地理信息系统应用程序设计与开发

第六章 AO管理空间数据

主讲：张宝一
Email：zhangbaoyi.csu@qq.com

教学目标

- 掌握GeoDatabase的概念及其数据类型
- 掌握Workspace及其相关对象的应用
- 熟悉Dataset对象及其获取工作区要素类的方法
- 掌握FeatureClass的创建、获取等常用操作
- 掌握Feature的创建、修改、删除等常用操作

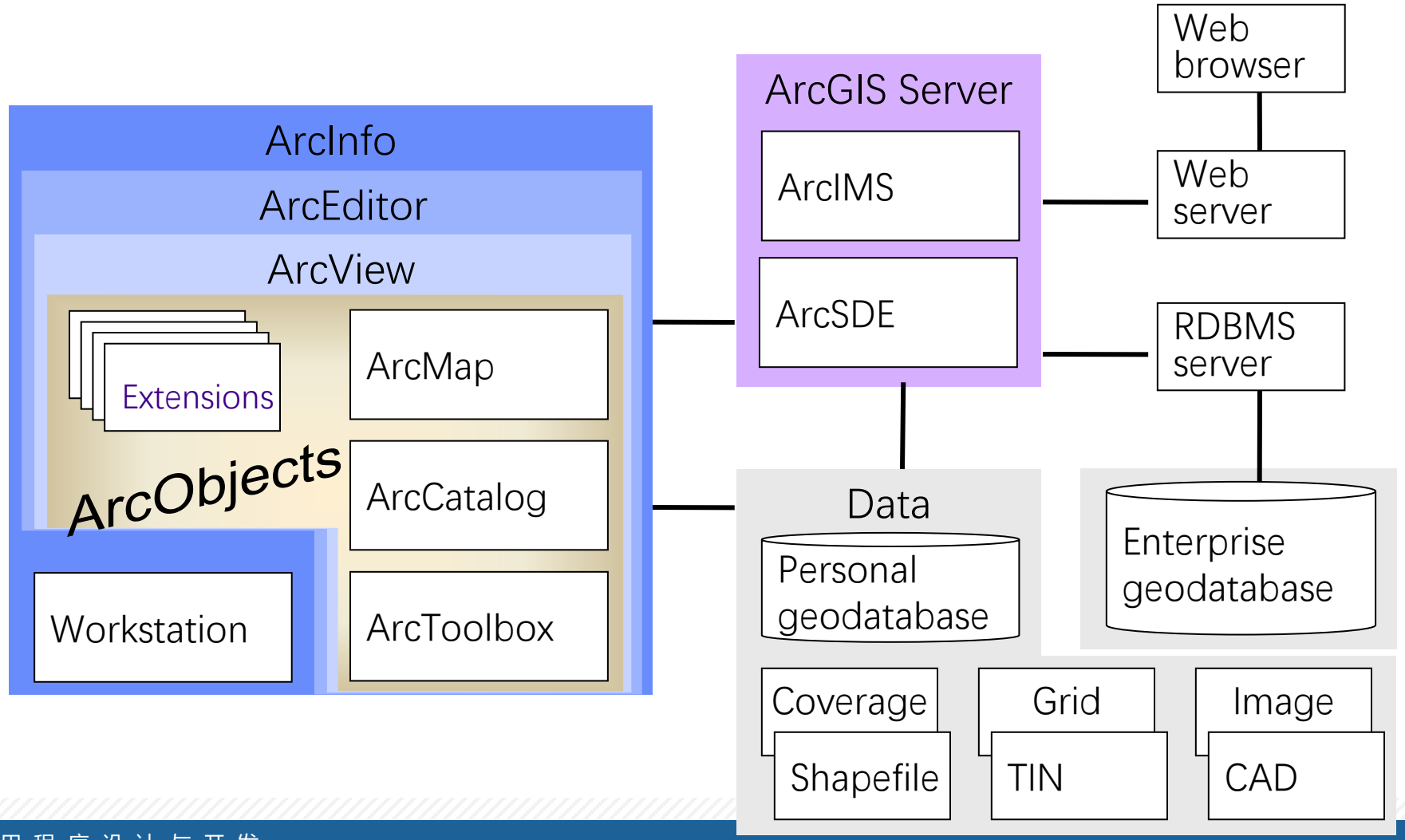
教学重点和难点

- Workspace的常用操作
- FeatureClass 和 Feature的常用操作

教学内容

- 6.1 GeoDatabase概念
- 6.2 Workspace及相关对象
- 6.3 Dataset对象
- 6.4 FeatureClass管理
- 6.5 Feature管理
- 6.6 开发实例

