# shapeFile

## 1.1 shapefile简介

ESRI Shapefile, 是美国环境系统研究所公司（ESRI）开发的一种空间数据开放格式，属于一种矢量图形格式，目前，该文件格式已经成为了地理信息软件界的一个开放标准，用于根据几何体对象：点、折线、多边形，实现描述各类地理信息和空间对象的属性。

## 1.2 shapefile组成

Shapefile一般由多个文件共同组成，同时文件必须放在同一文件夹，一般通过命名相互关联。

Shapefile可选的组成文件格式共有11种，有8种为可选，储存包括空间索引，几何体索引等等信息，另外3种为必须文件，主要功能如下：

1) .shp — 图形格式，用于保存元素的几何实体。

2) .shx — 图形索引格式。几何体位置索引，记录每一个几何体在shp文件之中的位置，能够加快向前或向后搜索一个几何体的效率。

3) .dbf — 属性数据格式，以dBase IV的数据表格式存储每个几何形状的属性数据。

## 1.3 打开shapefile

浏览shapefile,我们一般需要下载ESRI的软件ArcMap，这是ArcGis桌面端三大组件之一，属于付费软件，下载破解和安装可参考下面网站：

下载地址：

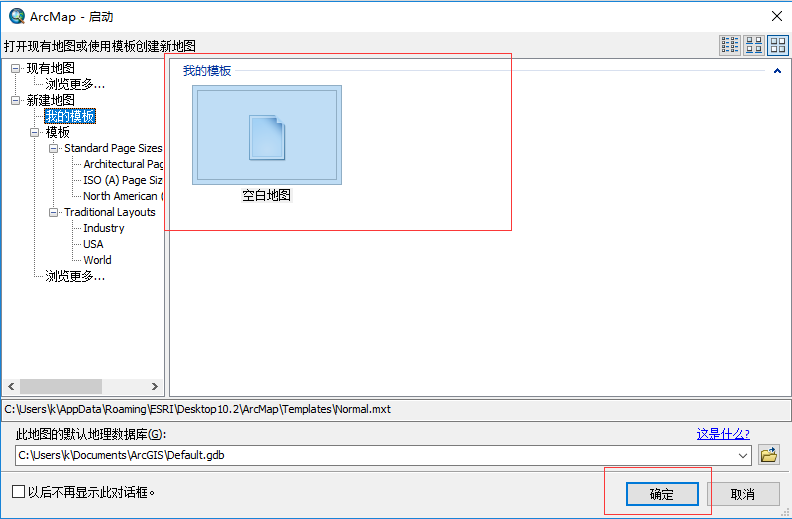
<https://pan.baidu.com/s/1kWWd09L>

密码：hjxy

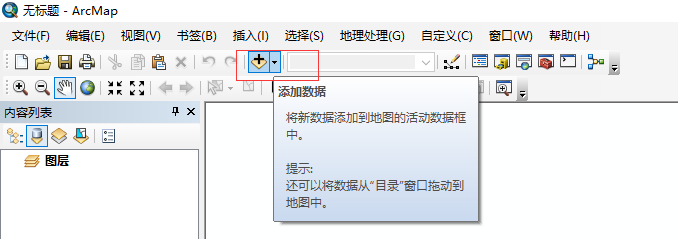
破解教程：

<https://jingyan.baidu.com/article/e73e26c0cb5c1324adb6a791.html>

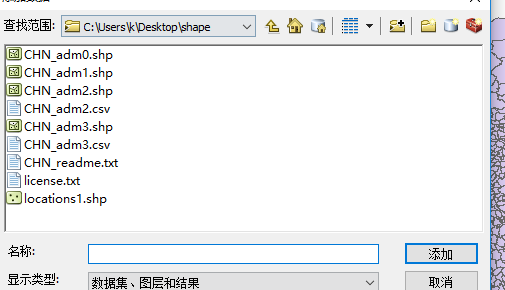
安装完成后，点击arcgis，系统会提醒我们需要先先创建一个arcgis图层，如下图使用最简单的空白模板即可。



