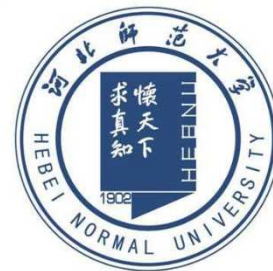
enumeration
firstValue - firstEnumeration
secondValue - secondEnumeration

Structure key		<<Struct>>
firstMember:	Type	
secondMember:	Type	



Class Diagram Key



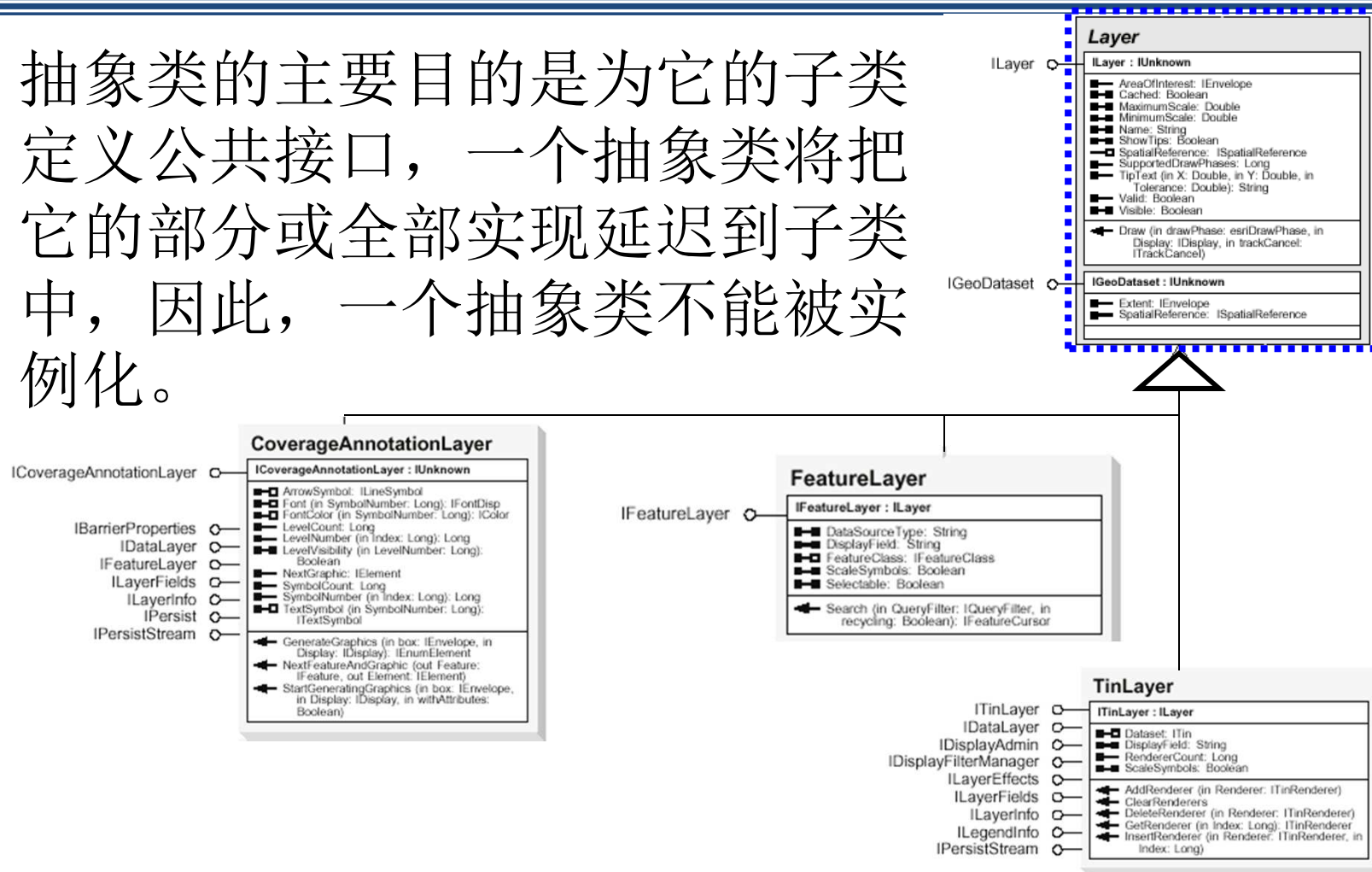


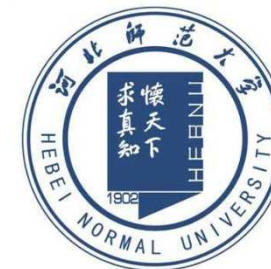
ArcObjects中的类与对象

- 在ArcObjects中存在三种类型的类：
 - 抽象类(Abstract Class): 不能创建或实例化, 但是可以用于定义一个子类。如 Layer。OMD符号为: 二维的内部有阴影的矩形。
 - 组件类(CoClass): 可以直接用new创建对象实例的类, 实例对象不依赖其他对象的存在而存在, 生存周期也不用其他对象管理。如MapDocument。OMD符号为: 带阴影的3D矩形符号。
 - 普通类(Class): 不能创建, 但它可以用其他普通类或组件类的方法产生。如FeatureClass。OMD符号为: 3D矩形内部没有阴影。

抽象类

- 抽象类的主要目的是为它的子类定义公共接口，一个抽象类将把它的部分或全部实现延迟到子类中，因此，一个抽象类不能被实例化。





组件类

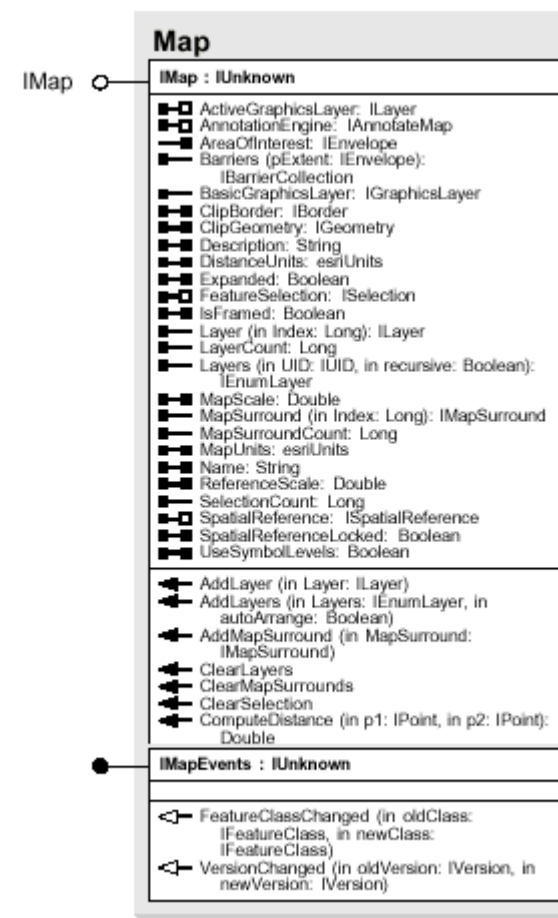
Creatable: Use the *New* keyword

```
IMap map = new MapClass();
```

Instantiable: Obtain from other objects

```
IMap map;
```

```
map= pMxDocument.FocusMap;
```

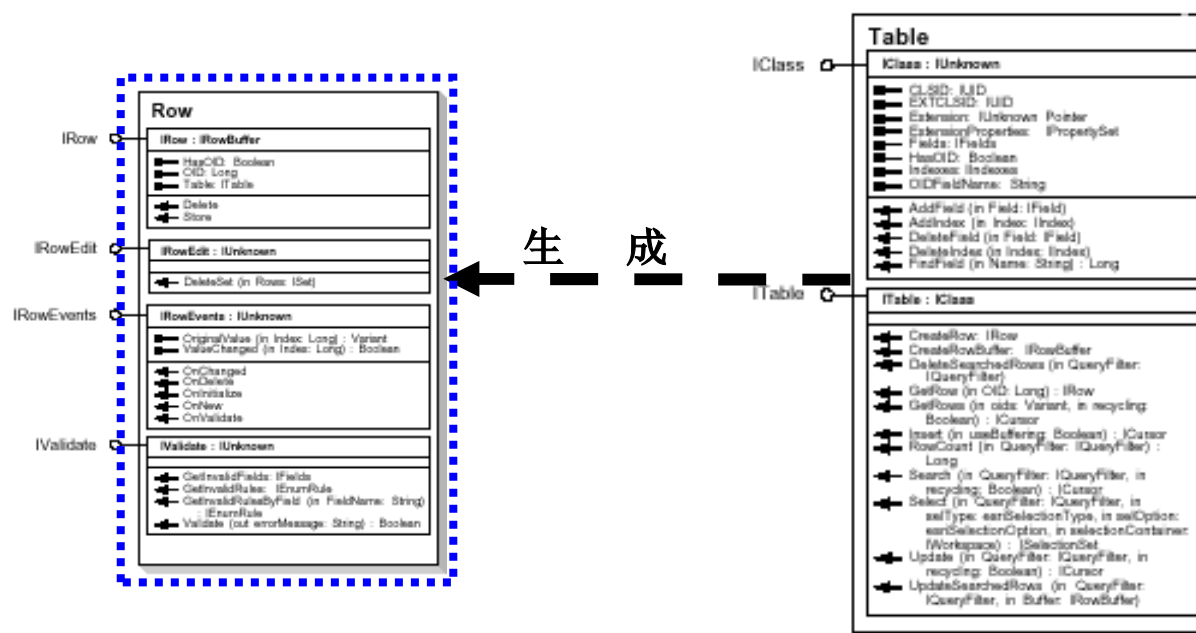


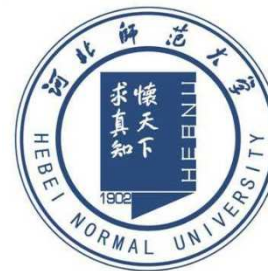
普通类

- Noncreatable class
 - Cannot create with the *New* keyword
- Obtain instances from other objects

IRow row

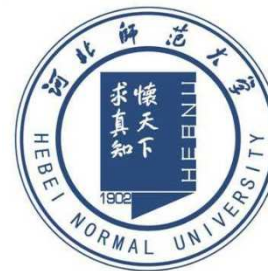
row = table.CreateRow();





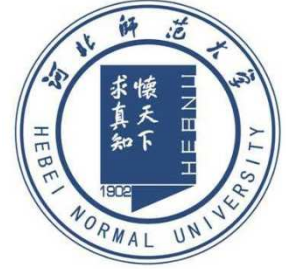
体会

- 个人感觉在OMD图中比较难理解的就是可实例化类Class。可用以下例子去理解它：
- ArcEngine中 workspace 就是一个可实例化类，Workspace不能用new创建，但可以通过WorkspaceFactory.OpenFromFile()方法来创建。如下：
IWorkspaceFactory pWorkspaceFactory = new AccessWorkspaceFactory();
IWorkspace pWorkspace =
pWorkspaceFactory.OpenFromFile(ConnectionString, 0);
- 但当我们尝试用new去创建一个Workspace类时，如下：
IWorkspace pWorkspace = new Workspace();
就会出现以下错误：
“类型” ESRI.ArcGIS.Geodatabase.Workspace未定义构造函数；
- 所以，Class和CoClass的主要区别就是CoClass是带构造函数的Class，因此它可以用new直接创建对象。



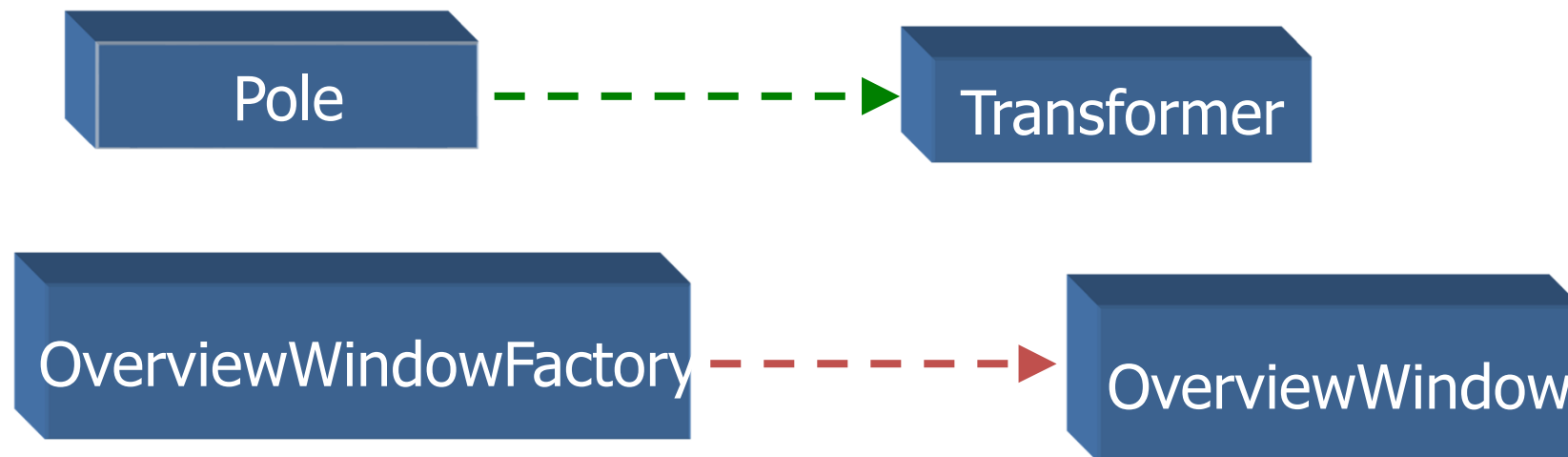
类与类的关系

- ArcObjects的类之间存在四种关系
 - 依赖关系
 - 关联关系
 - 组成关系
 - 类型继承



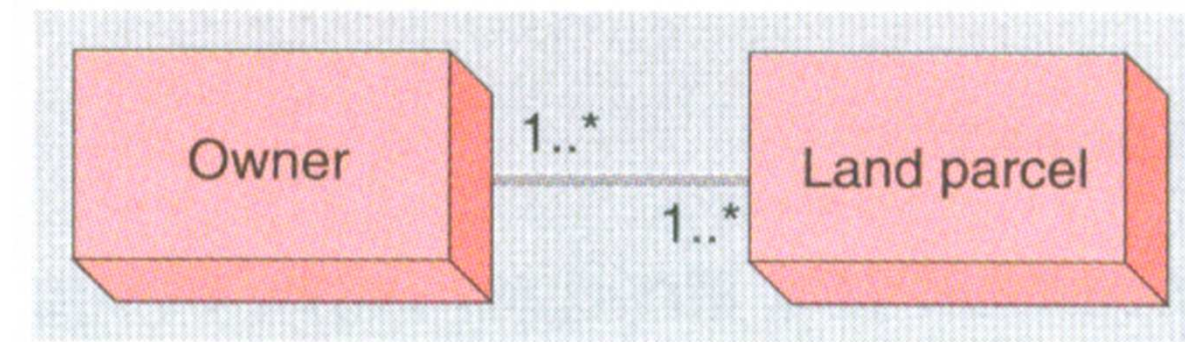
Instantiation实例化（依赖）

Instantiation specifies that one object from one class has a method with which it creates an object from another class.
A pole object might have a method to create a transformer object.



1. Associations (关联)

- Associations represent relationships between classes. They have defined multiplicities at both end.





2. Multiplicity (多重性)

A Multiplicity is a constraint on the number of objects that can be associated with another object. This is the notation for multiplicities:

1 – One and only one.

0..1 – Zero or one

M..N – From M to N (positive integers)

* Or 0..* – From zero to any positive integer

1..* – From one to any positive integer



Composition (组成)

Composition is a stronger form of aggregation in which objects from the “whole” class control the lifetime of objects from the “part” class.