ArcGIS的GeoDatabase

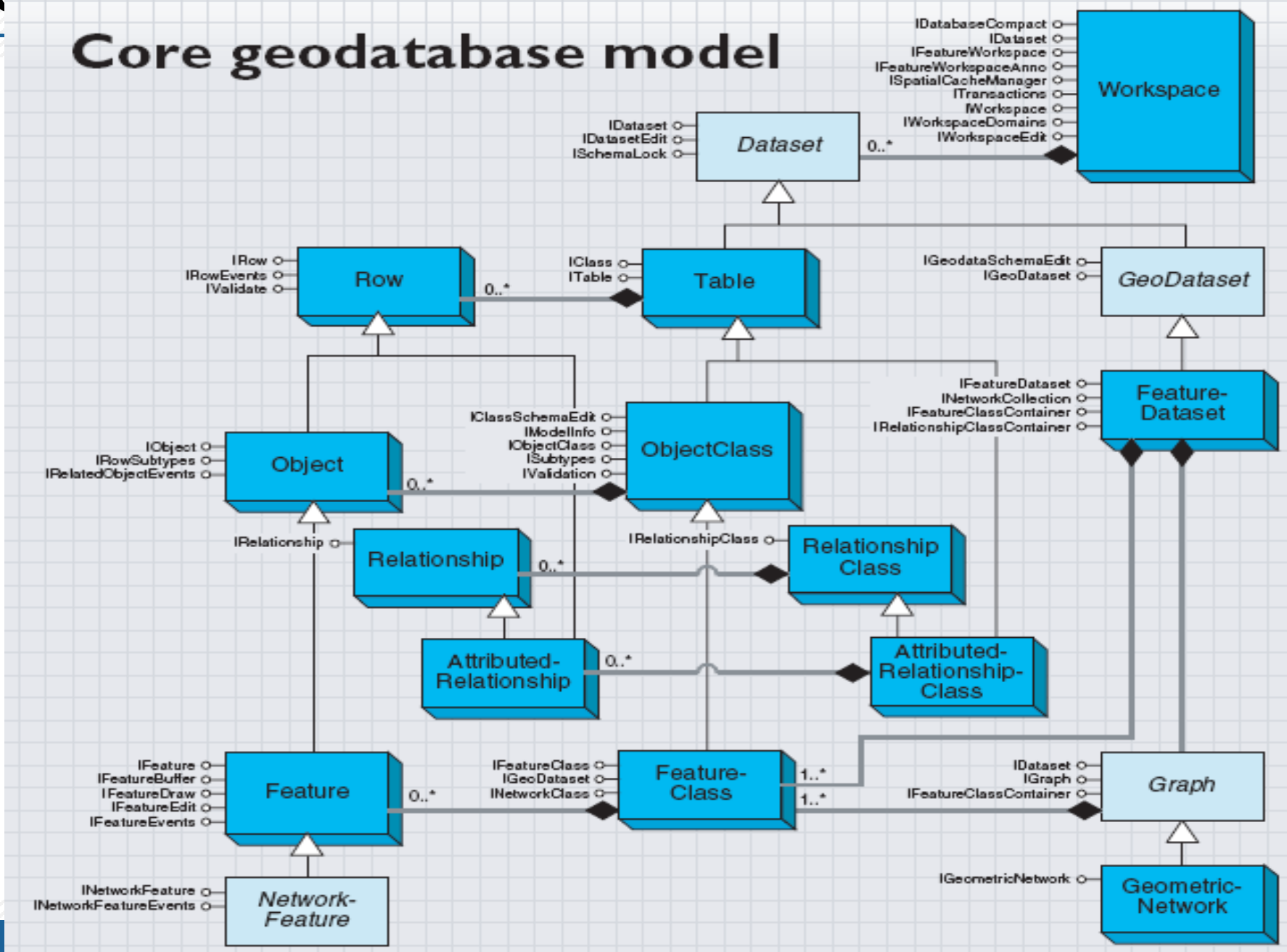


6.1 GeoDatabase概念

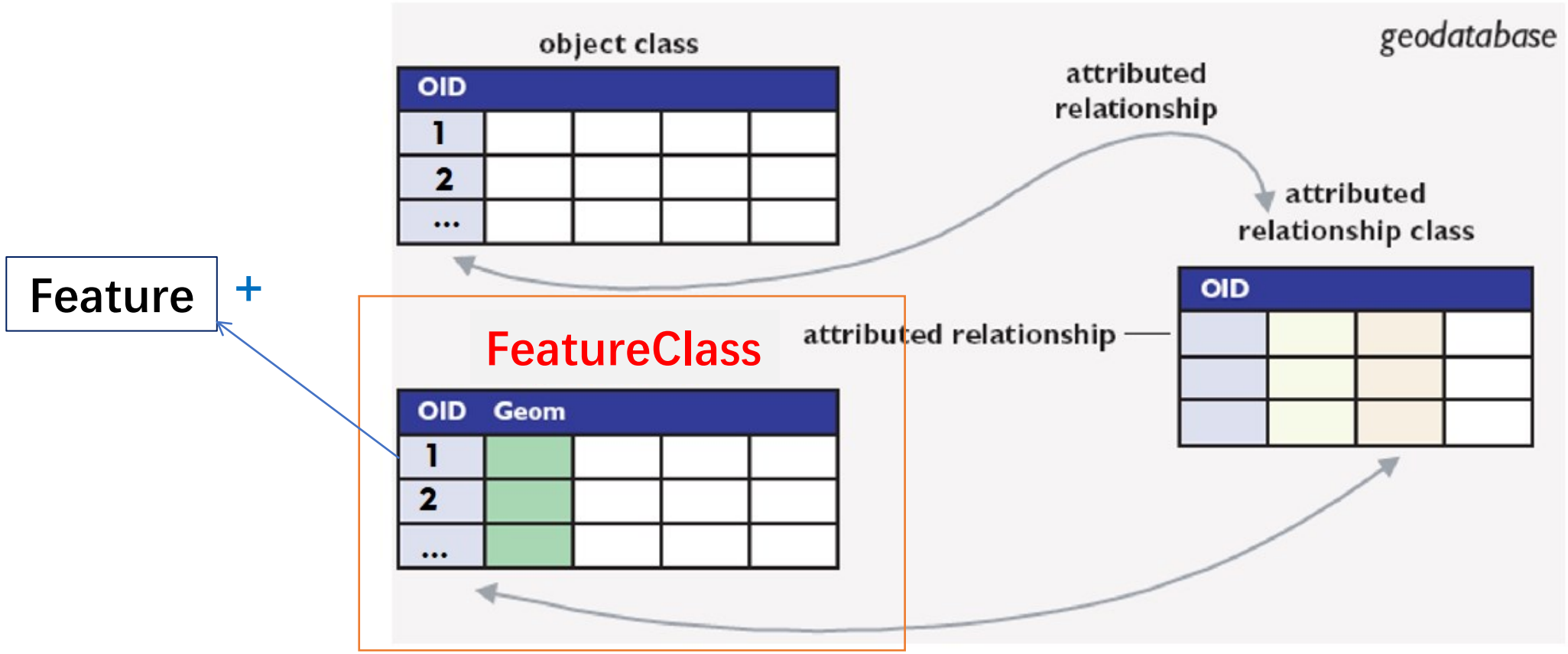
- GeoDatabase是“面向对象数据库”,具有以下特性:
  - 多态性
    - 多种类型的数据源，操作方式都一样
  - 封装性
    - 不需要内部的具体的工作机理
  - 继承性
    - 在已经存在的对象上继承和派生出具有新的功能的对象

6.1 GeoDatabase概念

- Workspace对象
  - 代表地理数据库
  - ShapeFile文件夹
  - 文件数据库
  - CAD文件
  - EXCEL文件
  - MDB数据库
- Dataset
  - 数据集
- GeoDataset
  - 地理数据集
- FeatureDataset
  - 要素类数据集



- 表、行(记录)、要素类和要素


geodatabase
object class
OID
1
2
...
attributed
relationship
attributed
relationship class
OID
attributed relationship
Feature
+
FeatureClass
OID
Geom
1
2
...

