

Android组件化开发技术

北京理工大学计算机学院
金旭亮

模块 (Module) 是构建Android应用的主要构造块，它封装了相应的功能代码与相关资源。

Android Application Modules

- 源码、资源文件、配置文件
- 打包为.apk文件

Test Modules

- 包容对JUnit单元测试框架的支持

Library Modules

- 包容源码和资源文件
- 主要用于重用，将被打包到.apk文件中

App Engine Modules

- 用于与Google的服务集成
- 能主动地向手机“推送”信息

Android Library Module的基础知识

无法生成可安装的.apk文件，无法生成Jar包，但可以导入其他的Jar包或Library Module

如果Application Module与Library Module中的资源ID同名，那么，前者优先，后者将会被屏蔽。可以使用“前缀”等方式，保证各模块中的资源ID不同名。

Android Library Module的基础知识-2

不支持raw assets，所用到的所有资源，都必须放到res文件夹中。

所使用的Android SDK版本，以及Java版本，必须与Application Module中的一致或更低。

模块类型转换

Application Module

```
apply plugin: 'com.android.application'  
  
android {...}
```

Library Module

```
apply plugin: 'com.android.library'  
  
android {...}
```

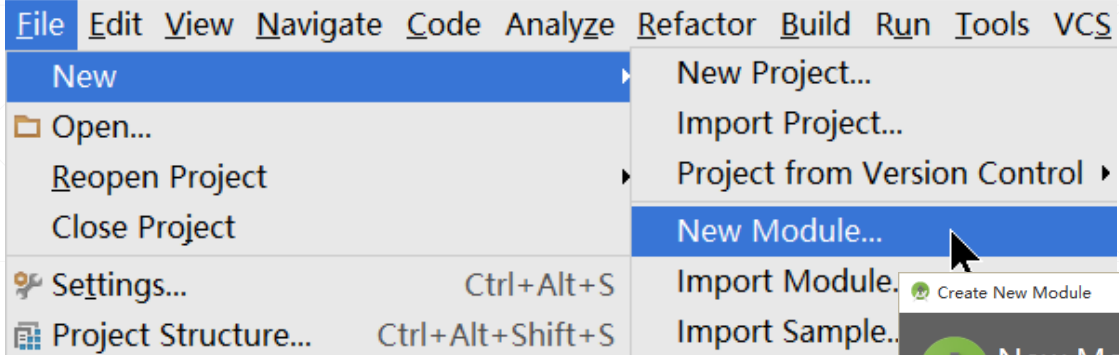
仅需修改build.gradle，就可以很方便地把一个模块从“Application”转换为“Library”

模块中的清单文件

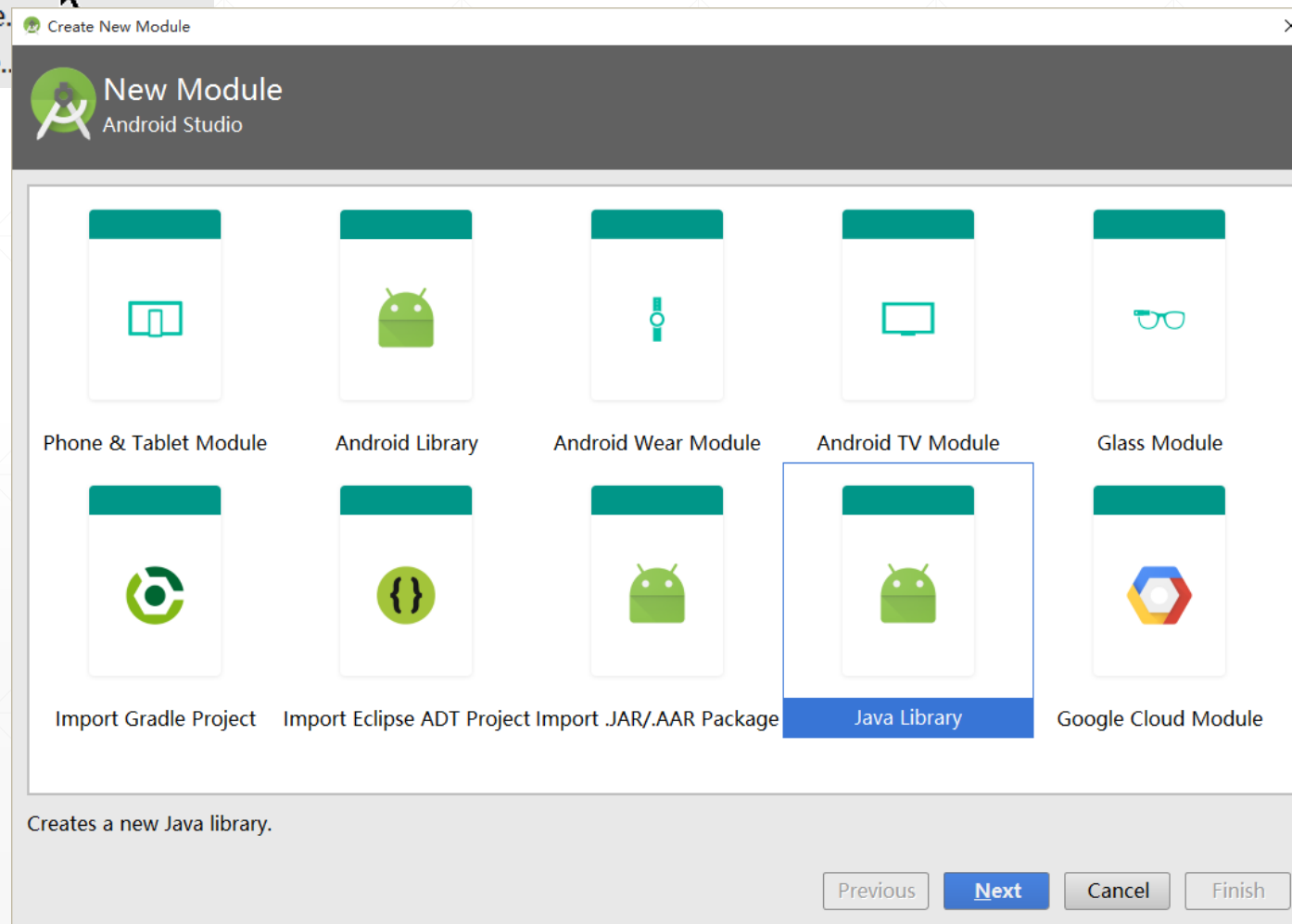
如果Android Library中定义了Activity、Service、Receiver、Provider等组件，那么，在Application Module中也必须在manifest.xml中添加相应的声明

```
<manifest>
...
<application>
...
  <activity android:name="com.example.android.mylibrary.MyLibraryActivity" />
...
</application>
</manifest>
```