A pole contains one or many crossarms. In This design, a crossarm cannot be recycled before the pole is removed. The pole object controls the lifetime of the crossarm object.



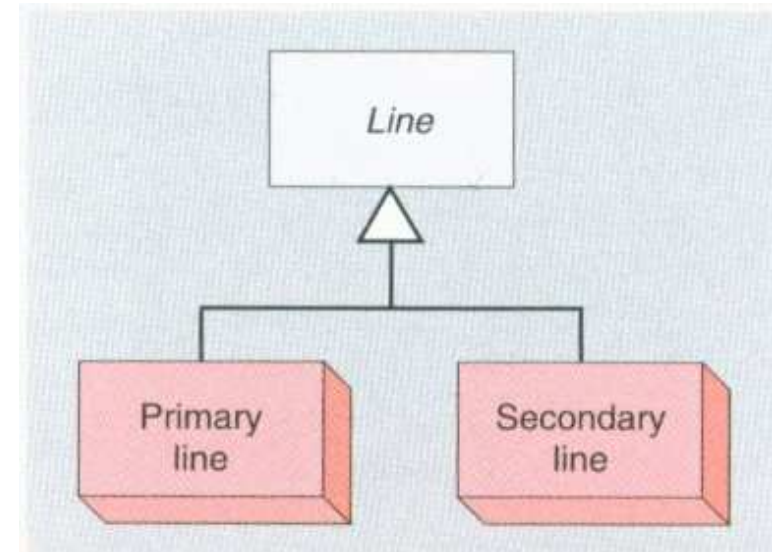
Composition (Cont.)

A MxDocument object contains none or many TOCView object:



Inheritance (继承)

Type inheritance defines specialized classes that share properties and methods with the superclass and have additional properties and methods.



This diagram shows that a primary line (creatable class) and secondary line (creatable class) are types of a line (abstract class).

Relationship symbols

- 继承



- 组成



- 依赖

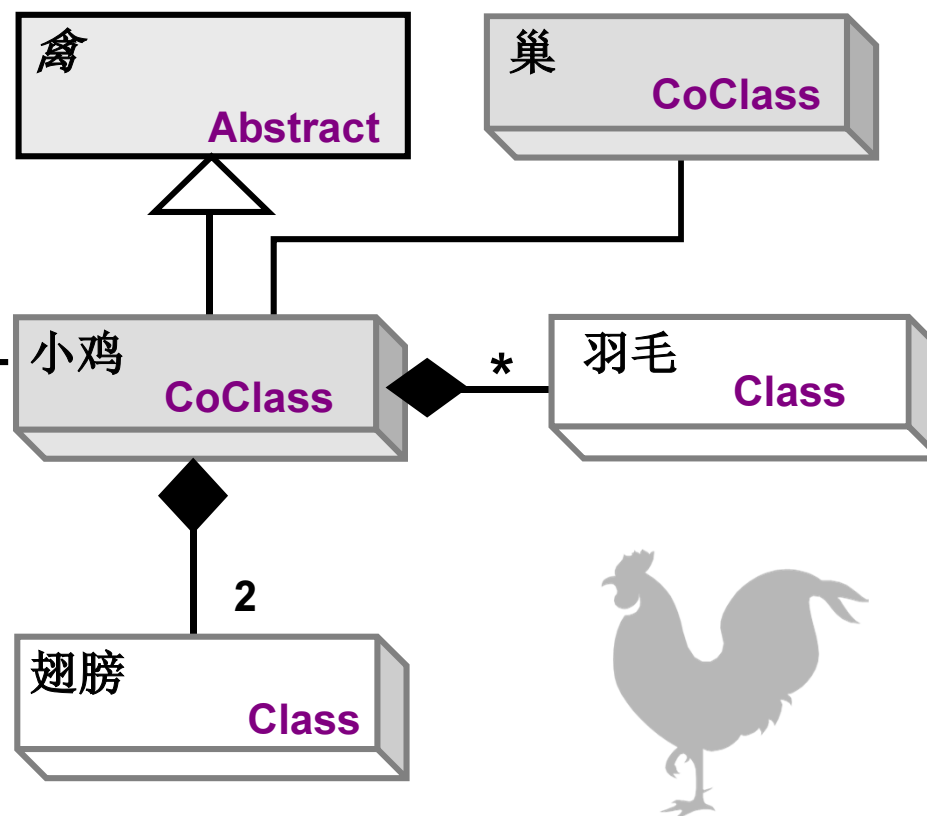


- 关联



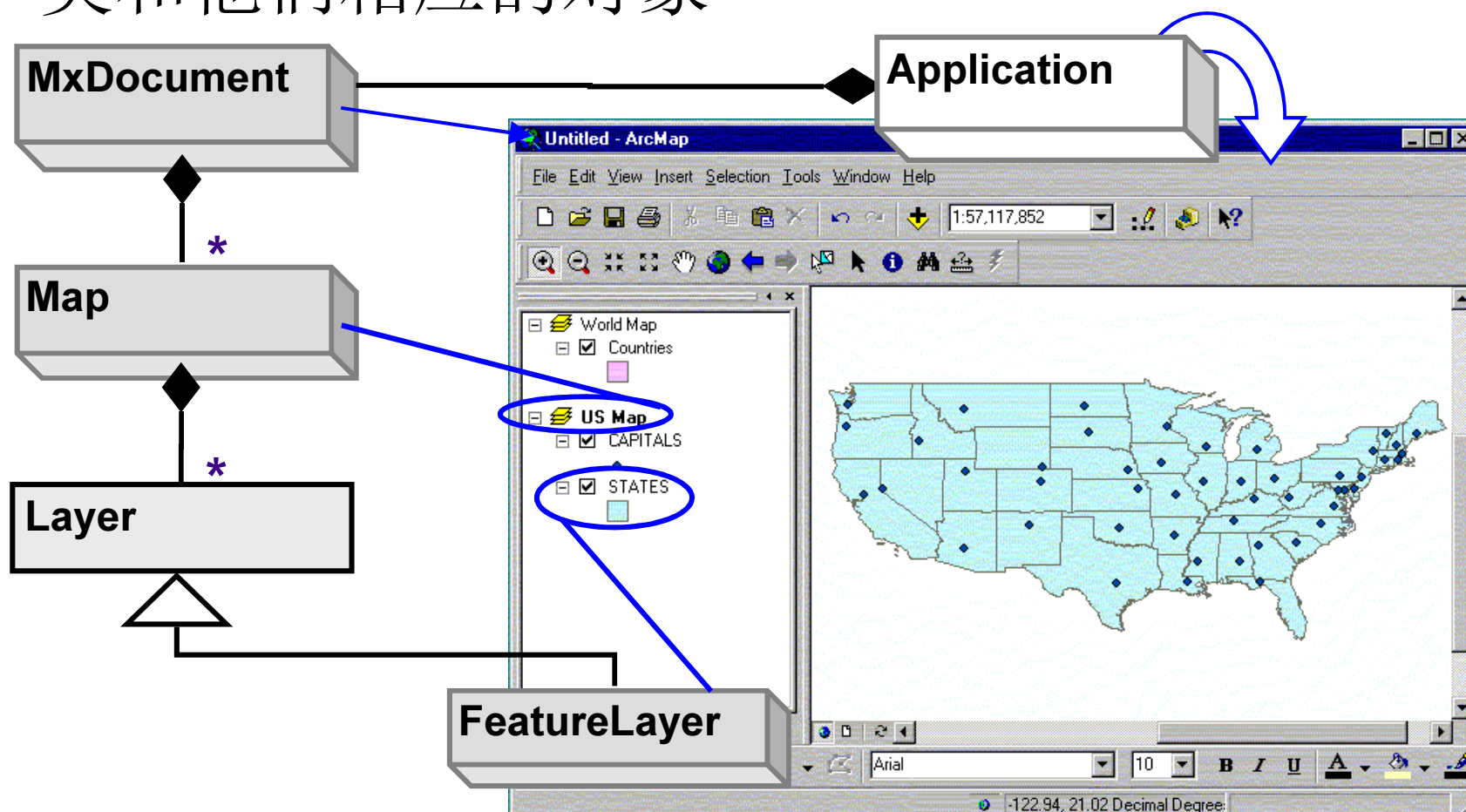
- 对应关系 1:N

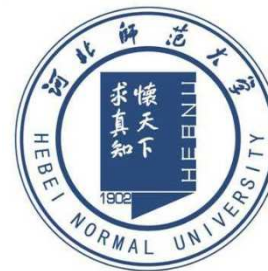
*



ArcMap objects

- 类和他们的相应的对象

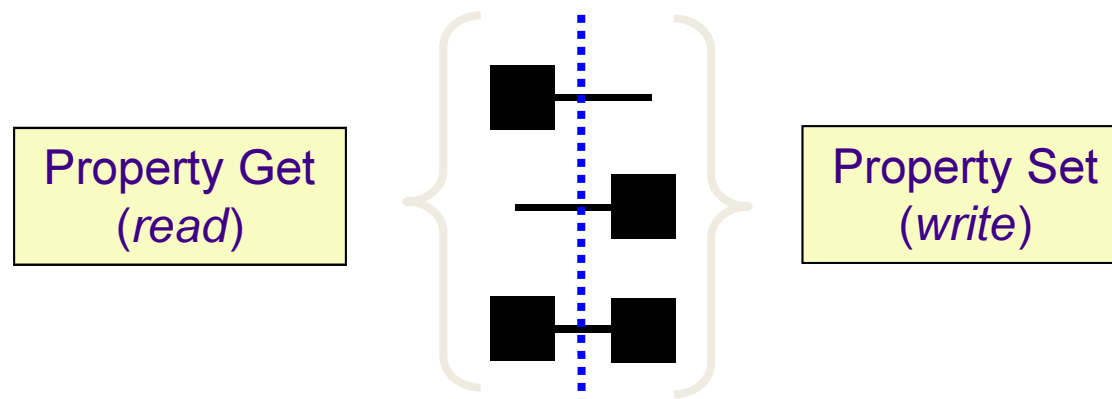




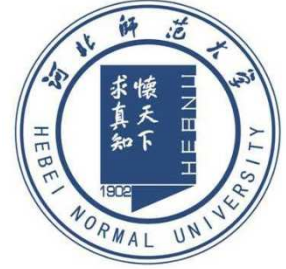
Property and method symbols

- Property 

— 哑铃形状的图标



- Method 



Setting properties

- Property Put: Most ArcObjects properties

- Property holds a value or a *copy* of an object

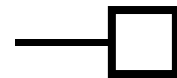
- Layer.Name = "Port Moresby";*



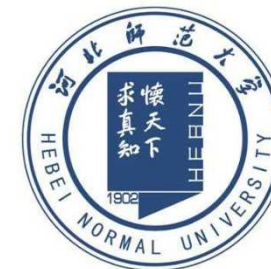
- Property Put by Reference: Some ArcObjects properties

- Property holds a *reference* to an object

- pLayer.FeatureClass = pMoresbyData;*



- 如果引用对象发生了变化，对象的属性将同步受到影响



Getting properties

- Return a value
 - DocumentFilename: String
- Return an object reference
 - ActiveView: IActiveView
 - Map: IMap

IMapDocument
DocumentVersion
IDocumentInfo
IDocumentInfo2
DocumentPreview

MapDocument

IMapDocument : IUnknown

- ActiveView: IActiveView
- DocumentFilename: String
- DocumentType: esriMapDocumentType
- DocumentVersion: esriMapDocumentVersionInfo
- IsMapDocument (in sDocument: String): Boolean
- IsPasswordProtected (in sDocument: String): Boolean
- IsPresent (in sDocument: String): Boolean
- IsReadOnly (in sDocument: String): Boolean
- IsRestricted (in sDocument: String): Boolean
- Layer (in mapIndex: Long, in layerIndex: Long): ILayer
- Map (in mapIndex: Long): IMap
- MapCount: Long

```
pMap = pMapDocument.get_Map(i);           //Return an object reference
Console.WriteLine(pMap.Name);              //Return a value
IEnumLayer pELayer = pMap.get_Layers(null, true); //Return an object reference
pELayer.Reset();
ILayer pLayer = pELayer.Next();             //Return an object reference
while (pLayer != null)
{
    Console.WriteLine(pLayer.Name);         //Return a value
    pLayer = pELayer.Next();
}
```

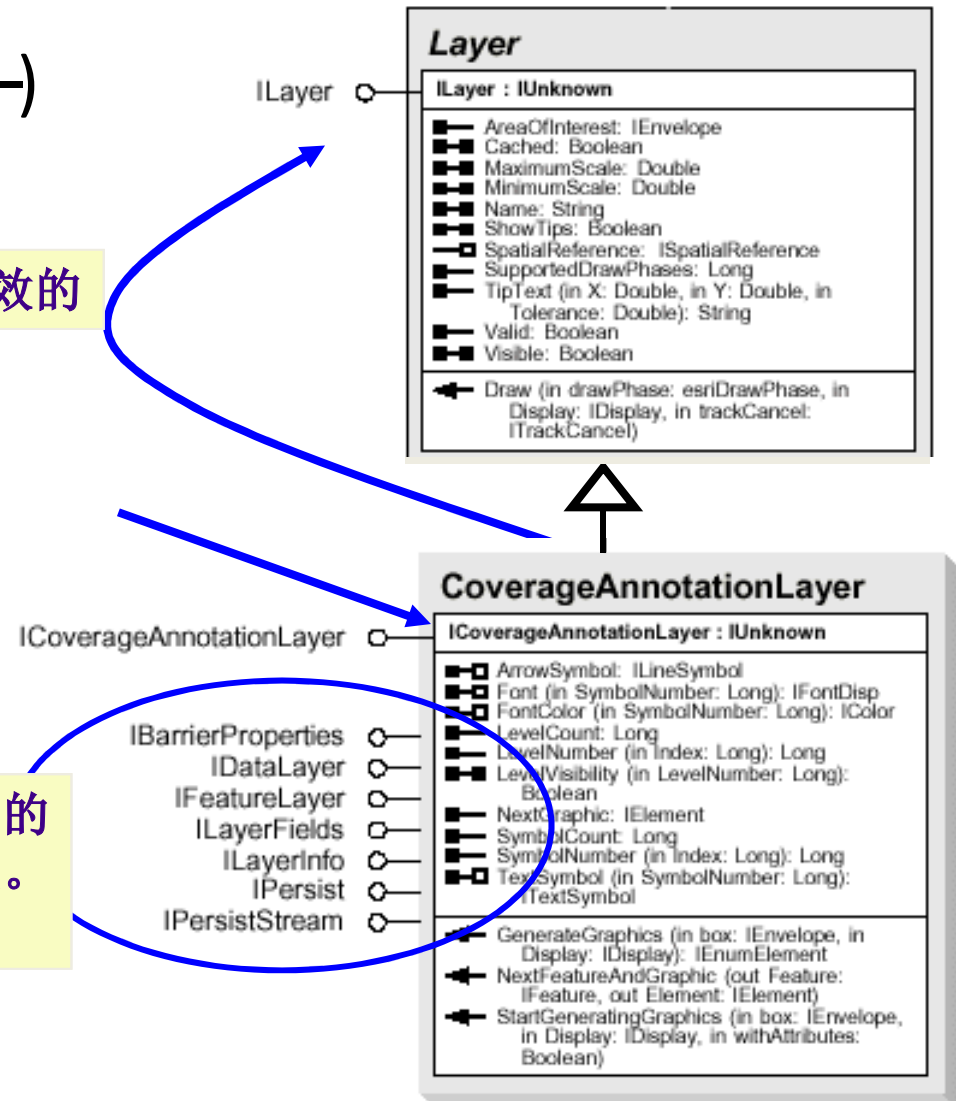
Finding interfaces

- 棒棒糖类型的图标 (○—)