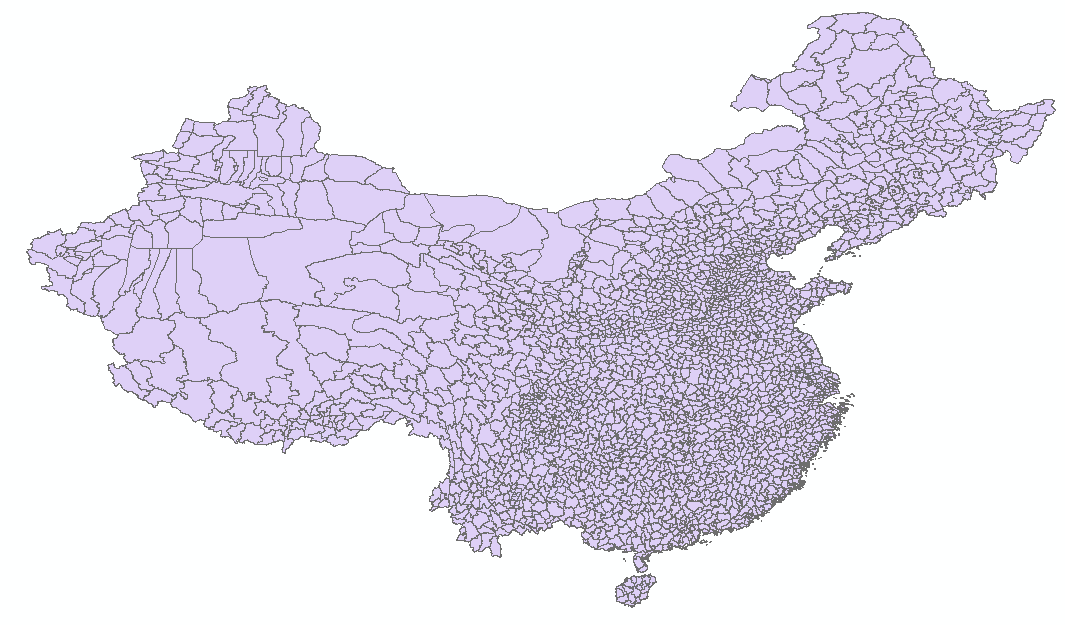
随后，如下图所示点击添加数据



选择需要打开的shapefile主文件.shp

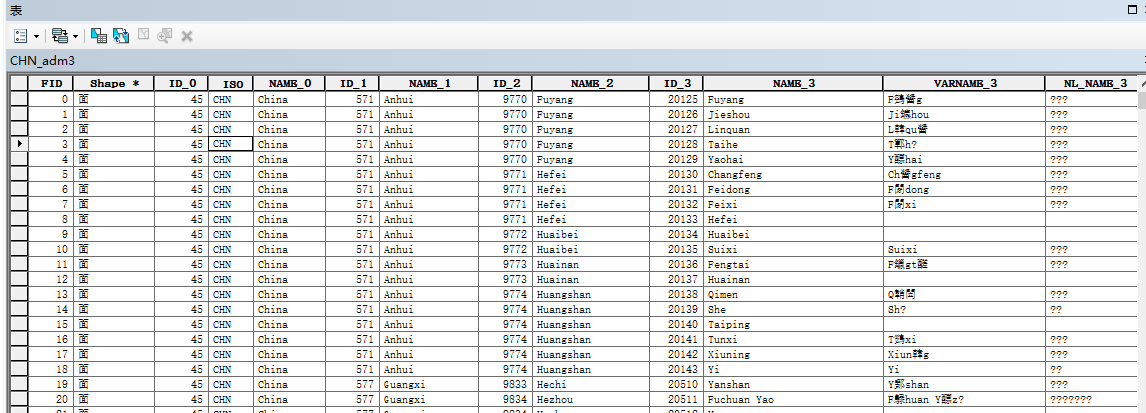


打开shapefile



## 1.4 简单了解shapefile

Shapefile文件通过点、线、面图形化展示地理信息，他可以按比例放大缩小展示。

同时我们在图层右侧选择shape文件，右键打开属性我们可以看到，arcgis将以表格形式把shapefile所包含的信息以表格形式展示出来。

我们的目的是编辑shapfile，其实就相当于我们需要编辑这一张表格，他的字段，数据都是可以编辑的。

# Geotools

2.1 **geotools简介**

编辑shapfile需要如arcGis一样打开shapfile文件，把几个文件的内容属性相互关联起来，进行编辑修改。这是比较繁杂的过程，我们可以把这些过程交给java开源工具geotools，我们只需要调用接口进行编辑即可。

Geotools是从1999年开始编写，至今保持更新，相对的，他的功能较为繁杂，包含多个jar包，项目也较大，需要根据需要对其进行精简。

更多关于geotools的资料可以到官网：<http://geotools.org> 进行了解，同时geotools项目也将在官网进行下载。

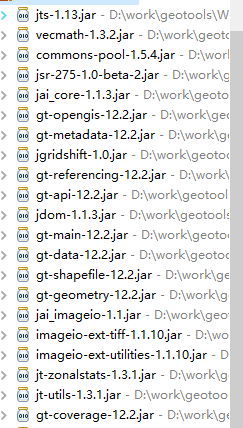
# Geotools读写shapefile

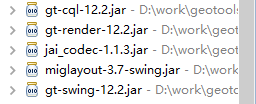
## 3.1 geotools jar包筛选

在官网我们可以下载到geotools整个项目和geotools的各个jar包及文档，考虑我们项目使用java7和稳定性，我们选用v12版本。

我们可以看到geotools已经发展了数十年，他的jar包数量非常庞大，有上百个之多，针对项目内所使用的书写功能，我们需要对官网下载的bin包内的jar包进行筛选。