使用Java Library Module





Java Library Module适合于封装那些没有可视化界面，不调用Android相关功能的“纯”Java代码。




Create New Module

X

Customize the Activity





Java Library

Creates a new Java library.

Library name:

javali

Java package name:

com.example

[Edit](#)

Java class name:

MyJavaClass

☒ Create .gitignore file

Previous

Next

Cancel

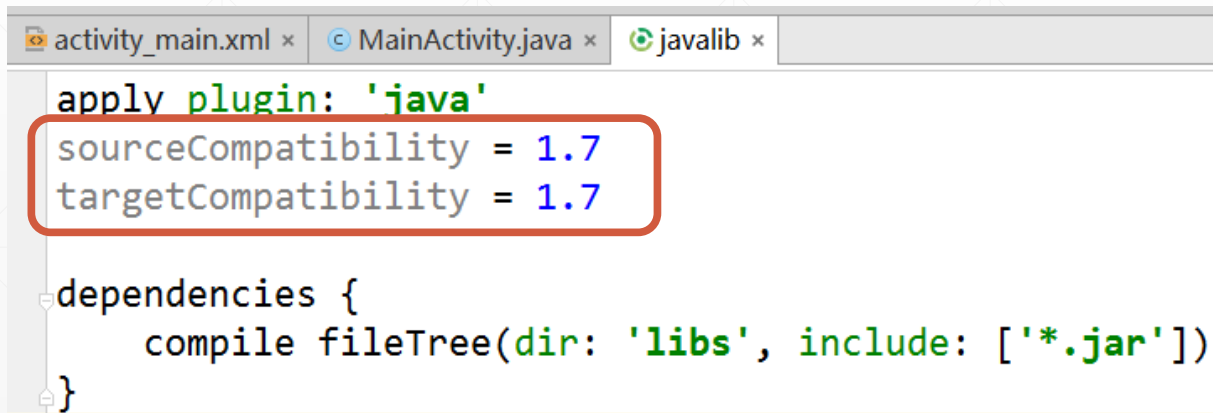
Finish

一个“坑”，注意别掉里头

生成项目Gradle时的报错信息

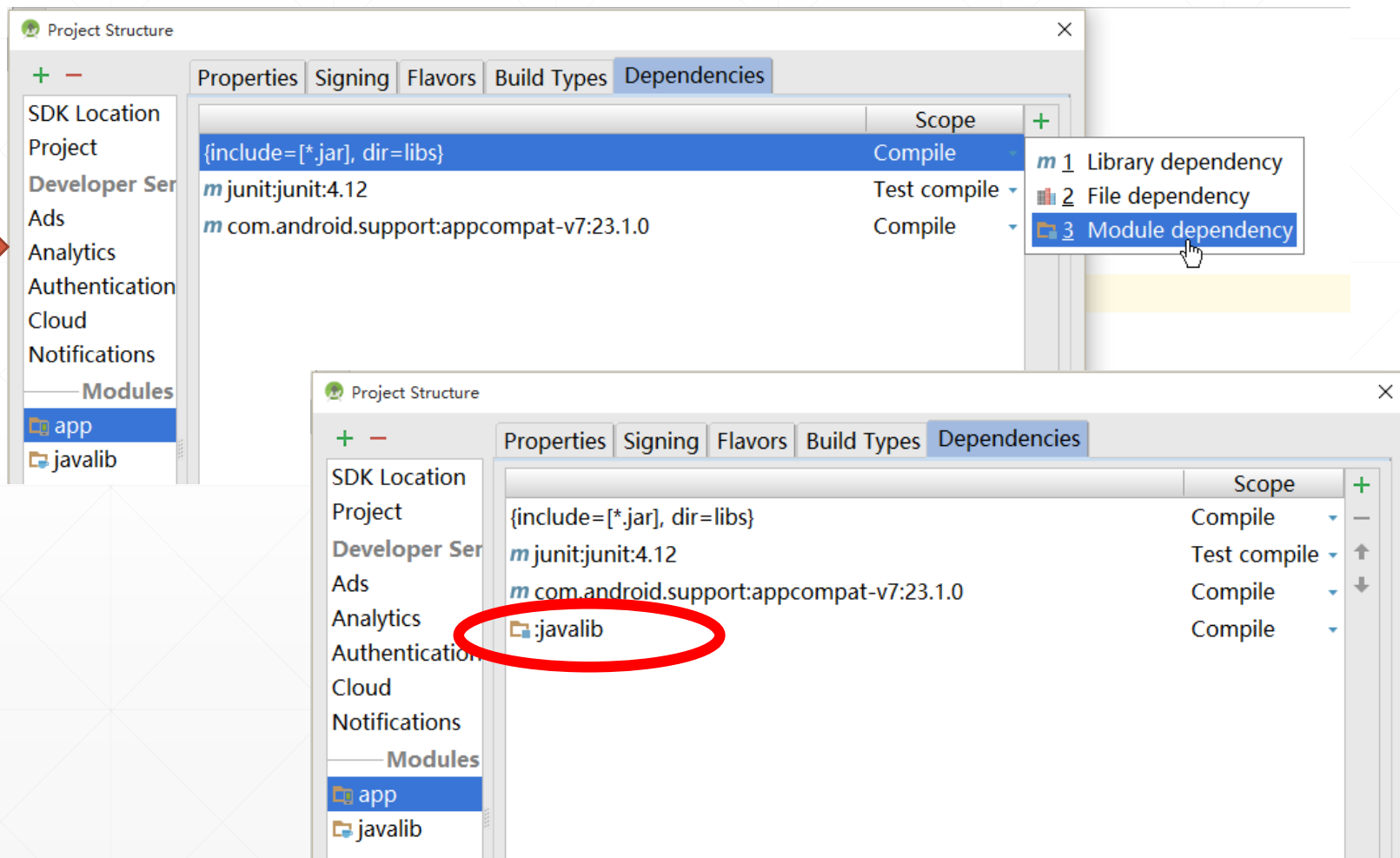
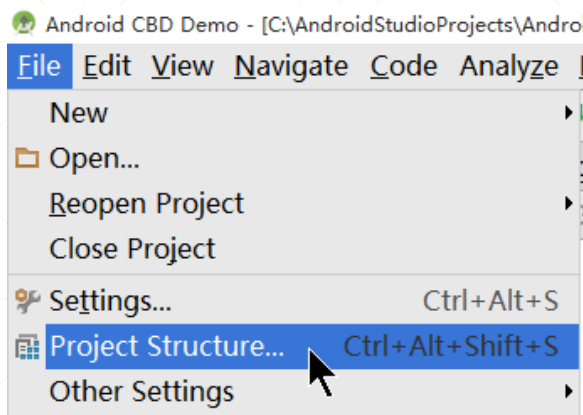
```
.....  
Caused by: com.android.dx.cf.iface.ParseException: bad class  
file magic (cafebabe) or version (0034.0000)  
.....
```