继承接口是有效的

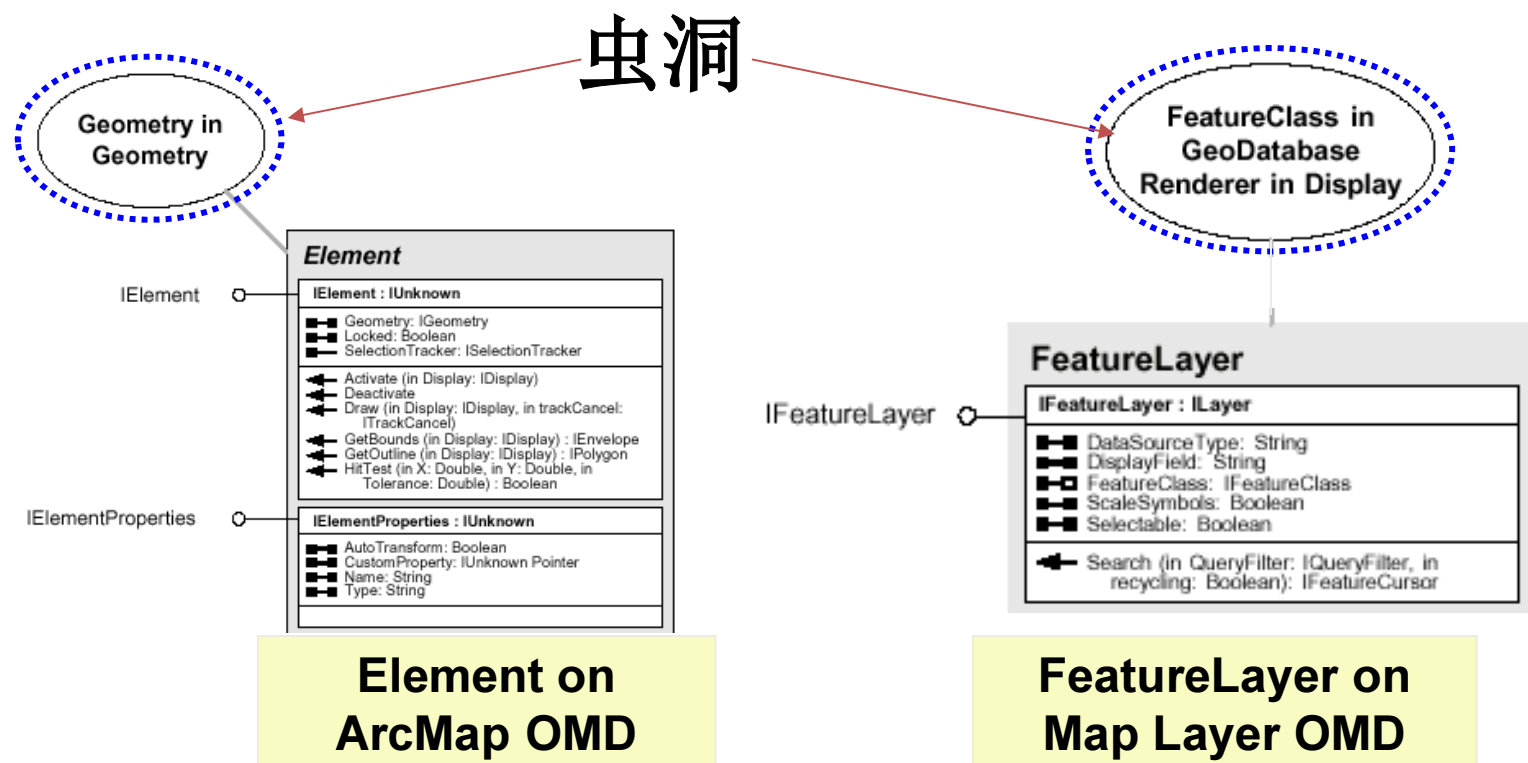
该接口被本类使用。
所有的属性和方法列表在类图中。

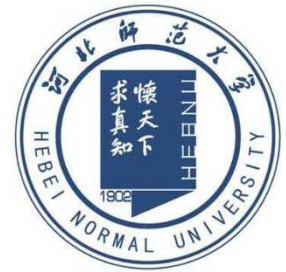
这些接口也是有效的。但是详细的
属性和方法必须在其它地方浏览。
(e.g., Object Browser).



Wormholes

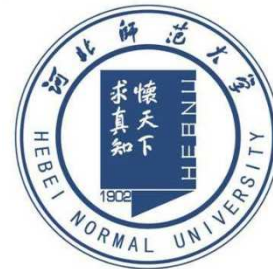
- 概念上，这是一个对象模型
- 事实上，被分别在几张图表里存放
 - 虫洞表现了图表与图表间的连接关系





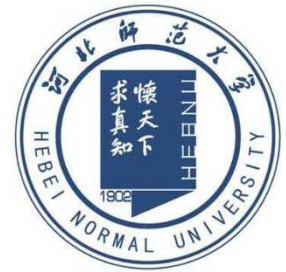
练习题

- ArcObjects 10.2有多少程序集，其中有多少程序集是AE可以使用的？
- 列举五个ArcGIS Engine中的常用程序集。
- AO中的类有那几种，分别具有什么特性？
- AO中的类存在哪几种关系，各有什么分别有什么特征？

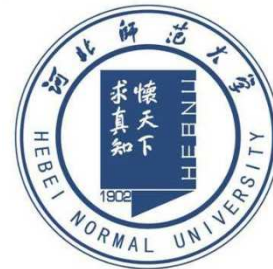


软件开发历史

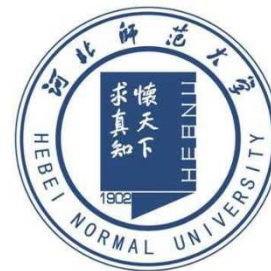
- **软件重用**和**开发效率**始终是软件开发的核心问题。程序员希望能够写一次代码，在任何地方都可以运行。
 - 在早期（面向过程语言），人们为了重用，将一些基本的数学计算函数和界面设计函数设计成函数库，让其他软件开发人员调用。它为应用开发者提供了方便，但是重用的粒度太小。



- 面向对象语言的出现，提高编程的粒度。面向对象语言的基本单位是类，它封装了数据成员和函数成员，将最小组件的粒度提高了一个等级。程序员需要直接调用的不是过程和函数了，而是类。但是，这并没有解决编程中所有问题，并且产生了新的问题。一个类提供了很多种方法和属性，这些方法和属性其实是可以分组的，为不同的功能服务，但类并没有做这个管理，他只是一个属性和方法的容器。

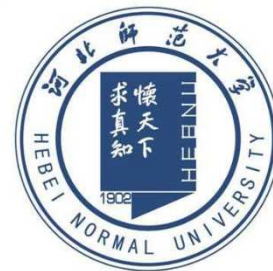


-
- **COM**(Component Object Model)技术（**接口**）的出现，解决了以上问题，将类的内部属性和方法进行分组。例如在**Map**类中有好几个接口，在这些接口中定义不同功能组的方法和属性，**Map**实现了这些接口，这样就可以使用接口进行定义。



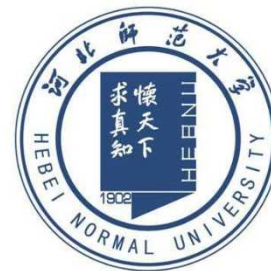
类与接口

- 类：属性和方法（已经实现的）
- 接口：定义不同功能组的方法和属性，在这里方法不被实现，而把这些方法的实现交给继承它的类.



接口的定义

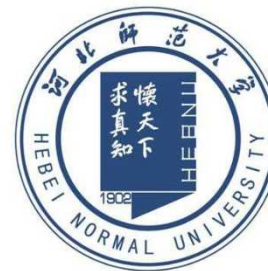
- 接口的定义是指定一组函数成员而不实现成员的引用类型，其它类型和接口可以继承接口。定义还是很好理解的，但是没有反映特点，接口主要有以下特点：
 - (1)通过接口可以实现多重继承，C#接口的成员不能有public、protected、internal、private等修饰符。原因很简单，接口里面的方法都需要由外面接口实现去实现方法体，那么其修饰符必然是public。C#接口中的成员默认是public的，java中是可以加public的。
 - (2)接口成员不能有new、static、abstract、override、virtual修饰符。有一点要注意，当一个接口实现一个接口，这2个接口中有相同的方法时，可用new关键字隐藏父接口中的方法。
 - (3)接口中只包含成员的签名，接口没有构造函数，所有不能直接使用new对接口进行实例化。接口中只能包含方法、属性、事件和索引的组合。接口一旦被实现，实现类必须实现接口中的所有成员，除非实现类本身是抽象类。
 - (4)C#是单继承，接口是解决C#里面类可以同时继承多个基类的问题。



程序演示

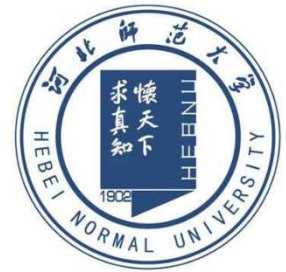
- 定义一个接口 `IPeople`，该接口有一个性别的方法，但是方法内没有内容。

```
interface IPeople
{
    void gender();
}
```



定义Boy类

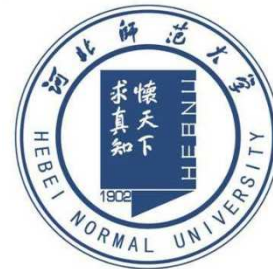
```
class Boy : IPeople
{
    public void gender()
    {
        Console.WriteLine("I'm a boy.");
    }
}
```

定义Gril类

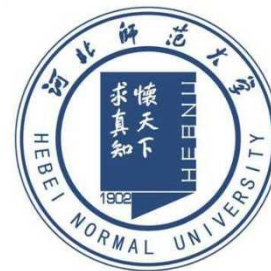
```
class Gril : IPeople
{
    public void gender()
    {
        Console.WriteLine("I'm a gril.");
    }
}
```

- 以上两个类明确了gender方法的作用



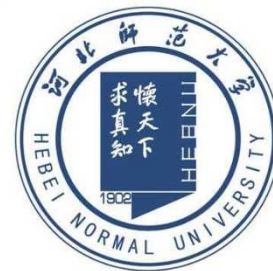
主函数中调用

```
IPeople Person;  
// 声明接口变量  
Person = new Boy();  
// 实例化，接口变量中存放对象的引用  
Person.gender();  
// 这个调用的是Boy中的gender方法  
Person = new Girl();  
// 实例化，接口变量中存放对象的引用  
Person.gender();  
// 这个调用的是Gril中的gender方法  
Console.ReadLine();
```



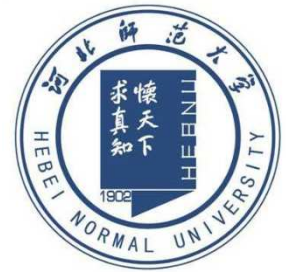
接口VS抽象类

- 接口用于规范，抽象类用于共性。
- 抽象类是类，所以只能被单继承，但是接口却可以一次实现多个。
- 接口中只能声明方法，属性，事件，索引器。而抽象类中可以有方法的实现，也可以定义非静态的类变量。
- 抽象类可以提供某些方法的部分实现，接口不可以。抽象类的实例是它的子类给出的。接口的实例是实现接口的类给出的。
- 在抽象类中加入一个方法，那么它的子类就同时有了这个方法。而在接口中加入新的方法，那么实现它的类就要重新编写（这就是为什么说接口是一个类的规范了）。
- 接口成员被定义为公共的，但抽象类的成员也可以是私有的、受保护的、内部的或受保护的内部成员（其中受保护的内部成员只能在应用程序的代码或派生类中访问）。此外接口不能包含字段、构造函数、析构函数、静态成员或常量。
- 还有一点，我们在VS中实现接口时会发现有2个选项，一个是实现接口，一个是显示实现接口。实现接口就是我们平常理解的实现接口，而显示实现接口的话，实现的方法是属于接口的，而不是属于实现类的。



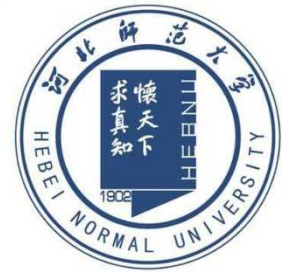
COM出现的原因

- 随着计算机的发展和应用范围不断延伸，计算机软件系统的规模也在不断扩大，结构也越来越复杂，代码越来越长，维护越来越困难，从几百行代码到几万甚至几十万、几百万行代码的软件系统随处可见。
- 软件系统管理、设计及其重要
 - 功能完善、结构优良、开发效率高、稳定性和安全性强、扩展方便、维护简单、易于复用、生命周期长、投资成本低的软件系统。



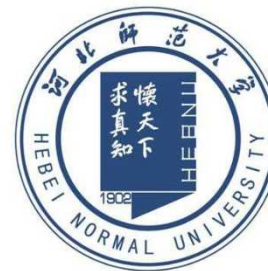
COM模型

- 基本思想是将大而复杂的软件应用分成一系列可先行实现、易于开发、理解和调用的软件单元组件。
- COM模型：是关于如何建立组件，以及如何通过组件建立应用程序的一个规范，说明了如何动态交替更新组件。
- COM技术标准：
 - COM（DCOM）：微软的组件对象模型((Distributed Component Object Model)
 - JavaBean：Sun公司的基于Java的组件技术
 - CORBA：对象管理组织(Object Management Group, OMG)的组件技术标准，公共对象请求代理结构(Common Object Request Broker Architecture, CORBA)。



COM技术知识要点

- COM一种二进制级别的组件通信标准--接口 (Interface)，它定义了组件之间应该如何通信、一个COM对象之间的不同接口如何查询等。
- COM对象基于面向对象标准，可以是DLL或EXE，包含接口的具体实现。使用者可以通过接口来访问其内部的方法、属性。
- COM对象必须实现IUnknown接口，它负责管理COM对象生命周期并在运行时刻提供类型查询，当COM对象不使用的時候，是这个接口定义的方法负责释放内存。



COM中的接口转换QI，即所谓查询接口

- 由于一个COM对象有很多个接口，不同的接口管理着COM的不同类型的方法、属性，因此从一个接口可以使用的属性方法转到另一个接口可以使用的属性方法的过程称为QI。这个过程也是由IUnknown接口管理的。

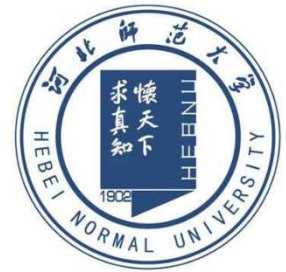
```
IMap pMap ; IActiveView pActiveView ;
```

```
pMap = axMapControl1.Map; //获取控件中的地图
```

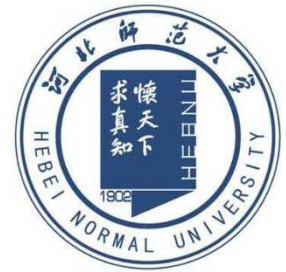
```
.....
```

```
pActiveView = pMap as IActiveView; //QI
```

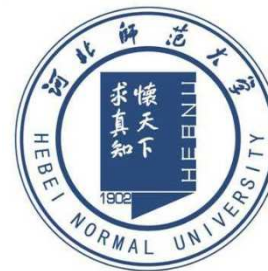
```
(Or pActiveView = (IActiveView)pMap ; )
```