6.2 Workspace

- ArcEngine通过Workspace（工作区）访问GeoDataset（地理数据集）
  - 是连接空间数据的通道
  - 包含了若干个数据集的数据库或数据源，数据源可以是表、要素类、关系类等
- Workspace的接口
  - IWorkspace，IFeatureWorkspace，IRasterWorkspace，等等
- Workspace由工作区空间工厂（WorkspaceFactory）创建
  - Shp文件工作区工厂 ：ShapefileWorkspceFactory
  - 个人空间数据库工作区工厂 ：AccessWorkspaceFactory
  - SDE企业数据库工作区工厂 ：SdeWorkspaceFactory
  - 文件地理数据库工作区工厂 ：FileGDBWorkspaceFactory
  - EXCEL文件工作区工厂 ：ExcelWorkspaceFactory
  - CAD数据源工作区工厂 ：CadWorkspaceFactory
  - OLEDB数据源工作区工厂 ：OLEDBWorkspaceFactory
  - 内存数据源工作区工厂 ：InMemoryWorkspaceFactory

- [示例] 打开SHP文件

① IWorkspace ws = null;
② IWorkspaceFactory wsf = new ShapefileWorkspaceFactory();
③ ws = wsf.OpenFromFile(@"d:\temp\csu", 0);
④ IFeatureWorkspace fws = (IFeatureWorkspace)ws;
⑤ IFeatureClass fc = fws.OpenFeatureClass("jmd.shp");

⑥ IFeatureLayer layer = new FeatureLayer();
⑦ layer.FeatureClass = fc;
⑧ layer.Name = fc.AliasName;
⑨ this.axMapControl1.AddLayer(layer);

- [示例] 打开个人空间数据库要素类

① IWorkspace ws = null;
② IWorkspaceFactory wsf = new AccessWorkspaceFactory();
③ ws = wsf.OpenFromFile(@"d:\temp\csu.mdb", 0);
④ IFeatureWorkspace fws = (IFeatureWorkspace)ws;
⑤ IFeatureClass fc = fws.OpenFeatureClass("jmd");

⑥ IFeatureLayer layer = new FeatureLayer();
⑦ layer.FeatureClass = fc;
⑧ layer.Name = fc.AliasName;
⑨ this.axMapControl1.AddLayer(layer);

6.3 Dataset对象

DataSet对象

- 代表了Workspace中数据集合的抽象类，是一个高级别的数据容器。
- Dataset分为两类
  - Table
    - 记录Row集合
  - GeoDataset
    - 要素类集合