由于目前Android还不支持Java 8，因此，在编译Java Library Module时，必须指定使用Java 7，否则报告错误。



```
activity_main.xml x MainActivity.java x javalib x  
apply plugin: 'java'  
sourceCompatibility = 1.7  
targetCompatibility = 1.7  
  
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
}
```

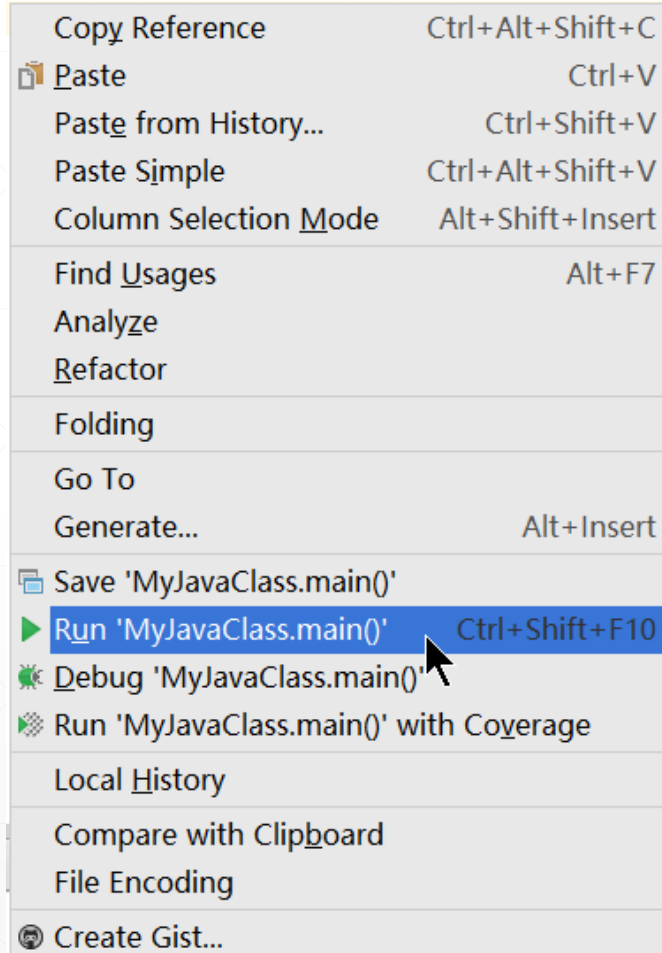
给主模块添加对Java Library Module的引用



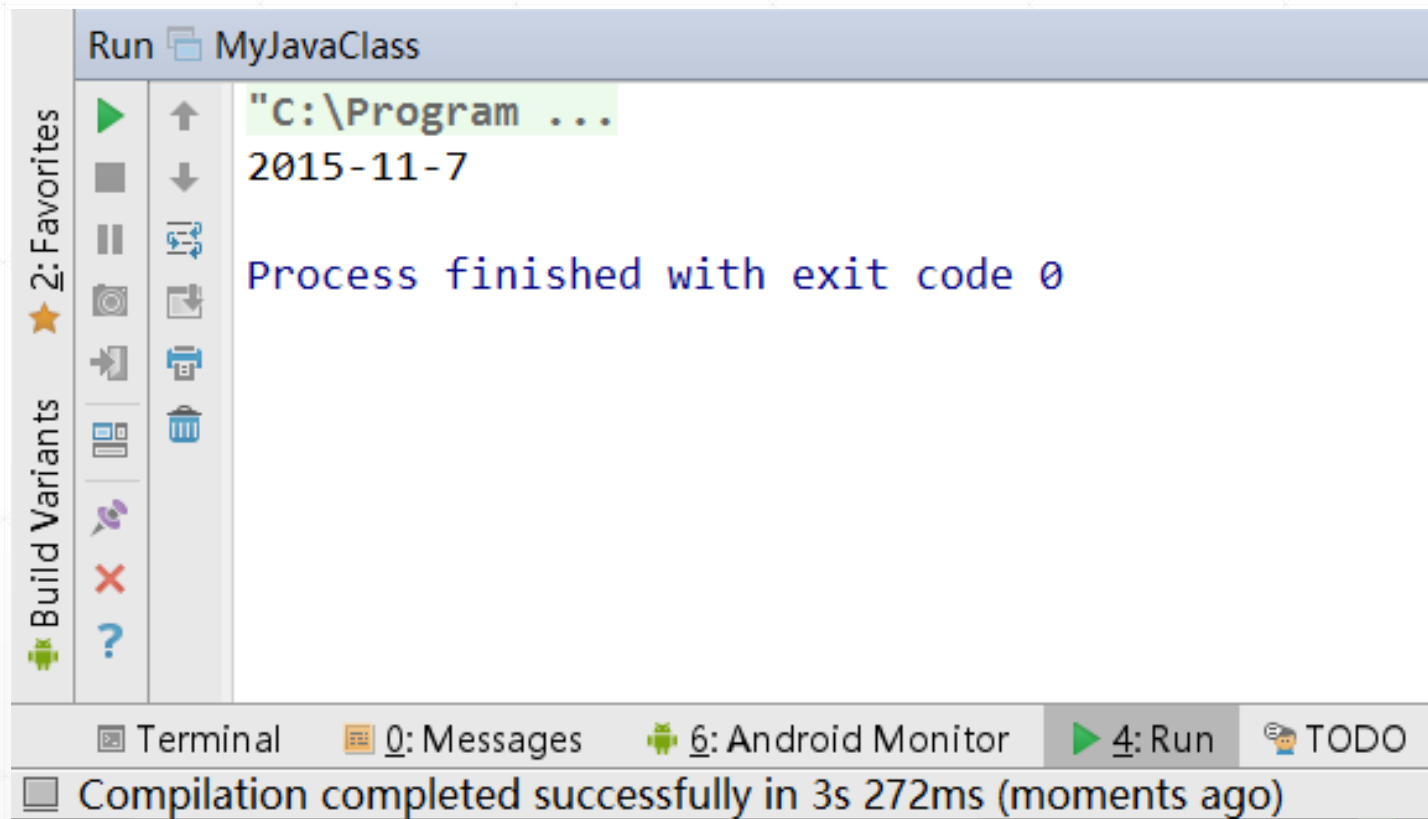
给Java Library Module编写可重用的代码

```
public class MyJavaClass {  
  
    /**...*/  
    public static String getNowDateString(){  
        String dateTimeString= DateFormat  
            .getDateInstance(DateFormat.MEDIUM)  
            .format(new Date());  
        return dateTimeString;  
    }  
  
    public static void main(String[] args){  
        //用于测试静态方法的输出  
        System.out.println(getNowDateString());  
    }  
}
```

可以直接在Java文件中添加main方法，
然后在Android Studio中运行之.....