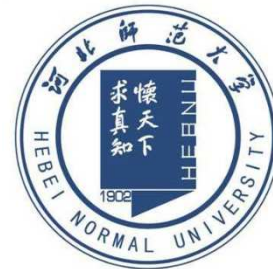
- 每个组件都有一个独一无二的标识GUID。接口的GUID称为IID，而组件类的GUID称为CLSID。CLSID还有一个文本别名，即所谓的ProgID，它是一个由项目名加组件类类名组成的字符串。
- 一个COM对象可以有多个接口，一个接口也可以被多个COM对象实现。
- 接口分为两种，即内向接口和外向接口。
 - 内向接口(Inbound Interface)是组织COM对象相关的**方法**和属性，COM对象必须实现所有的接口内容；
 - 外向接口(Outbound Interface)是用于组织COM对象相关的**事件**，开发人员根据需要实现事件接口。



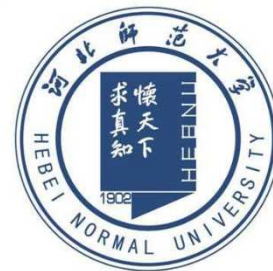
- 接口可以从多个父接口继承；
- COM对象的接口一旦被公布就不能再修改，只可以添加新的接口；
- COM组件必须注册。
- COM有很多优点，但它也有致命缺陷：由于COM对象可以被重用，这样多个程序可能共用一个COM对象，如果这个COM组件升级了，就很可能出现其中某个程序无法使用新组件，导致程序不能运行的情况，这种情况被称为“**DLL HELL**”(DLL灾难)。



- 针对Windows平台存在的缺陷，微软提出了全新的.NET技术体系，其主要包括
 1. .NET Framework
 2. Visual Studio.NET
- Visual Studio .NET 是一套完整的开发工具，用于创建各种类型的应用程序，包括 ASP Web 应用程序、XML Web services、桌面应用程序和移动应用程序

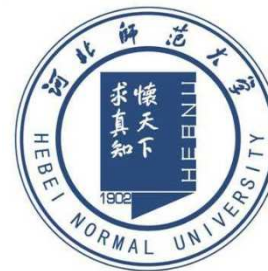


- .NET Framework 是生成、部署和运行 各种应用程序的平台。 .NET Framework 具有两个主要组成部分：
 1. 通用语言运行时（Common Language Runtime--CLR）,是 .NET Framework 的基础
 2. .NET Framework 类库。

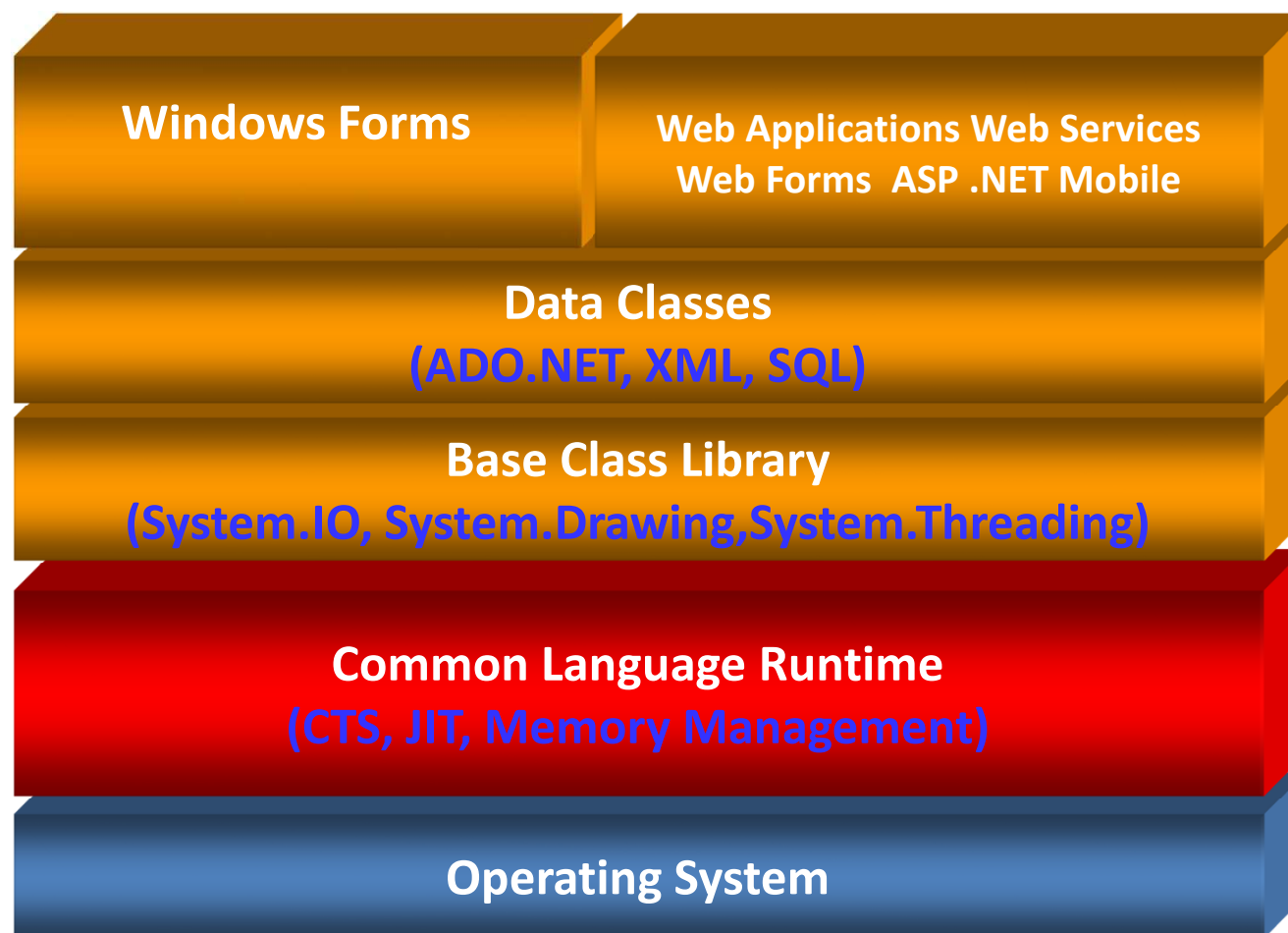


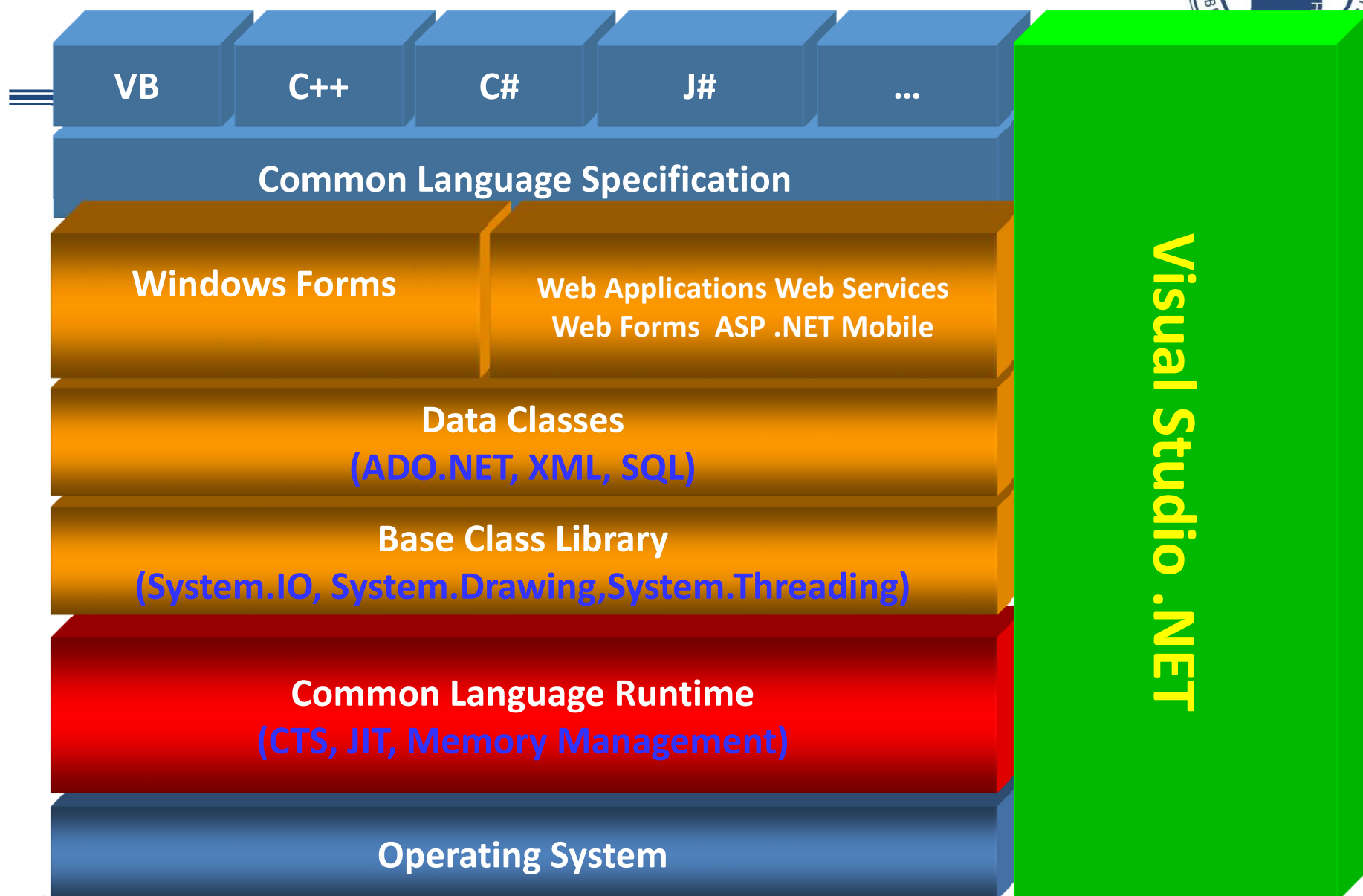
通用语言运行时（Common Language Runtime-CLR）

- 通用语言运行时管理内存、线程执行、代码执行、代码安全验证、编译以及其他系统服务。
- CLR处于.NET Framework的最底层，是.NET Framework的基础，它为C#.NET、VB.NET、C++、J#等多种语言提供了统一的运行环境，使得用某种高级语言编写的应用程序可以在不同的操作系统和硬件上运行，而不需要按照特定环境重写应用程序。



.NET Framework体系结构







The .NET Framework

Browser
Apps

Web Services
Apps

Local
Apps

Other
Apps

Windows
Forms

Enterprise
Services

ASP.NET

ADO.NET

More

.NET Framework Class Library

Common Language Runtime

Windows



The Java Environment

Browser
Apps

Web Services
Apps

Local
Apps

Other
Apps

Swing

Enterprise
JavaBeans

JavaServer
Pages

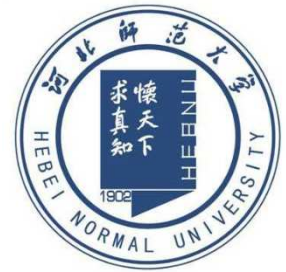
JDBC

More

Standard Java Packages

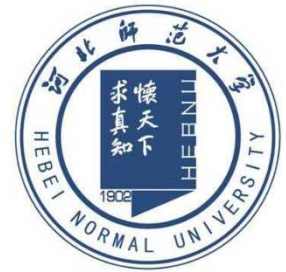
Java Virtual Machine (VM)

Windows, Solaris, Linux, others

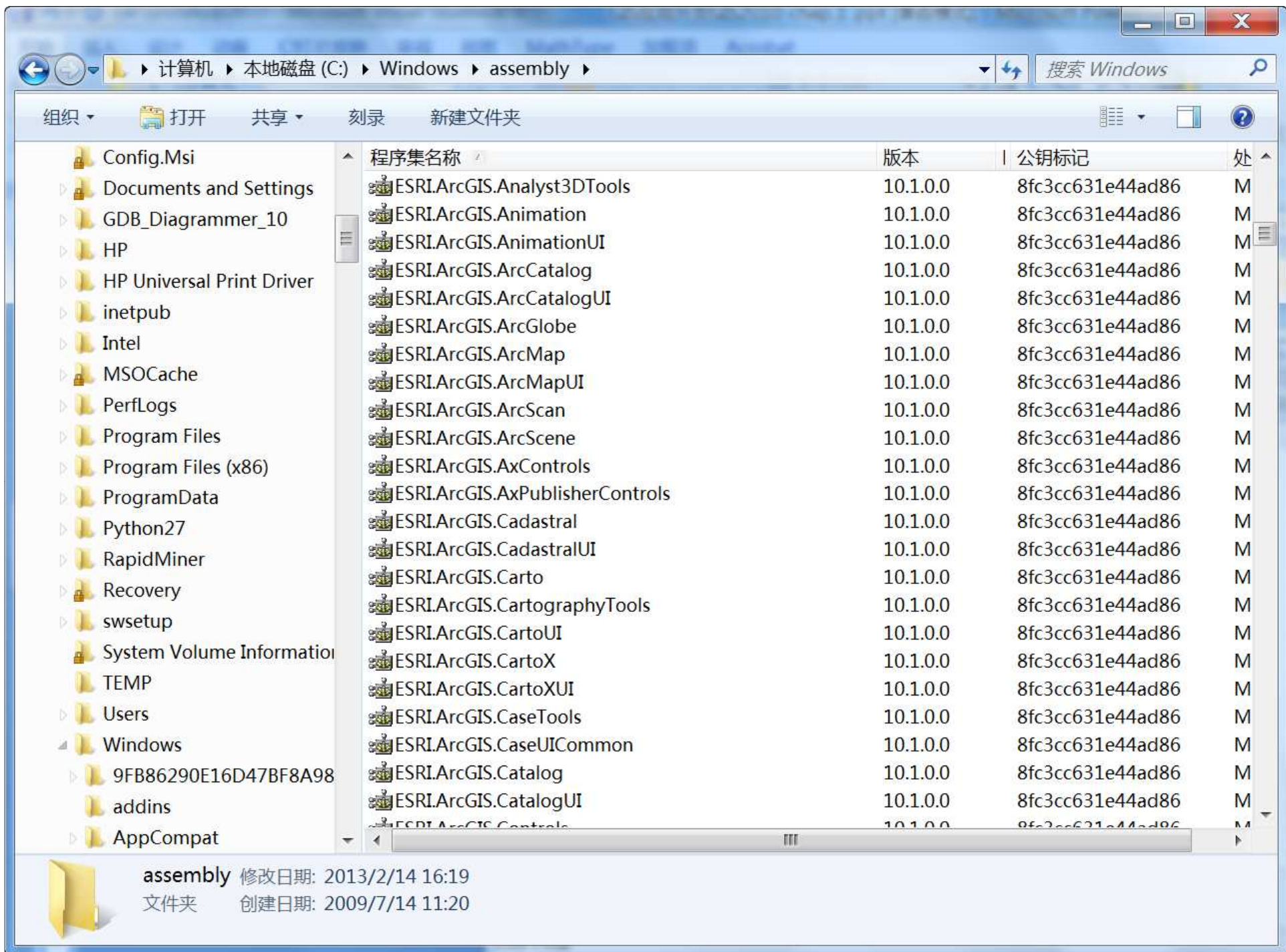


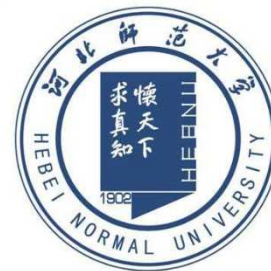
解决DLL Hell-- .NET 的强命名程序集

- 以前在Windows下开发程序时常会遭遇“DLL Hell”问题，即动态链接库的向后兼容问题。
- 微软在.Net产生前，使用Guid来唯一的标识每一个COM组件，为了运行COM组件就必须在注册表中对其进行注册。多个应用程序可以共享相同的COM组件，当COM组件版本升级时，可能导致原来引用此COM组件的程序不能正确运行，等等。
- 在.Net中，微软引入了一种新的解决方案：强命名(Strong Name)程序集，以及与之配套的全局程序集缓存（GAC --Global Assembly Cache）来解决这个问题。