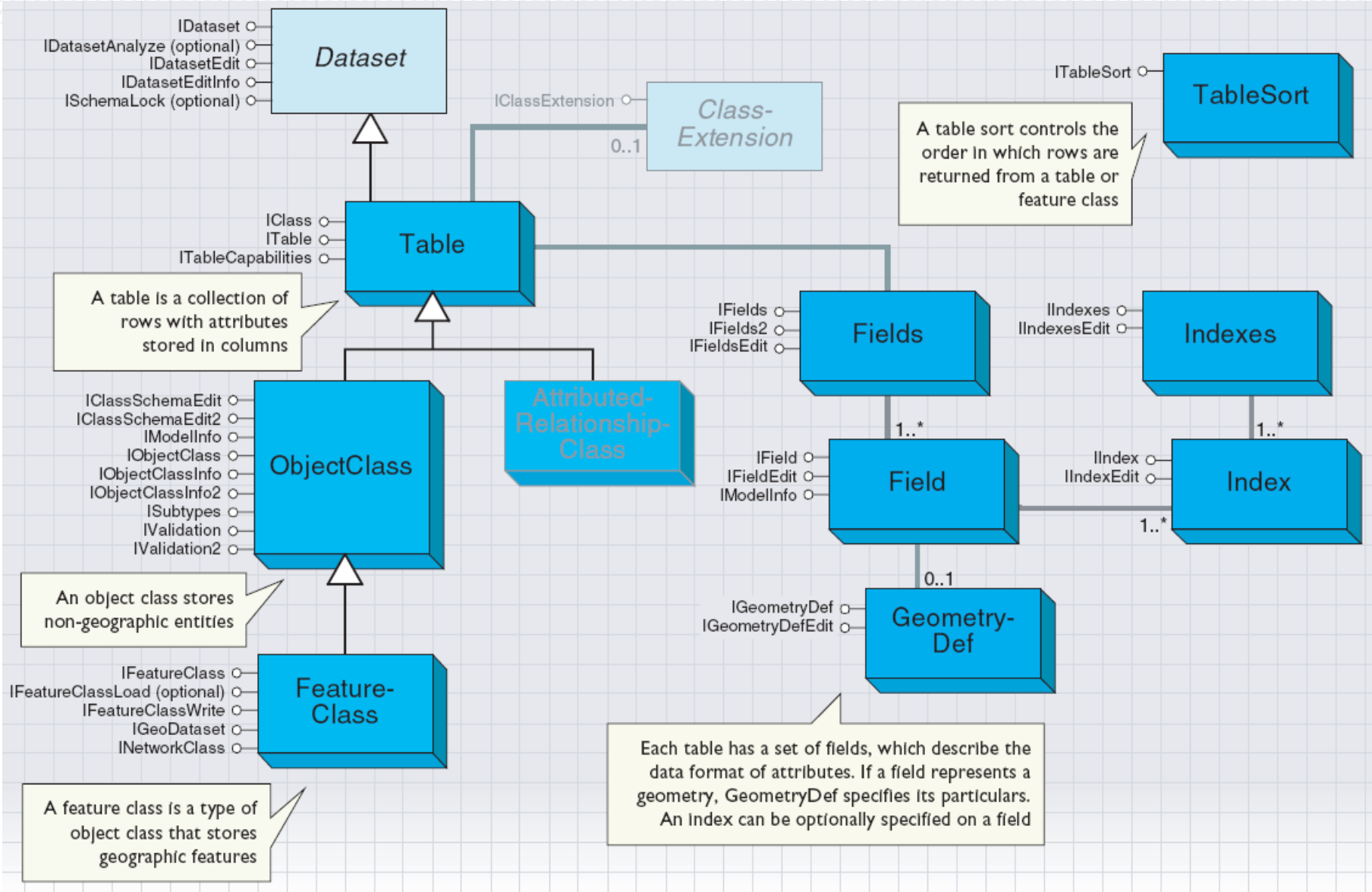
[示例] 获取shp文件目录下所有shp文件

```
① IWorkspace ws = null;
② IWorkspaceFactory wsf = new ShapefileWorkspaceFactory();
③ ws = wsf.OpenFromFile(@"d:\temp\csu", 0);
④ IFeatureWorkspace fws = (IFeatureWorkspace)ws;
⑤ List<IFeatureClass> list = new List<IFeatureClass>();
⑥ IEnumDatasetName datasetName =
   ws.DatasetNames[esriDatasetType.esriDTFeatureClass]; //筛选出shp
⑦ IDatasetName dn = datasetName.Next(); //获取的数据集名称无后缀名
⑧ while (dn != null)
⑨ {
⑩       list.Add(fws.OpenFeatureClass(dn.Name));
⑪       dn = datasetName.Next();
⑫ }
```

思考：如何遍历mdb中所有要素类？

6.4 FeatureClass管理

- FeatureClass
  - 是一类Feature的集合
  - 继承ObjectClass
  - ObjectClass继承Table
- Table
  - 数据库中的一张表
  - 存储数据
  - 行/记录的集合



6.4 FeatureClass管理

- FeatureClass类实现的常用接口
  - **IFeatureClass** : 要素类属性、方法,用于创建要素、编辑字段等
  - IFeatureClassManage : 更新要素类的Extent,方法: UpdateExtent( )
  - IFeatureClassWrite : 底层要素类数据管理，包括删除、保存要素等。
  - IDataset : 数据集管理，包括获取/修改名称、删除数据集等
  - IDatasetAnalyze : 分析数据
  - IDatasetEdit : 访问数据集编辑状态
  - IDatasetEditInfo : 获取数据集是否可编辑、是否可撤销/重做等
  - ITable : 管理数据表，包括字段管理、数据查找等
  - IValidation : 管理数据有效性规则
  - … …

6.4 FeatureClass管理

(1)创建

```
/// <summary>
/// 使用要素类描述对象创建最简单的要素类，只包含2个必要字段“SHAPE”和“OBJECTID”，无空间参考
/// </summary>
/// <param name="featureWorkspace">目标工作空间</param>
public void CreatSimpleFeatureClass(IFeatureWorkspace featureWorkspace)
{
    //ESRI 要素类描述对象
    IFeatureClassDescription fcDesc = new FeatureClassDescriptionClass();
    IObjectClassDescription ocDesc = (IObjectClassDescription)fcDesc;
    IFeatureClass targetFClass = featureWorkspace.CreateFeatureClass("描述对象",
        ocDesc.RequiredFields,
        null, null, esriFeatureType.esriFTSimple, fcDesc.ShapeFieldName, "");
}
```