Android Studio运行Java Library中的main()方法



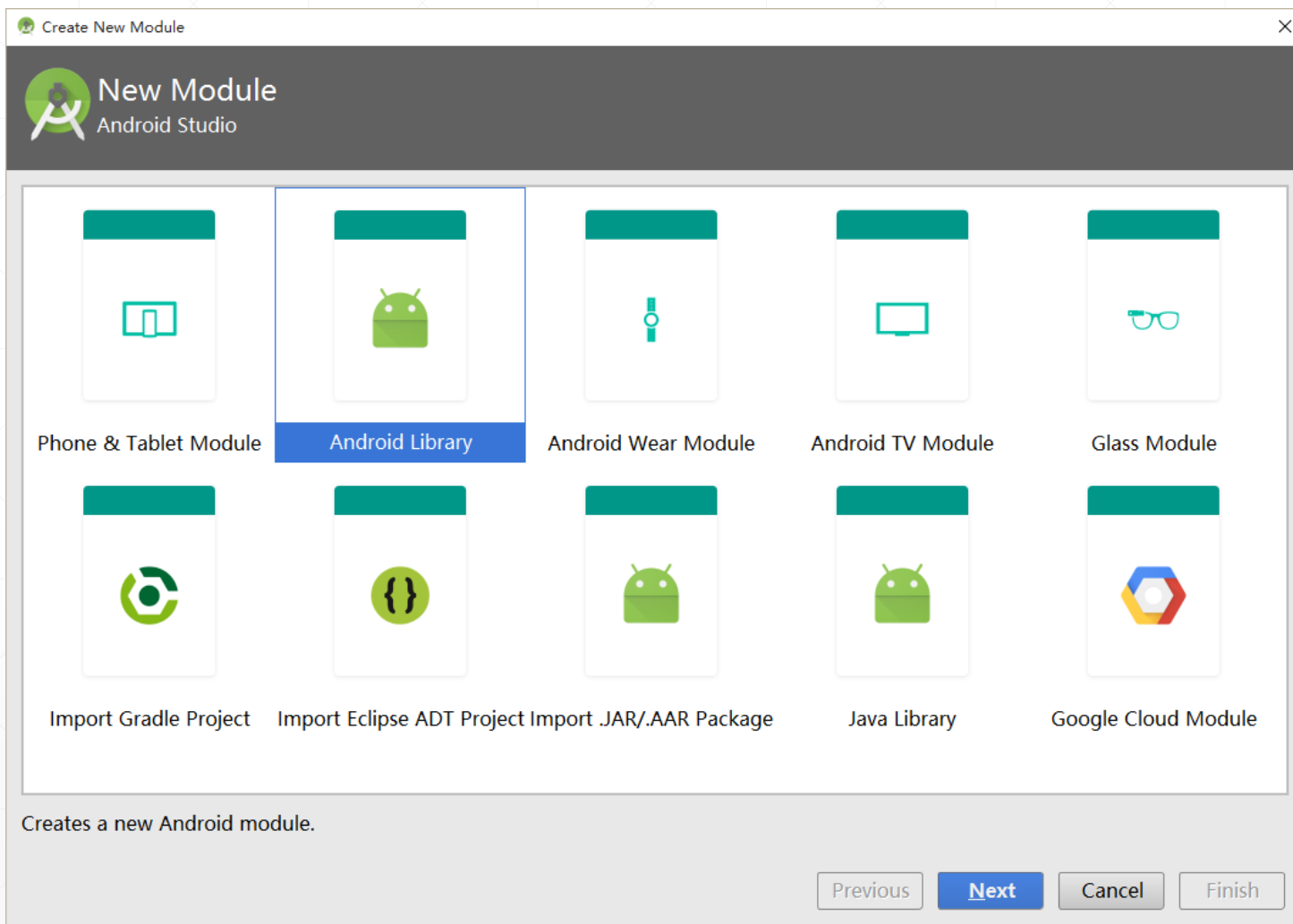
在Android Module中调用Java Library



使用Java Library Module中的
组件实现的功能

示例：AndroidCBDDemo


使用Android Library



当有一些代码需要在多个Android项目中重用时，可以创建Android Library。

Android Library最适合于封装那些需要调用Android SDK中各种功能的组件。

Create New Module

 **Phone & Tablet Module**
Configure your new module

Configure your new module

Application/Library name:

My Android Library

Module name:

myandroidlibrary

Package name:

cn.edu.bit.jinxuliang.myandroidlibrary

[Edit](#)