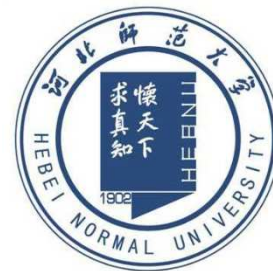
- 强命名程序集：.Net使用Name、Version、Culture、PublicKey四个属性来唯一标识一个程序集。
- 创建一个强命名程序集需要获得一个用强命名实用工具（Strong Name Utility，即SN.exe）产生的密钥。
- 全局程序集缓存（GAC）：如果一个Assembly要被多个应用程序访问，那么它就必须放在一个CLR已确知的目录下，并且CLR在探测到有对该Assembly的引用时，它必须能自动到该目录下寻找这个程序集。这个已确知的目录称作GAC，就是全局程序集缓存。它一般位于下面的目录下：C:\Windows\Assembly\
- 使用gacutil.exe命令可以将具有强命名的程序集添加到全局程序集缓存。





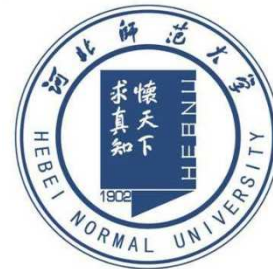
什么时候使用

- 在开发一般的、非共享的程序时，不需要使用强命名的程序集，仅将项目（**Project**）编译成**.DLL**或者**.EXE**即可。但是，如果开发的是组件库、框架时，通过对程序集进行强命名，并将其部署到**GAC**中，可以保证程序集不会出现版本问题。



.NET Framework 4.0新特性

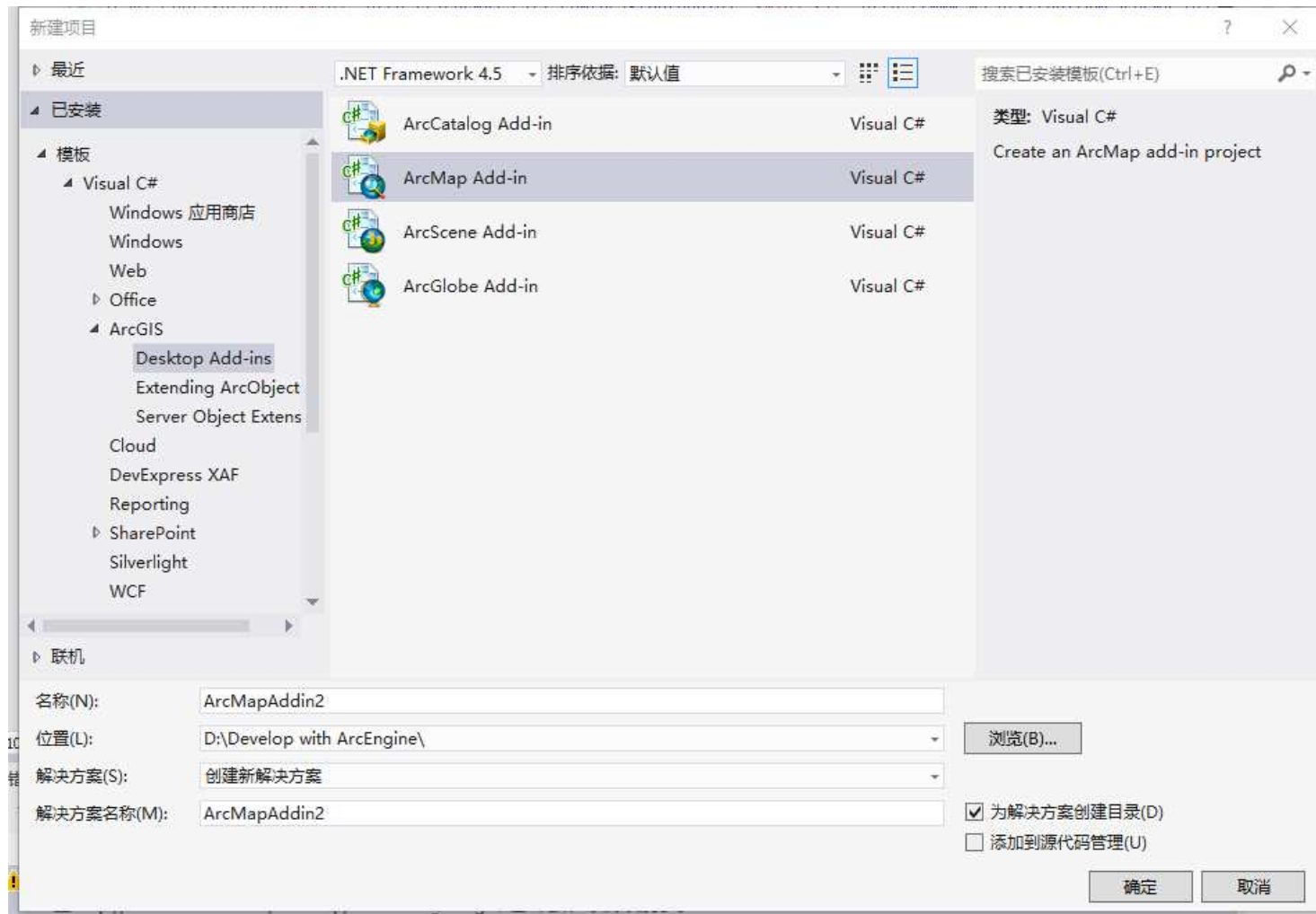
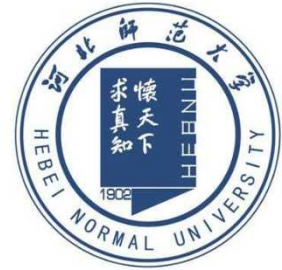
- .NET4框架引入了一个改进的安全模式，.NET4框架不会自动使用当前版本通用语言运行时刻库来运行使用以前版本的.NET框架构建应用程序。
- 因此，在.NET4框架下运行以前的应用程序时，必须使用Visual Studio项目属性中指定的目标.NET框架来重新编译应用程序，或者应用程序配置文件中元素来指定支持的运行时刻库。（可能会带来异常）

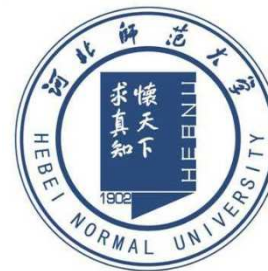


练习题


- 1、简述类和接口的定义，并对两者的异同点进行描述。
- 2、给出COM模型的基本思想，列举常用的COM技术标准
- 3、什么是QI接口，QI接口的作用是什么，举例说明QI的使用方式。
- 4、简述AO与AE的定义和区别。

addin







ArcGIS Add-Ins Wizard


 **Welcome**
Enter information for the add-in.

Welcome

Add-in Types

Add-in Name:  ArcMapAddin1

Company/
Publisher: 

Author:  chango




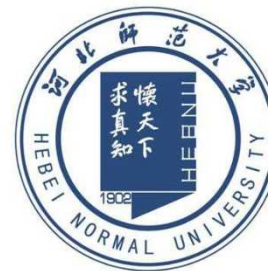
Description: 

Image:  



ArcGIS Add-Ins Wizard

Available Add-in Components

Check components to be included in this Add-in project and modify its properties.

Welcome

Add-in Types

- ☒ Button
- ☐ Tool
- ☐ Combo Box
- ☐ Multi-Item
- ☐ Construction Tool
- ☐ Extension
- ☐ Editor Extension
- ☐ Dockable Window

Class Name:

Caption:

Image:

Category:

On Demand: ☒

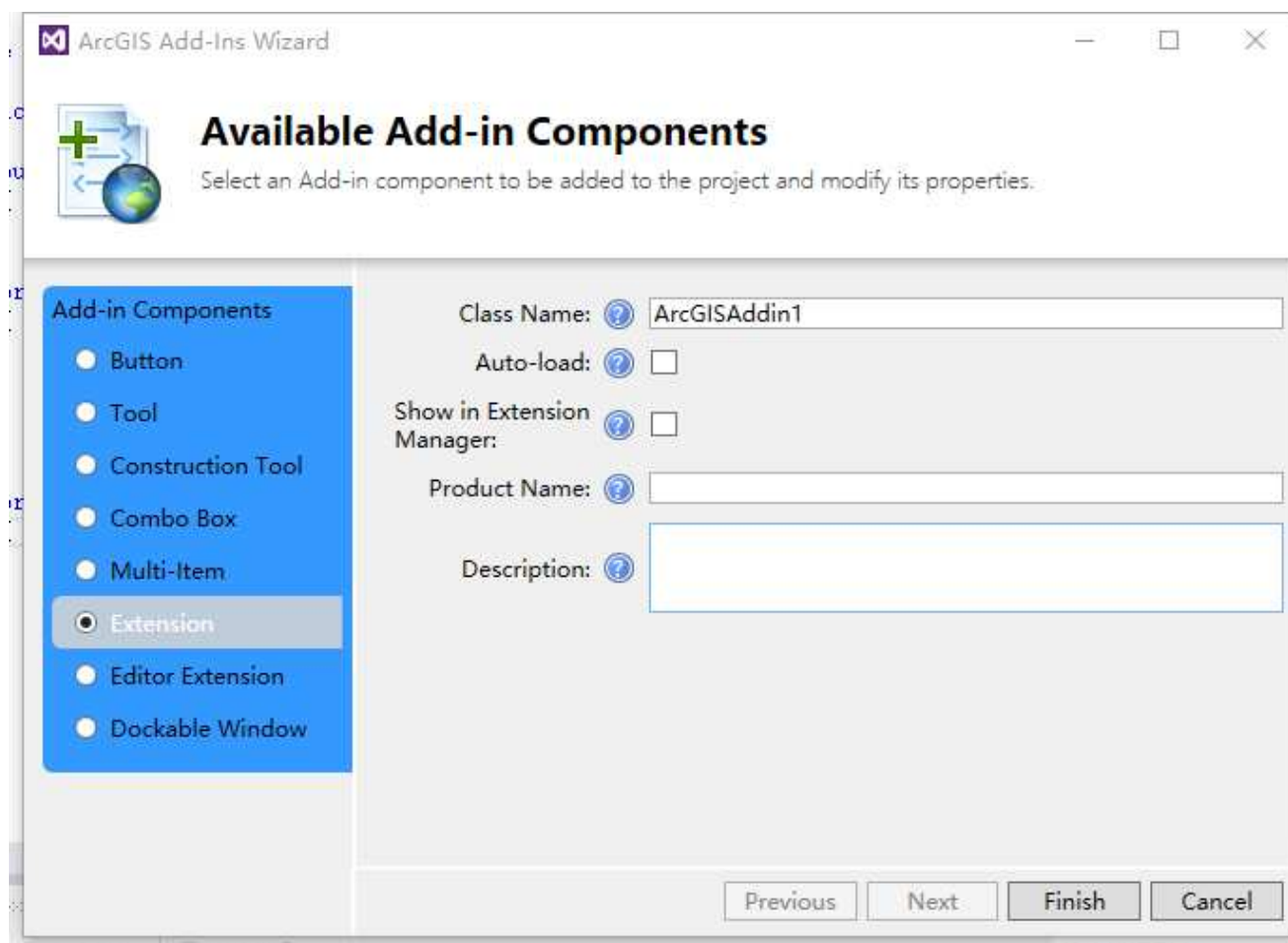
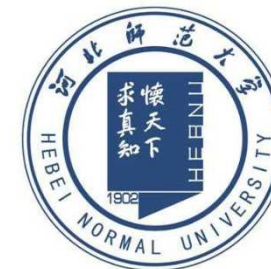
Tooltip:

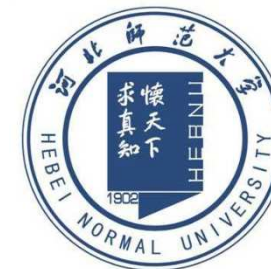
Description:

Use File Source: ☒

Previous Next Finish Cancel

添加项





ArcGIS Add-Ins Wizard

Available Add-in Components

Select an Add-in component to be added to the project and modify its properties.

Add-in Command Bars

- ☒ Tool Palette
- ☐ Toolbar
- ☐ Menu
- ☐ Context Menu

Caption:

Category:

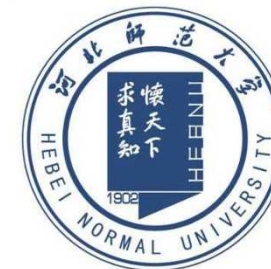
Menu Style: ☐

Columns:

Items:




Reference ID	SubType	Separator
Add item...		

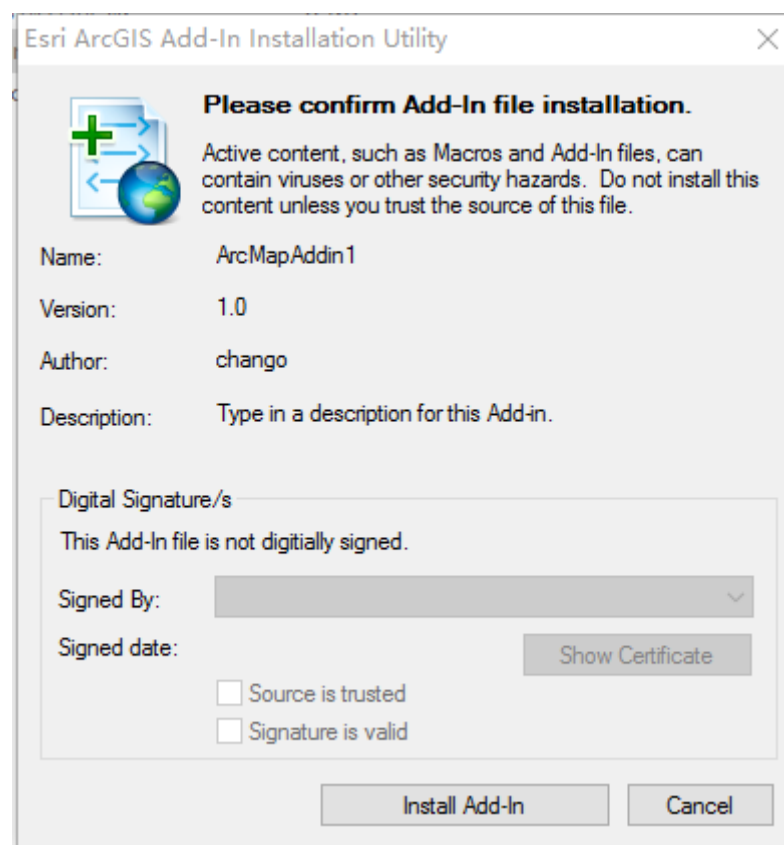
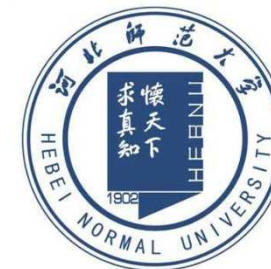
Previous Next Finish Cancel