6.4 FeatureClass管理

- (1)创建 (默认字段)
  - IWorkspaceFactory wf = new ShapefileWorkspaceFactory();
  - IFeatureWorkspace fws = wf.OpenFromFile(folder, 0) as IFeatureWorkspace;

```
② IFeatureClassDescription fcDesc = new FeatureClassDescriptionClass();
③ IObjectClassDescription ocDesc = (IObjectClassDescription)fcDesc;
④ // Use IFieldChecker to create a validated fields collection.
⑤ IFieldChecker fieldChecker = new FieldCheckerClass();
⑥ IEnumFieldError enumFieldError = null;
⑦ IFields validatedFields = null;
⑧ fieldChecker.ValidateWorkspace = (IWorkspace)fws;
⑨ fieldChecker.Validate(fieldsCollection, out enumFieldError, out validatedFields);
⑩ IFeatureClass featureClass = fws.CreateFeatureClass( “abc.shp”, validatedFields,
   ocDesc.InstanceCLSID, ocDesc.ClassExtensionCLSID,
   esriFeatureType.esriFTSimple, shapeFieldName, "");
```

生成OID、Shape字段

生成有效字段列表

6.4 FeatureClass管理

- (1)创建（自定义字段)

```
① IWorkspaceFactory wf = new ShapefileWorkspaceFactory();
② IFeatureWorkspace fws = wf.OpenFromFile(folder, 0) as IFeatureWorkspace;
③ // 生成默认字段
④ IFeatureClassDescription fcDesc = new FeatureClassDescriptionClass();
⑤ IObjectClassDescription ocDesc = (IObjectClassDescription)fcDesc;
⑥ IFields fields = ocDesc.RequiredFields;
⑦ int shapeFieldIndex = fields.FindField(fcDesc.ShapeFieldName);
⑧ IField field = fields.get_Field(shapeFieldIndex);
⑨ IGeometryDef geometryDef = field.GeometryDef;
⑩  IGeometryDefEdit geometryDefEdit = (IGeometryDefEdit)geometryDef;
⑪ geometryDefEdit.GeometryType_2 = esriGeometryType.esriGeometryPoint;
```

(1)创建（自定义字段)续

```
⑫  //添加自定义字段
⑬  IFieldsEdit fields2 = fields as IFieldsEdit;
⑭  IFieldEdit field2 = new Field() as IFieldEdit;
⑮  field2.Name_2 = "Name";
⑯  field2.Type_2 = esriFieldType.esriFieldTypeString;
⑰  field2.Length_2 = 20;
⑱  fields2.AddField(field2);
⑲  //生成有效的字段集
⑳  IFieldChecker fieldChecker = new FieldCheckerClass();
21  IEnumFieldError enumFieldError = null;
22  IFields validatedFields = null;
23  fieldChecker.ValidateWorkspace = (IWorkspace)fws;
24  fieldChecker.Validate(fields, out enumFieldError, out validatedFields);
25  //创建要素类
26  IFeatureClass targetFClass = fws.CreateFeatureClass(shpName,
27        validatedFields,
28        null, null, esriFeatureType.esriFTSimple, fcDesc.ShapeFieldName, "");
```

6.4 FeatureClass管理

(2)使用IFeatureClass中的方法管理字段、要素等

- AddField : 添加字段
- DeleteField : 删除字段
- FeatureCount : 获取要素数量
- FeatureType : 要素类型
- Fields : 要素类全部字段集合
- FindField : 查询字段并返回字段所在序号
- GetFeature : 获取根据OID获取要素
- GetFeatures : 获取所有要素并返回要素游标IFeatureCursor
- CreateFeature : 创建要素
- CreateFeatureBuffer : 创建要素缓冲区
- Insert : 返回可以创建要素的游标IFeatureCursor
- Search : 返回满足查询条件的要素游标
- Select : 选择要素
- ShapeFieldName : 几何图形字段名称
- ShapeType : 几何图形类型
- Update : 返回满足条件的更新游标

6.4 FeatureClass管理

[示例]为已有要素类添加字段

```
① IWorkspaceFactory wf = new ShapefileWorkspaceFactory();
② IFeatureWorkspace fws = wf.OpenFromFile(folder, 0) as IFeatureWorkspace;
③ IFeatureClass featureClass = fws.OpenFeatureClass(shpName);

④ //添加字段
⑤ IFieldEdit field = new Field() as IFieldEdit;
⑥ field.Name_2 = "Num";
⑦ field.Type_2 = esriFieldType.esriFieldTypeInteger;
⑧ featureClass.AddField(field);

⑨ MessageBox.Show(" 字段添加成功!", "提示");
```