Minimum SDK

API 19: Android 4.4 (KitKat) ▼

Previous

Next

Cancel

Finish

为Android Library Module编写代码

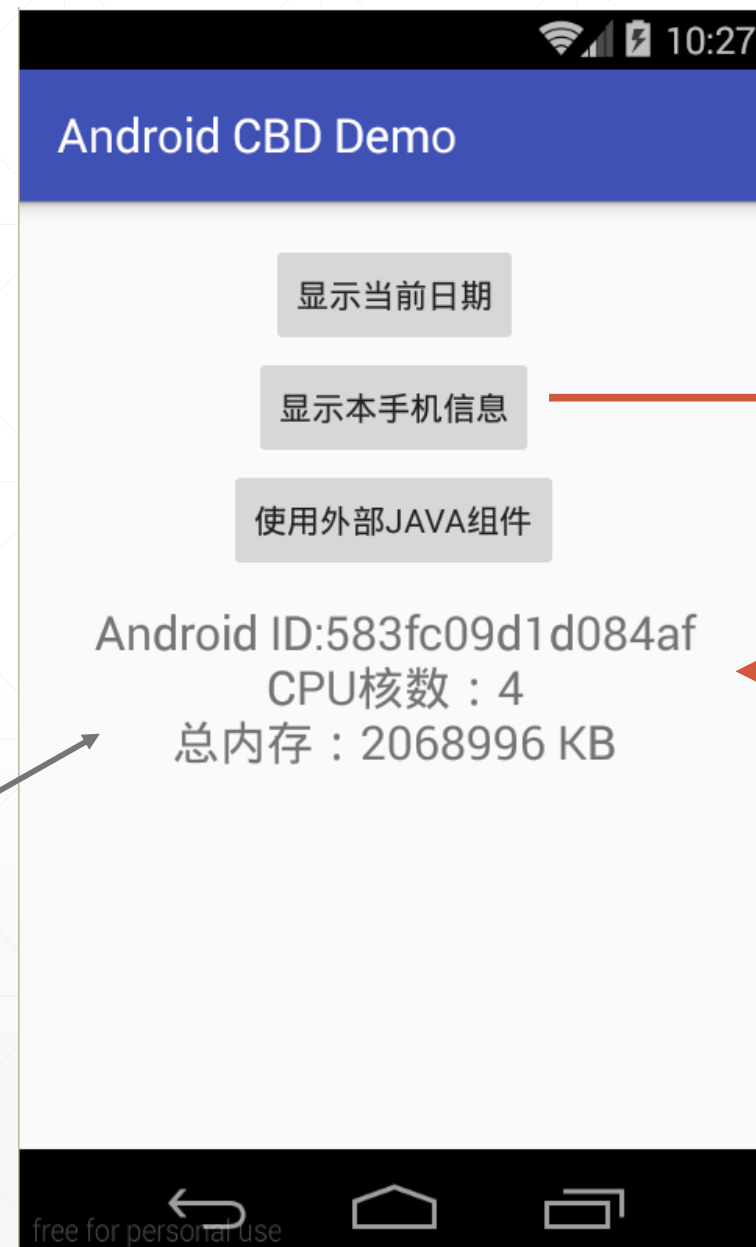
```
public class AndroidSystemHelper {  
    /**...*/  
    public static String getAndroidId(Context context) {...}  
  
    /**...*/  
    public static void installApk(Context context, File apkFile) {...}  
  
    /**...*/  
    public static int getNumCores() {...}  
  
    /**...*/  
    public static long getTotalMemory() {...}  
}
```

在Android Library中，可以直接访问Android SDK.....