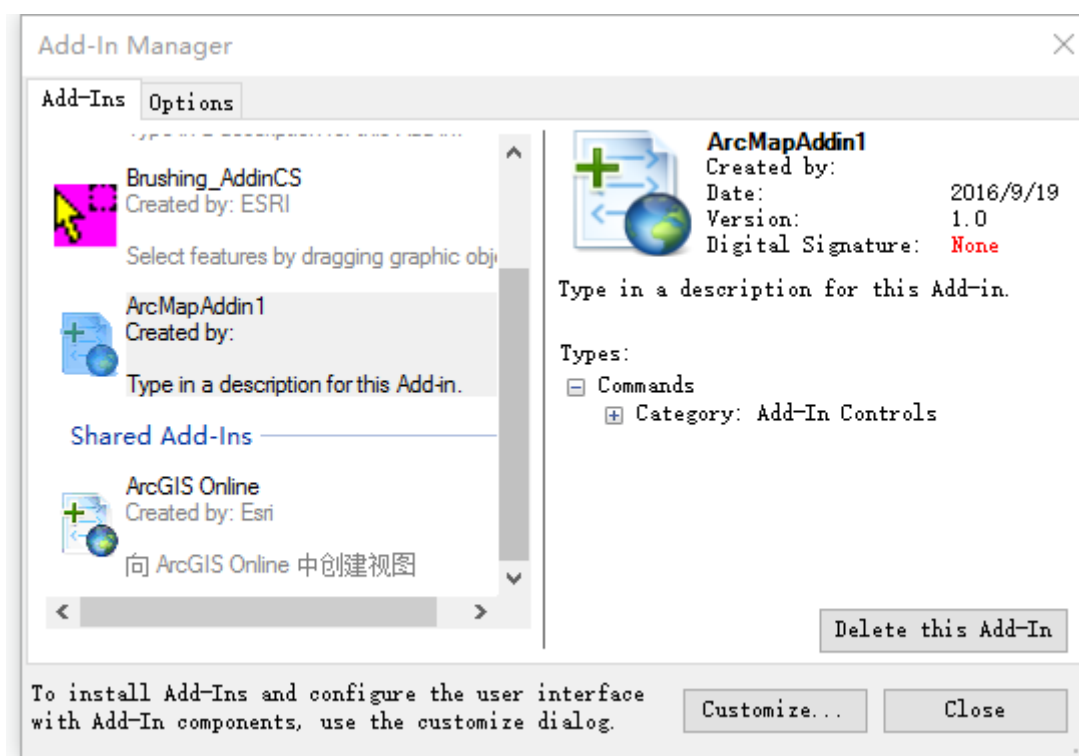
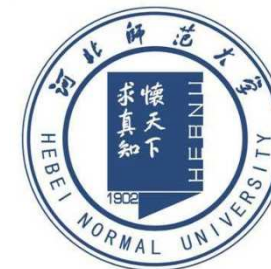


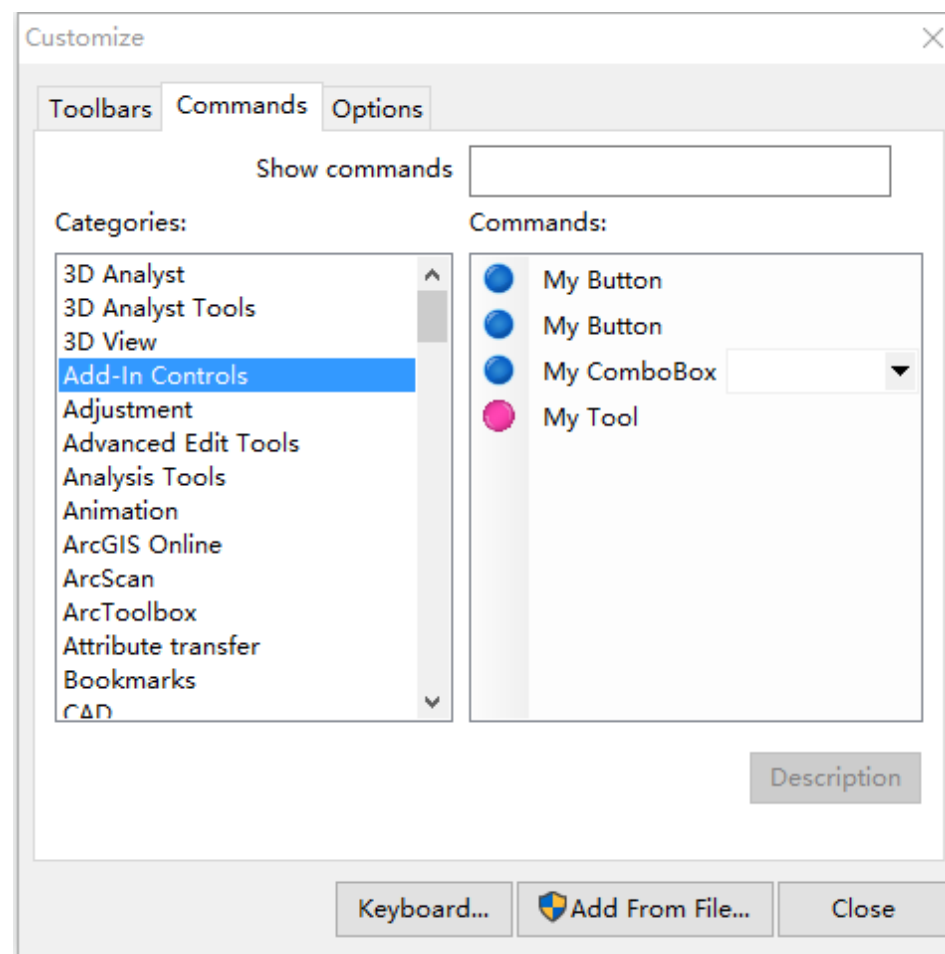
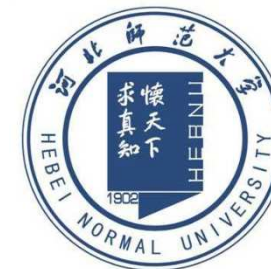
addin安装部署

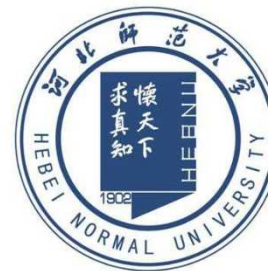
电脑 > 本地磁盘 (D:) > Develop with ArcEngine > ArcMapAddin1 > ArcMapAddin1 > bin > Debug

名称	修改日期	类型	大小
 ArcMapAddin1.dll	2016/9/19 15:44	应用程序扩展	8 KB
 ArcMapAddin1.esriAddIn	2016/9/19 15:44	Esri AddIn File	38 KB
 ArcMapAddin1.pdb	2016/9/19 15:44	Program Debug...	24 KB

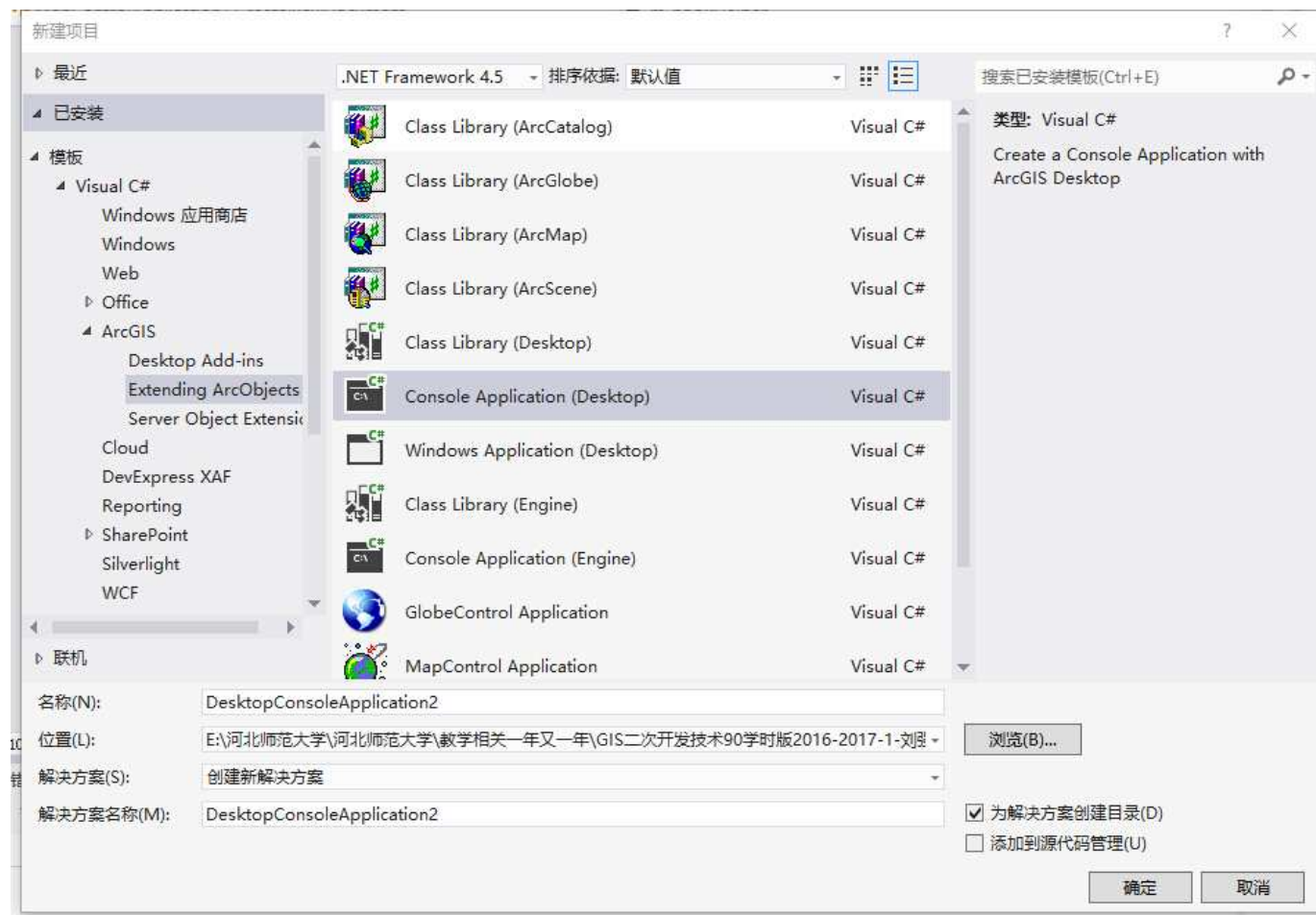




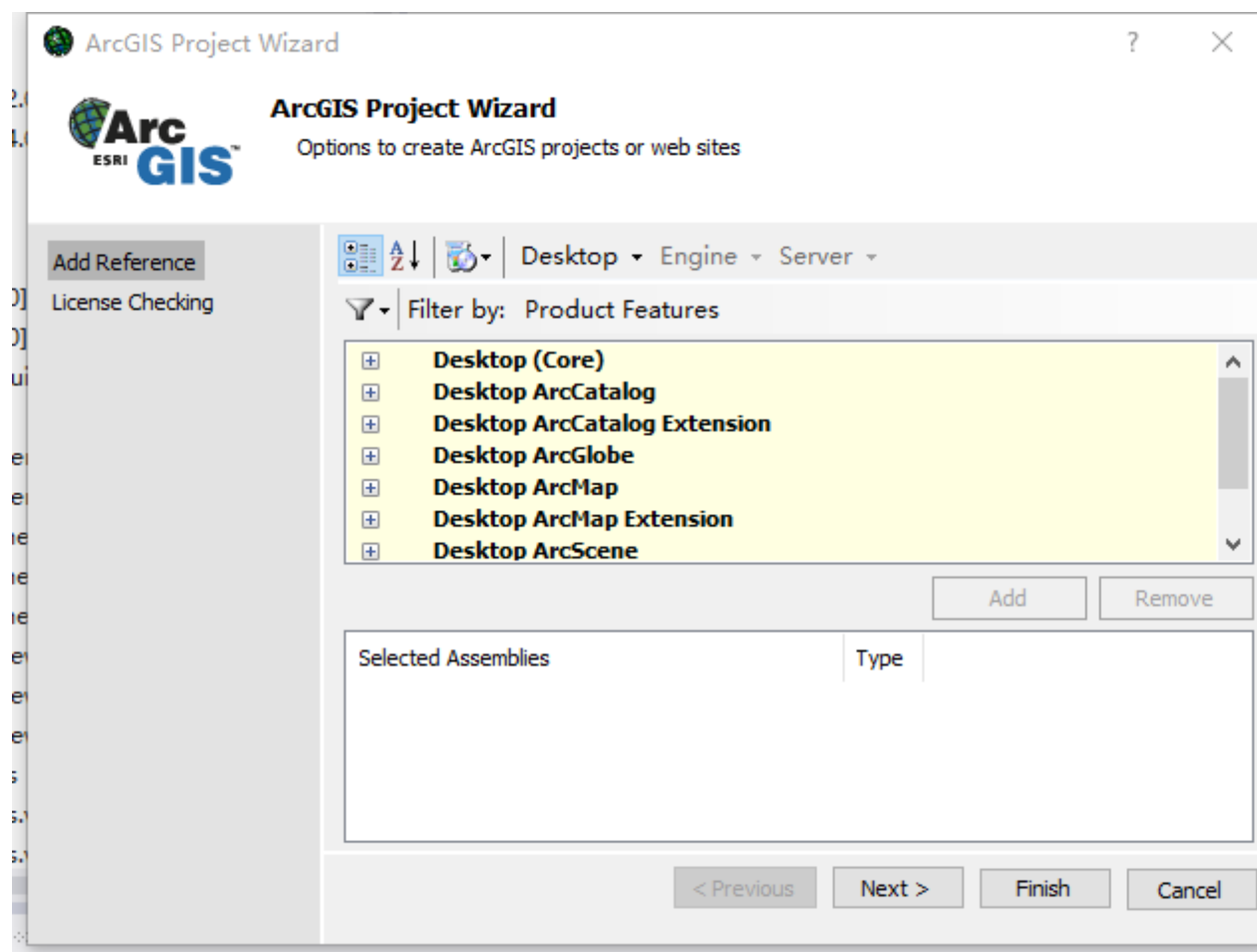
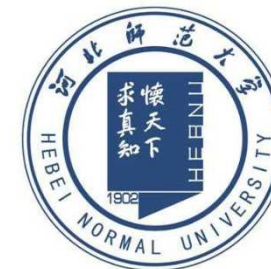