筛选后jar包引用如下图：





## 读取shapefile内容

代码如下：

**import** java.io.File;

**import** java.nio.charset.Charset;

**import** java.util.Iterator;

**import** java.util.Map;

**import** org.geotools.data.shapefile.ShapefileDataStore;

**import** org.geotools.data.shapefile.ShapefileDataStoreFactory;

**import** org.geotools.data.simple.SimpleFeatureIterator;

**import** org.geotools.data.simple.SimpleFeatureSource;

**import** org.opengis.feature.Property;

**import** org.opengis.feature.simple.SimpleFeature;

**import** org.opengis.feature.type.Name;

**import** org.opengis.feature.type.PropertyDescriptor;

**import** org.opengis.feature.type.PropertyType;

**public** **class** ReadShape {

@SuppressWarnings("unused")

**public** **static** **void** main(String[] args) {

ShapefileDataStoreFactory dataStoreFactory = **new** ShapefileDataStoreFactory();

**try** {

ShapefileDataStore sds = (ShapefileDataStore)dataStoreFactory.createDataStore(**new** File("C:\\Users\\k\\Desktop\\shape\\CHN\_adm2.shp").toURI().toURL());

sds.setCharset(Charset.*forName*("GBK"));

SimpleFeatureSource featureSource = sds.getFeatureSource();

SimpleFeatureIterator itertor = featureSource.getFeatures().features();

**while**(itertor.hasNext()) {

SimpleFeature feature = itertor.next();

Iterator<Property> it = feature.getProperties().iterator();

**while**(it.hasNext()) {

Property pro = it.next();

PropertyType type = pro.getType();

Name name = pro.getName();

PropertyDescriptor descriptor = pro.getDescriptor();

Map<Object, Object> userData = pro.getUserData();

Object value = pro.getValue();

}

}

itertor.close();

} **catch** (Exception e) {

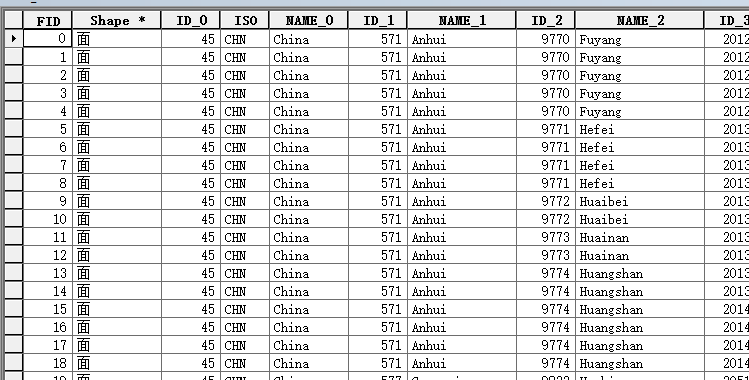
e.printStackTrace();

}

}

}

根据上述代码，参考之前提到的arcgis的属性框（如下图所示）



Geotools读取shapefile是一列一列进行遍历的，从shape开始往后遍历，那么我们只要取出数据就讲可以进行加工。

## 写入shapefile内容

写入shapefile内容，官方推荐是以csv格式文件进行写入，这在官网<http://docs.geotools.org/latest/userguide/tutorial/feature/csv2shp.html> 有较为详尽的介绍。同时，还可以通过下列直接写入固定数据的形式写入内容。由于geotools发展历程长远，版本过多，下述实现只能在v12后的版本可以运行。

Shapefile写入点：

**import** java.io.File;

**import** java.io.IOException;

**import** java.io.Serializable;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** org.geotools.data.DefaultTransaction;