在主Module中添加对于
Android Library的依赖引用

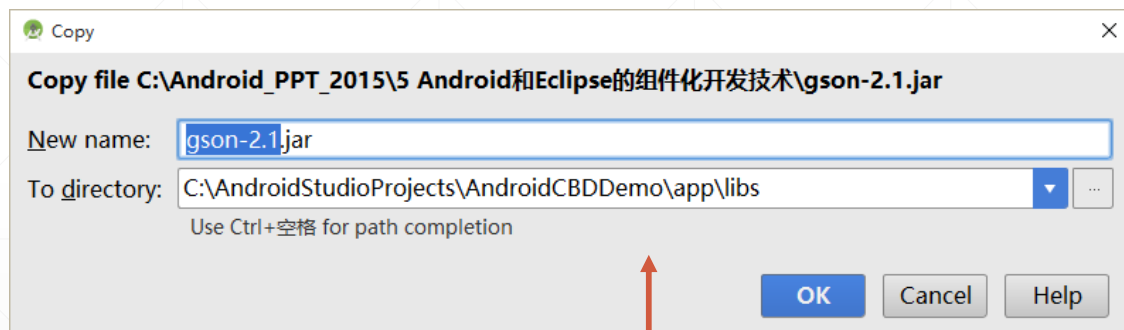
添加引用之后，Android
Library中的代码与本模块中的
代码使用上没有任何区别。



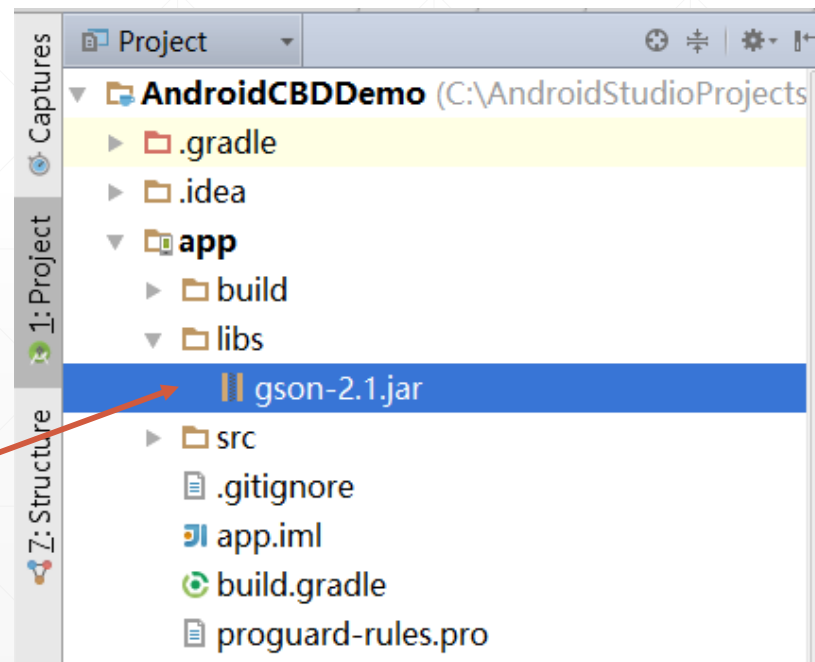
使用外部Jar包

多年以来，Java平台已经积累了丰富的代码资源，许多以Jar包的形式存在。

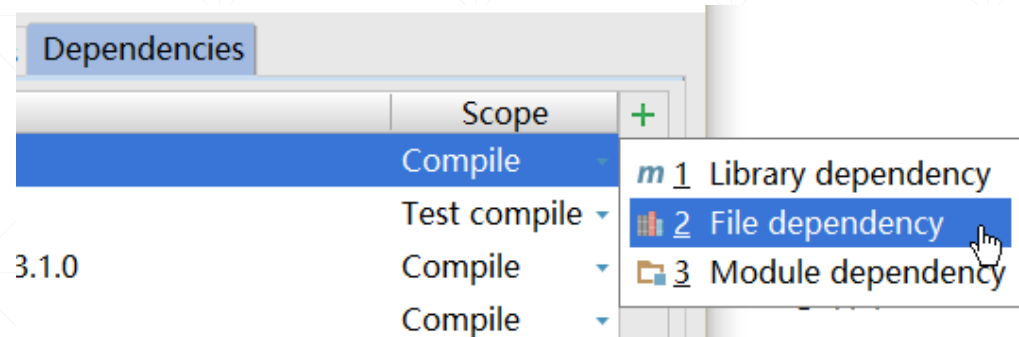
通过把Jar包直接复制到模块的libs文件夹，Android应用能够很方便地重用这些组件。