6.5 要素管理

- (1)生成单个要素

① IWorkspaceFactory wf = new ShapefileWorkspaceFactory();
② IFeatureWorkspace fws = wf.OpenFromFile(folder, 0) as IFeatureWorkspace;
③ IFeatureClass featureClass = fws.OpenFeatureClass(shpName);
④ IFeature feature = featureClass.CreateFeature();
⑤ //生成点
⑥ IPoint point = new ESRI.ArcGIS.Geometry.Point();
⑦ point.X = 80;　　point.Y = 80;
⑧ feature.Shape = point;
⑨ feature.Value[featureClass.FindField("Name")] = string.Format("Point_{0}",featureClass.FeatureCount(null));
⑩ feature.Store();

(2)批量添加要素

```
① IFeatureClass featureClass = fws.OpenFeatureClass(shpName);
② // Create the feature buffer.
③  IFeatureBuffer featureBuffer = featureClass.CreateFeatureBuffer();
④ // Create insert feature cursor using buffering.
⑤ IFeatureCursor featureCursor = featureClass.Insert(true);
⑥ int index = featureClass.FindField("Name");
⑦ Random rand = new Random();
⑧ for (int i=0;i<500;i++){
⑨     IPoint point = new PointClass();
⑩     point.PutCoords(rand.NextDouble() * 100, rand.NextDouble() * 100);
⑪     featureBuffer.Shape = point;
⑫     featureBuffer.Value[index] = "Point_"+rand.Next(1000,9999).ToString();
⑬     featureCursor.InsertFeature(featureBuffer);
⑭ }
⑮ // Attempt to flush the buffer
⑯ featureCursor.Flush();
```

讨论：如果改用IFeatureClass.CreateFure方法批量添加要素，和使用游标批量添加有什么区别?

6.5 要素管理

(3)删除要素

```
① IQueryFilter filter = new QueryFilter();
② filter.WhereClause = "FID>400";
③ ITable table = (ITable)featureClass;
④ table.DeleteSearchedRows(filter); //如果filter=null，则删除全部要素
```

(4)删除要素方法2

```
① IQueryFilter queryFilter = new QueryFilterClass();
② queryFilter.WhereClause = "FID>400";
③ IFeatureCursor updateCursor = featureClass.Update(queryFilter, false);
④ IFeature feature = updateCursor.NextFeature();
⑤ while (feature != null) {
⑥     updateCursor.DeleteFeature(feature);
⑦     feature = updateCursor.NextFeature();
⑧ }
```

(5)批量更新要素

```
① IQueryFilter filter = new QueryFilter();
② filter.WhereClause = "FID>400";
③ //利用FeatureCursor进行数据更新
④ IFeatureCursor updateCursor = featureClass.Update(queryFilter, false);
⑤ IFeature feature = updateCursor.NextFeature();

⑥ while (feature != null)
⑦ {
⑧     feature.Value[2] ="X_"+feature.Value[0].ToString( );
⑨     updateCursor.UpdateFeature(feature);
⑩     feature = updateCursor.NextFeature();
⑪ }
```

6.6 综合实例

- 检查建筑物是否存在相交?

```
① IFeatureClass fc = this.OpenFeatureClass();
② if (fc == null)              return;
③ IFeature f1,f2;
④ IFeatureCursor cur1 = fc.Search(null, true);
⑤ while ((f1 = cur1.NextFeature()) != null)
⑥ {
⑦       ISpatialFilter filter = new SpatialFilter();
⑧       filter.WhereClause = "FID<>" + f1.OID.ToString();
⑨       filter.SpatialRel = esriSpatialRelEnum.esriSpatialRelIntersects;
⑩       filter.Geometry = f1.Shape as IGeometry;
⑪       ISelectionSet set =  fc.Select(filter, esriSelectionType.esriSelectionTypeHybrid,
⑫                                      esriSelectionOption.esriSelectionOptionNormal, null) ;
⑬       if(set.Count>0)  { Debug.WriteLine(string.Format("{0} 有相交!", f1.OID));    }
⑭ }
⑮ MessageBox.Show("检查完成!");
```

本章小结

- GeoDatabase
- 工作区Workspace
- Dataset数据集
- 要素类的管理
- 要素的管理