**import** org.geotools.data.Transaction;

**import** org.geotools.data.collection.ListFeatureCollection;

**import** org.geotools.data.shapefile.ShapefileDataStore;

**import** org.geotools.data.shapefile.ShapefileDataStoreFactory;

**import** org.geotools.data.simple.SimpleFeatureCollection;

**import** org.geotools.data.simple.SimpleFeatureSource;

**import** org.geotools.data.simple.SimpleFeatureStore;

**import** org.geotools.feature.simple.SimpleFeatureBuilder;

**import** org.geotools.feature.simple.SimpleFeatureTypeBuilder;

**import** org.geotools.geometry.jts.JTSFactoryFinder;

**import** org.geotools.referencing.crs.DefaultGeographicCRS;

**import** org.geotools.swing.data.JFileDataStoreChooser;

**import** org.opengis.feature.simple.SimpleFeature;

**import** org.opengis.feature.simple.SimpleFeatureType;

**import** com.vividsolutions.jts.geom.Coordinate;

**import** com.vividsolutions.jts.geom.GeometryFactory;

**import** com.vividsolutions.jts.geom.Point;

**public** **class** SimpleWrite {

**public** **static** **void** main(String[] args) **throws** IOException {

SimpleFeatureTypeBuilder builder = **new** SimpleFeatureTypeBuilder();

builder.setName("Location");

builder.setCRS(DefaultGeographicCRS.*WGS84*); // <-坐标参考系统

// 往订单中填加属性

builder.add("the\_geom", Point.**class**);

builder.length(15).add("Name", String.**class**); // <- 15名称字段

builder.add("number",Integer.**class**);

// 构建类型

**final** SimpleFeatureType TYPE = builder.buildFeatureType();

List<SimpleFeature> features = **new** ArrayList<>();

/\*

\* GeometryFactory将用于创建每个要素的几何属性，

\*使用Point对象作为位置。

\*/

GeometryFactory geometryFactory = JTSFactoryFinder.*getGeometryFactory*();

SimpleFeatureBuilder featureBuilder = **new** SimpleFeatureBuilder(TYPE);

Point point = geometryFactory.createPoint(**new** Coordinate(116.46, 39.92));

featureBuilder.add(point);

featureBuilder.add("test");

featureBuilder.add(22);

SimpleFeature feature = featureBuilder.buildFeature(**null**);

features.add(feature);

/\*

\*获取输出文件名并创建新的shapefile

\*/

String path = "D://test//";

String newPath =path+ "11.shp";

JFileDataStoreChooser chooser = **new** JFileDataStoreChooser("shp");

chooser.setDialogTitle("Save shapefile");

chooser.setSelectedFile(**new** File(newPath));

File newFile = chooser.getSelectedFile();

ShapefileDataStoreFactory dataStoreFactory = **new** ShapefileDataStoreFactory();

Map<String, Serializable> params = **new** HashMap<>();

params.put("url", newFile.toURI().toURL());

params.put("create spatial index", Boolean.*TRUE*);

ShapefileDataStore newDataStore = (ShapefileDataStore) dataStoreFactory.createNewDataStore(params);

/\*

\* TYPE用作模板来描述文件内容

\*/

newDataStore.createSchema(TYPE);

/\*

\* 将特征写入shapefile

\*/

Transaction transaction = **new** DefaultTransaction("create");

String typeName = newDataStore.getTypeNames()[0];

SimpleFeatureSource featureSource = newDataStore.getFeatureSource(typeName);

SimpleFeatureType SHAPE\_TYPE = featureSource.getSchema();

/\*

\* Shapefile格式有一些限制：

\* - “the\_geom”始终是第一个，用于几何属性名称

\* - “the\_geom”必须是Point，MultiPoint，MuiltiLineString，MultiPolygon类型

\* - 属性名称的长度是有限的

\* - 不支持所有数据类型Timestamp表示为Date）

\*

\*每个数据存储都有不同的限制，因此请检查生成的SimpleFeatureType。

\*/

System.*out*.println("SHAPE:"+SHAPE\_TYPE);

**if** (featureSource **instanceof** SimpleFeatureStore) {

SimpleFeatureStore featureStore = (SimpleFeatureStore) featureSource;

/\*

\* SimpleFeatureStore有一个方法来添加来自

\* SimpleFeatureCollection对象的特征，所以我们使用ListFeatureCollection

\*类来包装我们的特征列表。

\*/

SimpleFeatureCollection collection = **new** ListFeatureCollection(TYPE, features);

featureStore.setTransaction(transaction);

**try** {

featureStore.addFeatures(collection);

transaction.commit();

} **catch** (Exception problem) {

problem.printStackTrace();

transaction.rollback();

} **finally** {

transaction.close();

}

}

}

}

# 4. CAD

CAD原指辅助设计人员进行图形设计的软件，由于AutoCAD的普及，现在通常指autoCAD软件生成的文件。这类文件包含三种格式：

（1）DWG：是原始图纸格式，包含了图纸所有的信息，Autodesk公司出于安全考虑没有给出详细的格式说明。

（2）DWF：比DWG文件小很多，用于其他用户浏览，添加备注等，不能编辑。

（3）DXF：是用于和其他CAD系统交换数据的文件格式，分二进制和ASCII格式，AutoCAD的帮助里包含了DXF文件的详细描述。