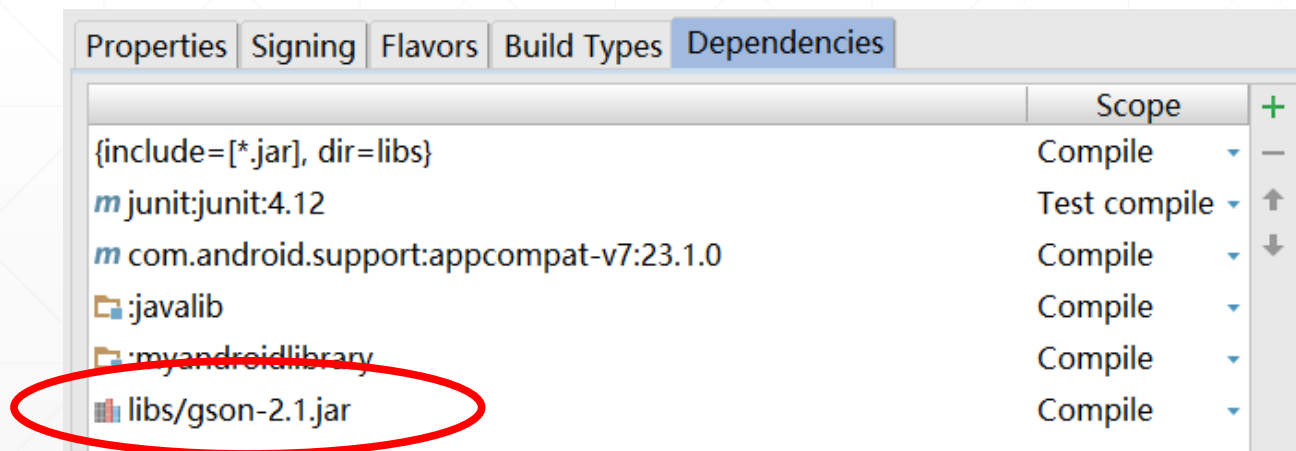
Gson是Google提供的一个序列化和反序列化json格式数据的一个组件。



在Android模块中使用外部Java包

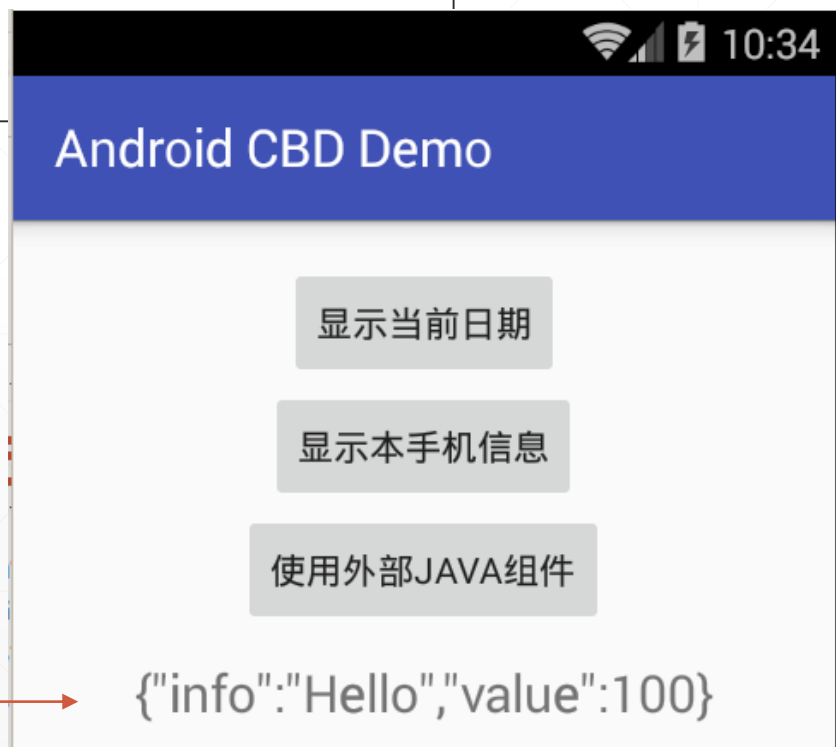


使用添加“文件依赖”的方式
引用已经复制到libs文件夹中的
java包



编写测试代码

```
private void useGson() {  
    Gson gson=new Gson();  
    String json=gson.toJson(new MyClass(100, "Hello"));  
    infoTextView.setText(json);  
}
```