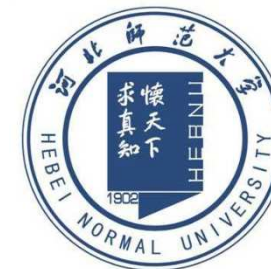


项目模板

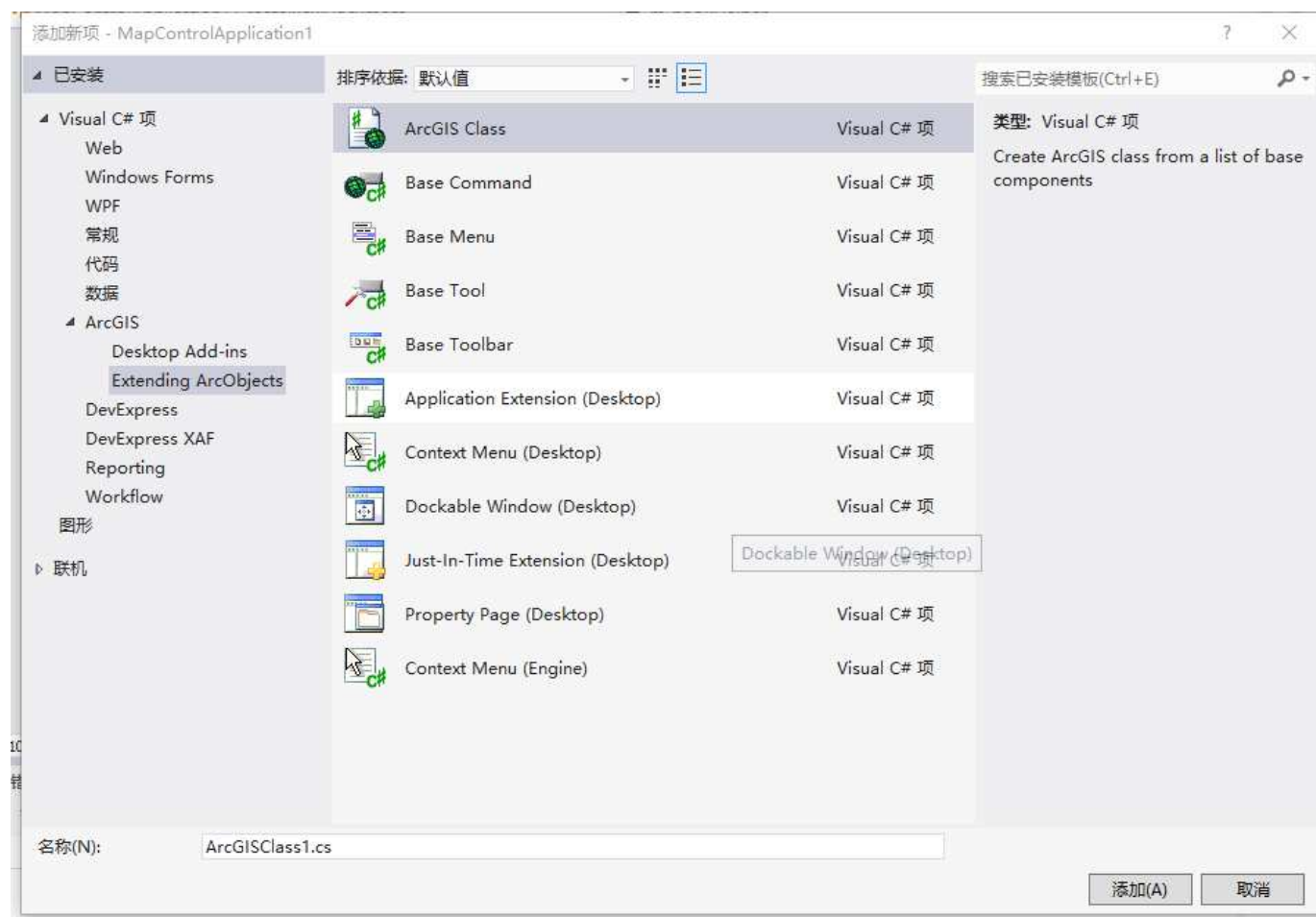


项目向导

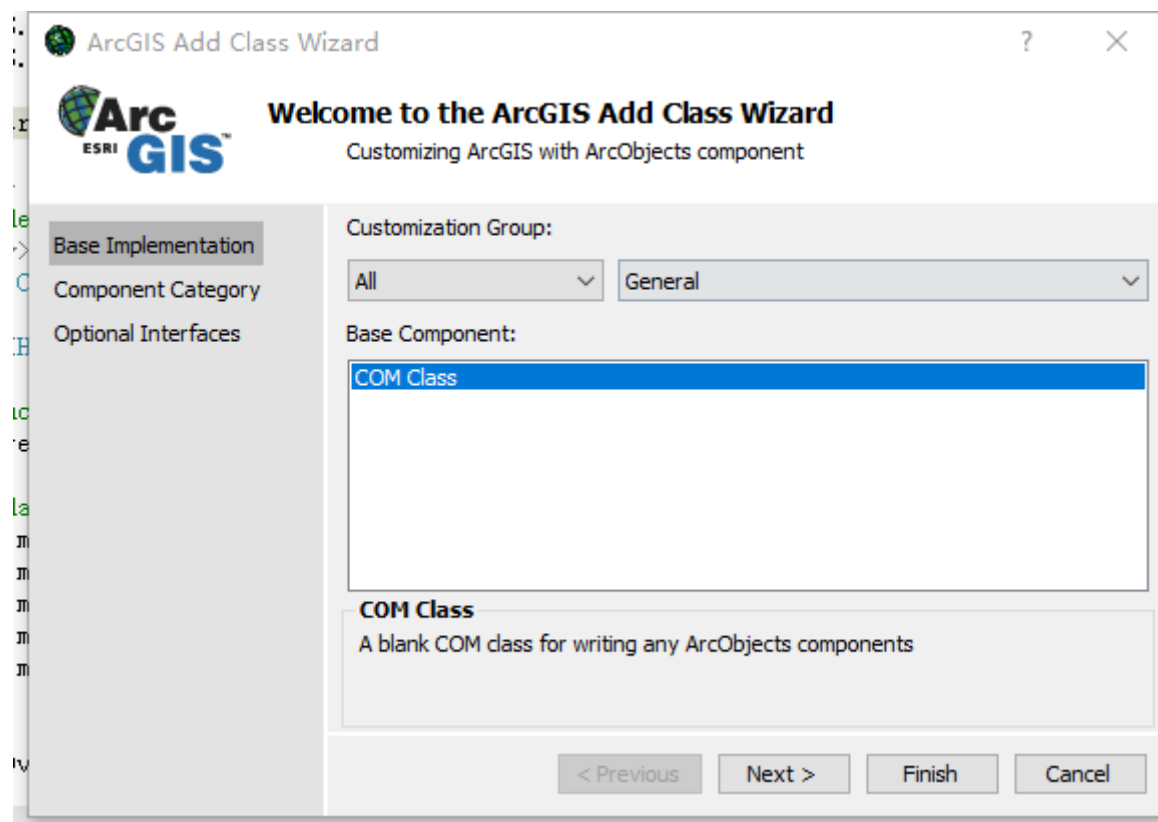
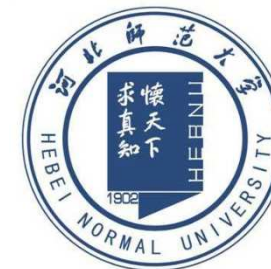


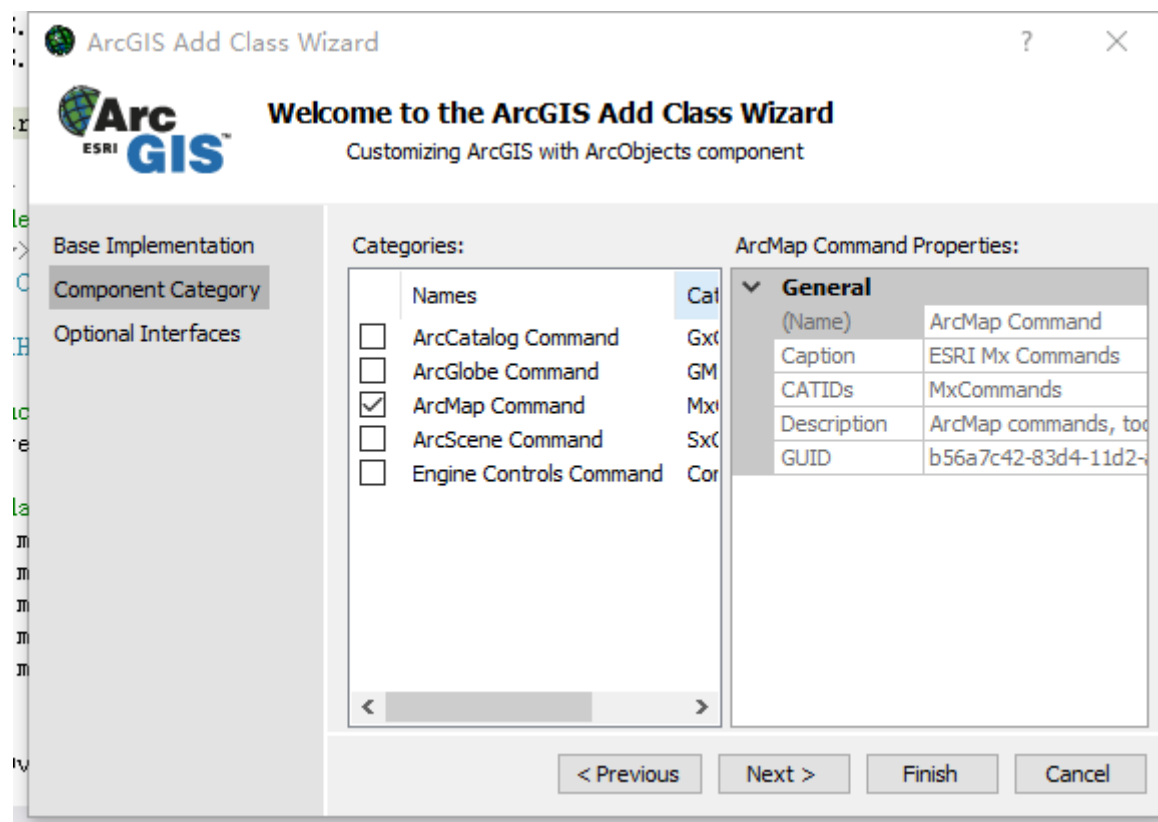
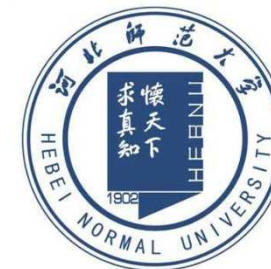


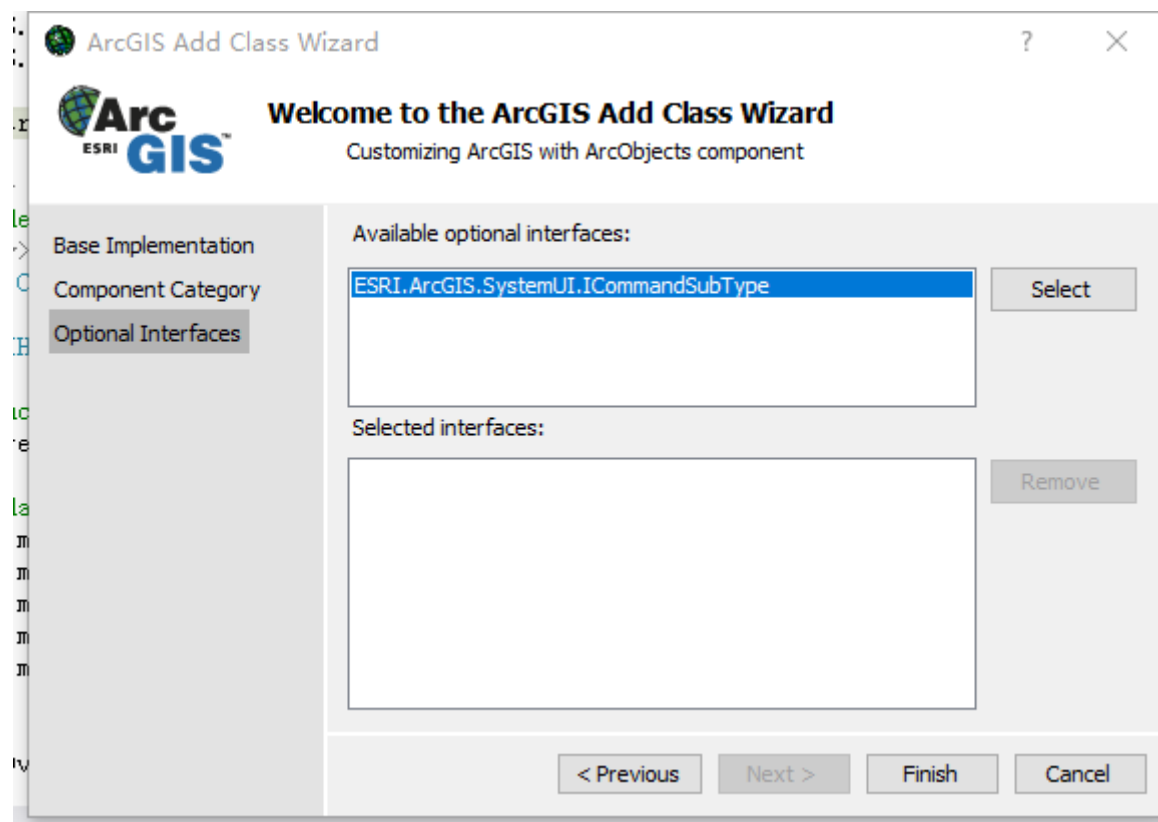
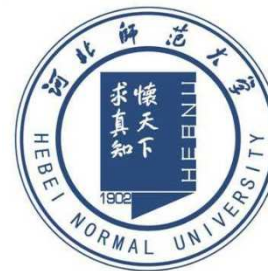
类模板

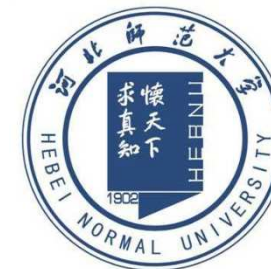


ArcGIS Class

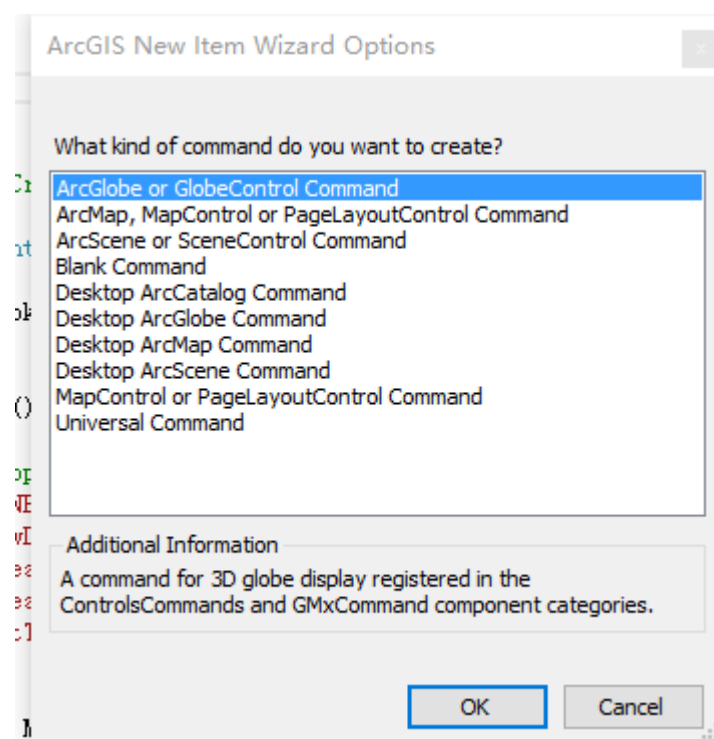


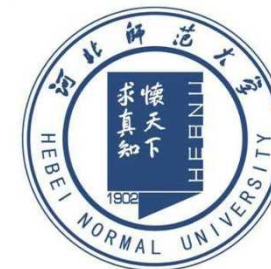




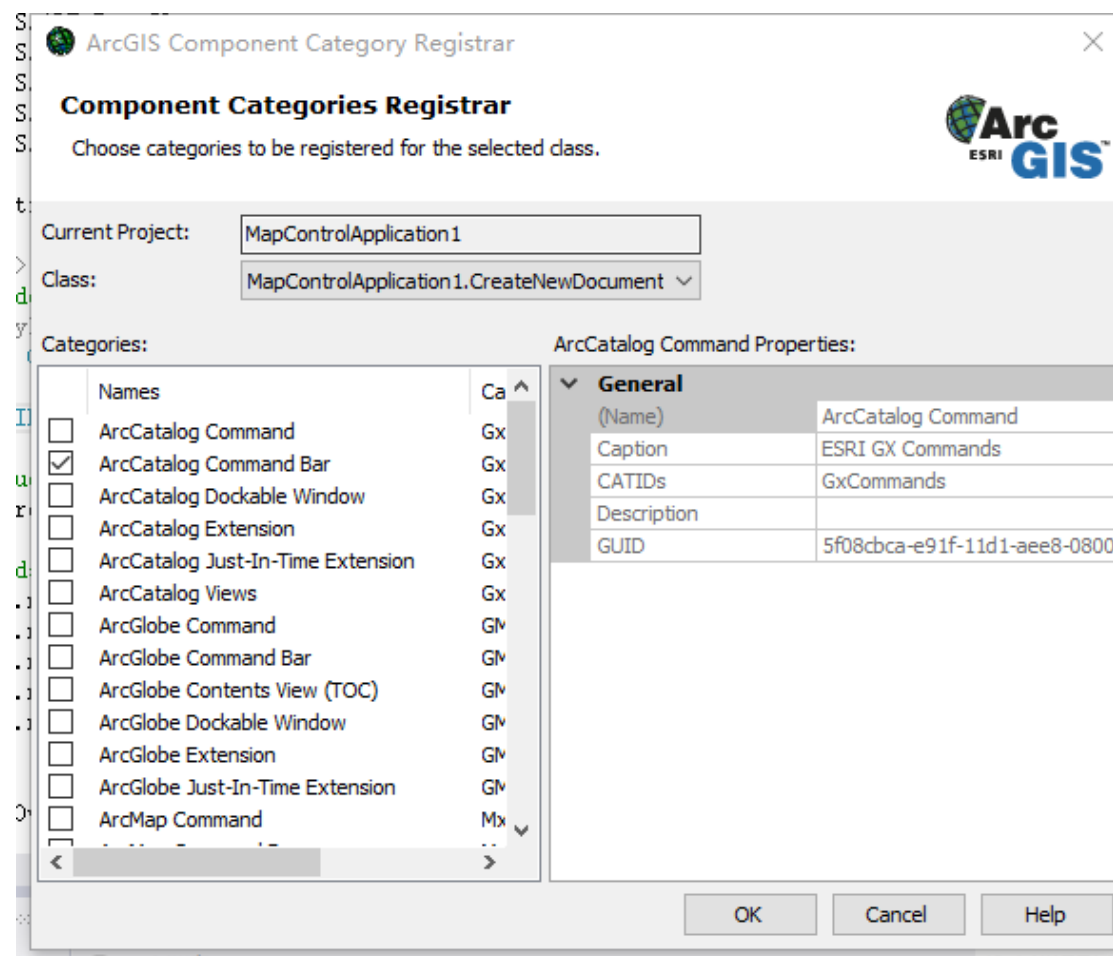


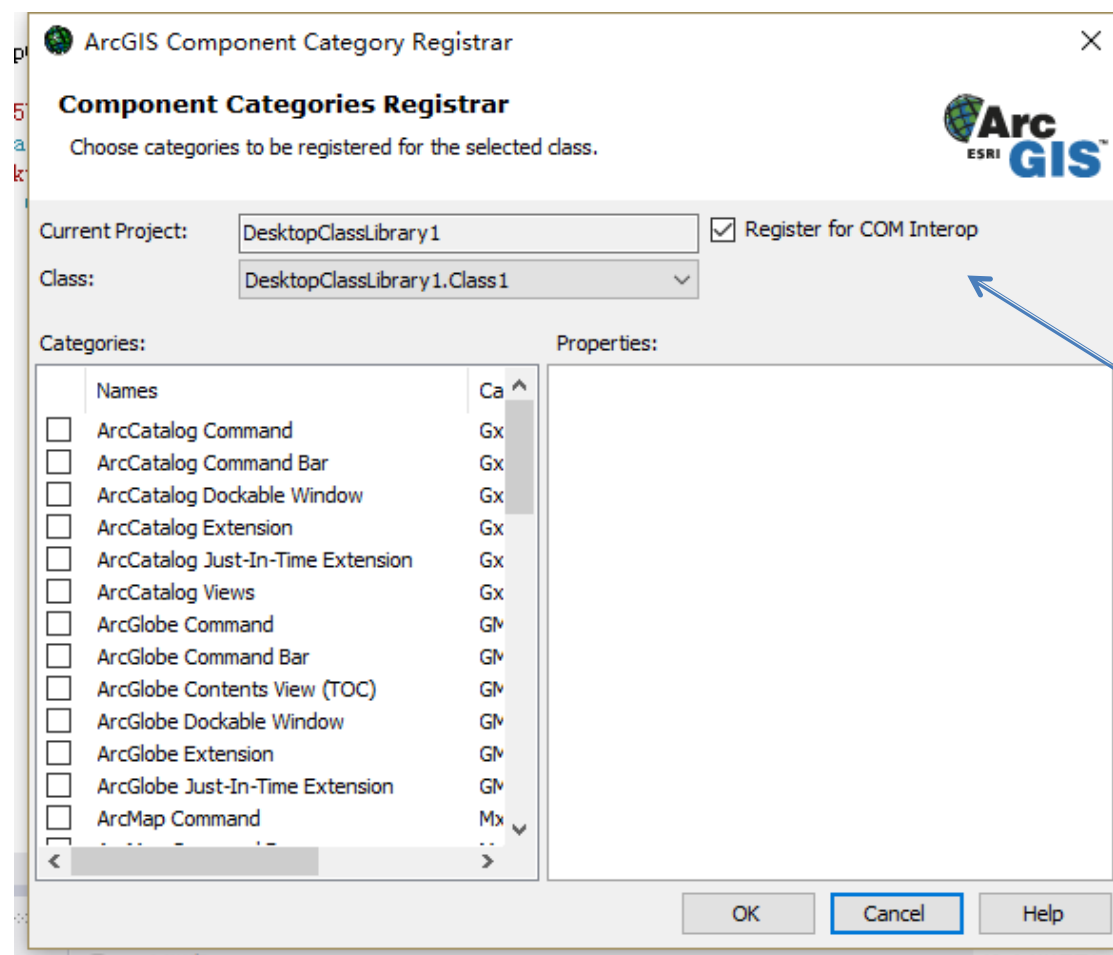
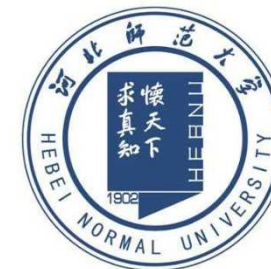
Base command and base tool

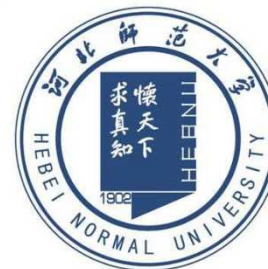




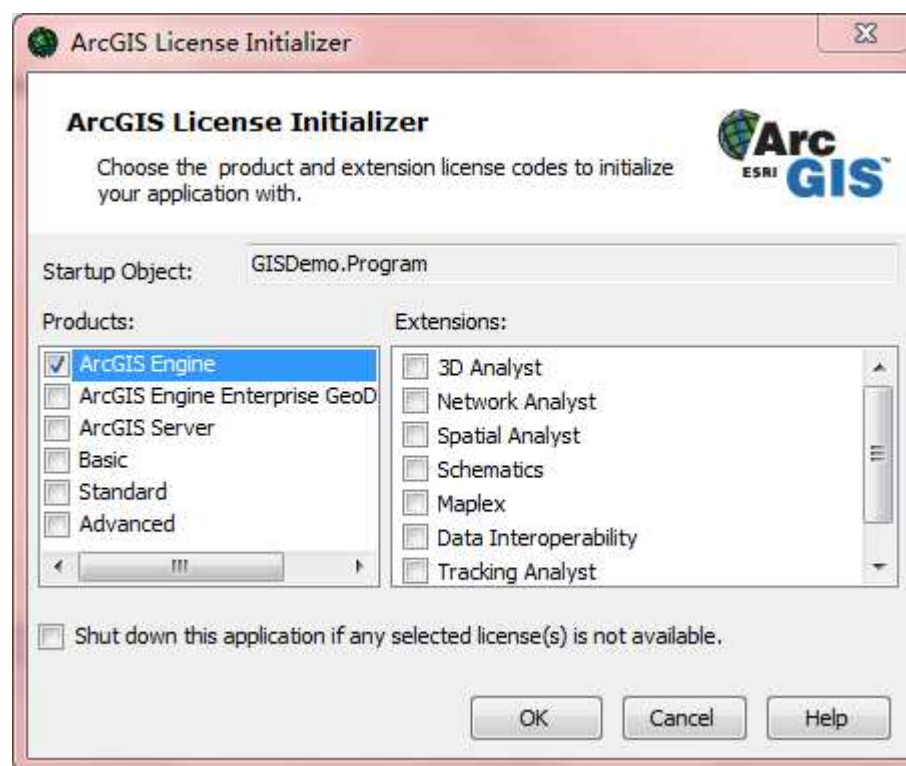
Component Category Registrar

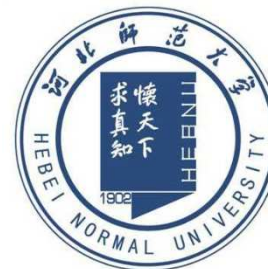




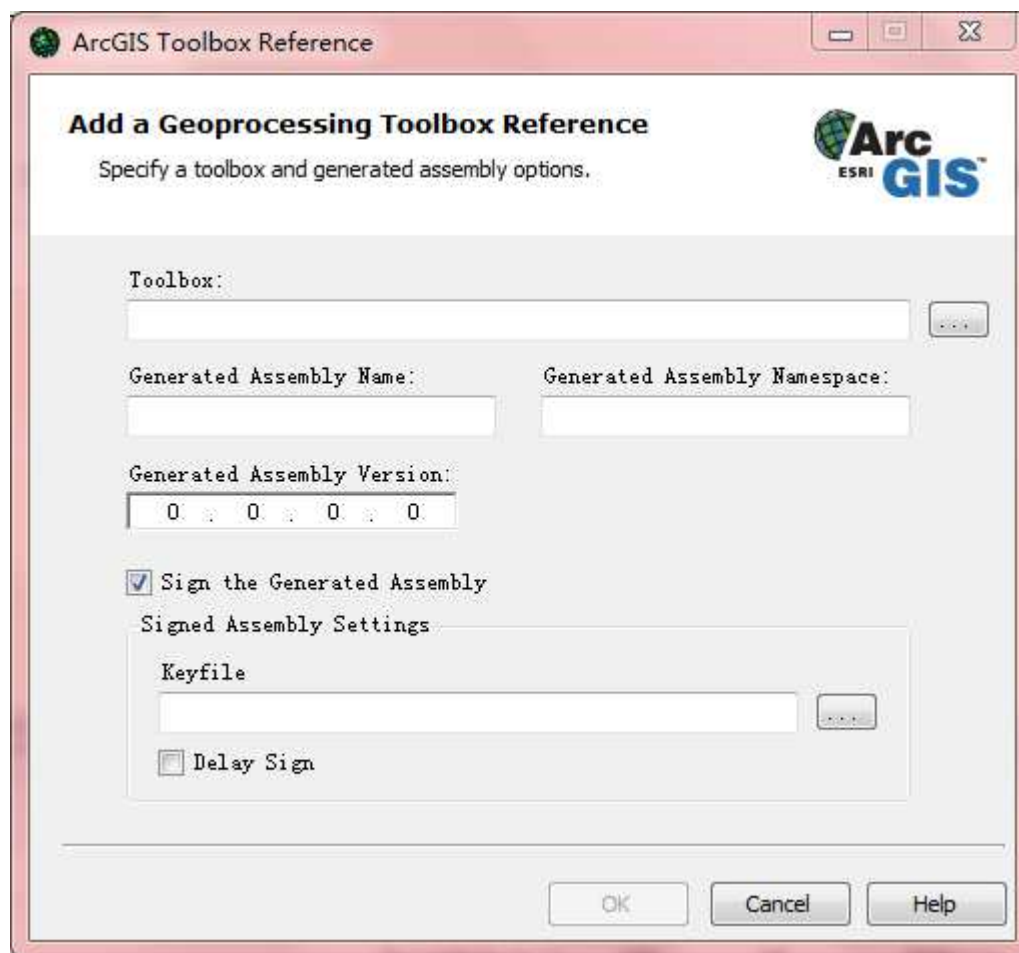


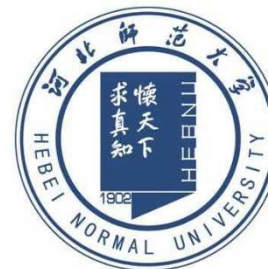
License_INITIALIZER



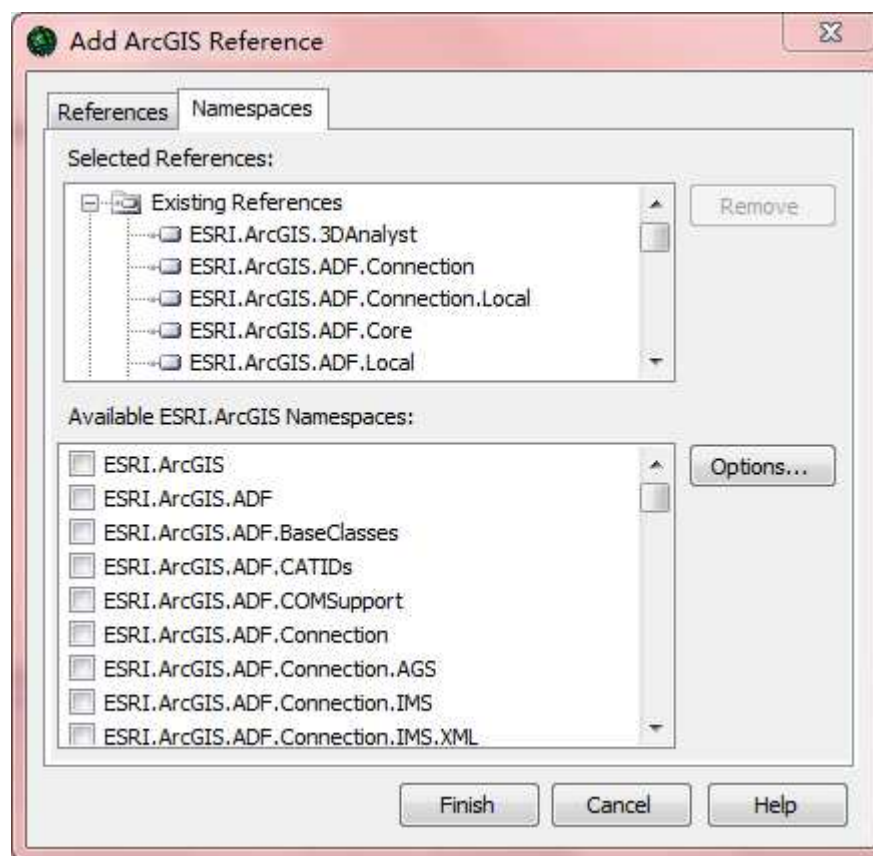


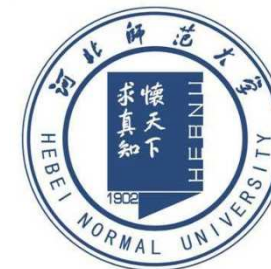
Toolbox Reference





Add Reference





代码片段

	查看设计器(D)	Shift+F7
	重构(R)	▶
	组织 using(O)	▶
	生成序列图(N)...	
	运行测试(T)	Ctrl+R, T
	调试测试(D)	Ctrl+R, Ctrl+T
ArcGIS Snippet Finder		
ArcGIS Snippet Editor Wizard		
	Insert ArcGIS Using Directives	
	插入代码段(I)...	Ctrl+K, X
	外侧代码(S)...	Ctrl+K, S
	转到定义(G)	F12
	查找所有引用(A)	Ctrl+K, R
	查看调用层次结构(H)	Ctrl+K, Ctrl+T
	断点(B)	▶
	运行到光标处(N)	Ctrl+F10
	将标记的线程运行到光标处(F)	
	剪切(T)	Ctrl+X
	复制(Y)	Ctrl+C
	粘贴(P)	Ctrl+V
	大纲显示(L)	▶
	在解决方案中查找匹配的克隆(M)	