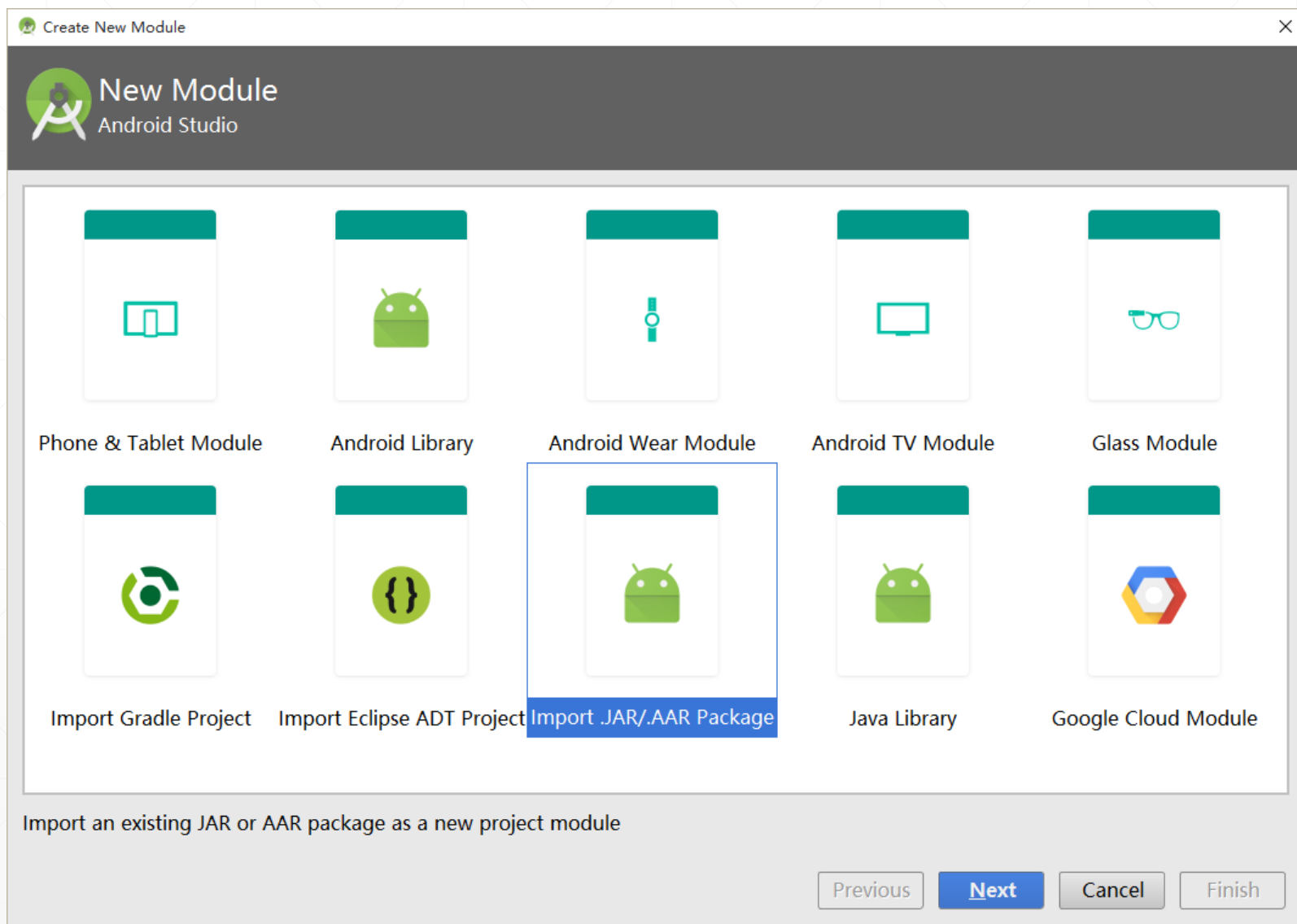


序列化为json的Java对象



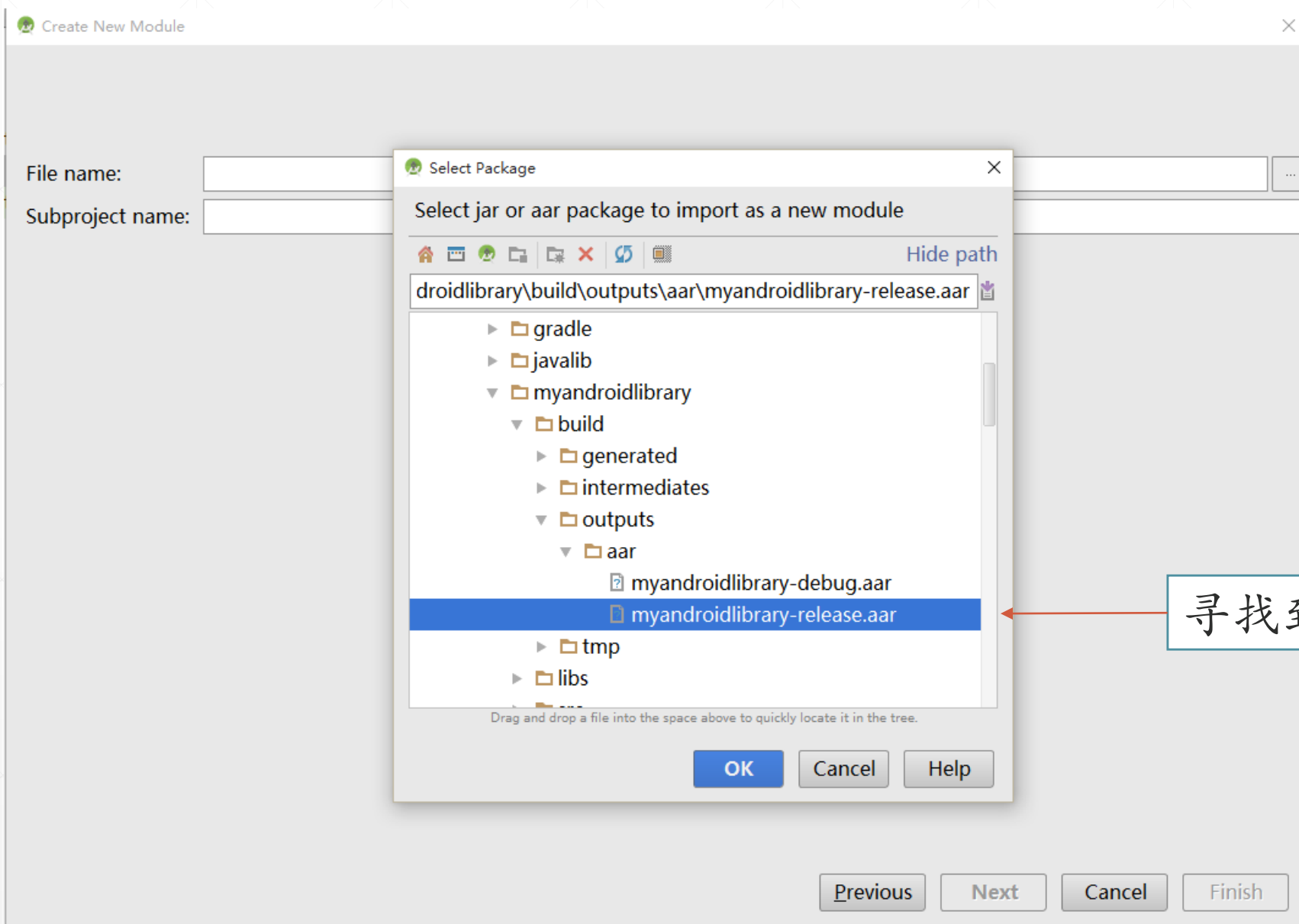
{"info":"Hello","value":100}

导入现有的Library

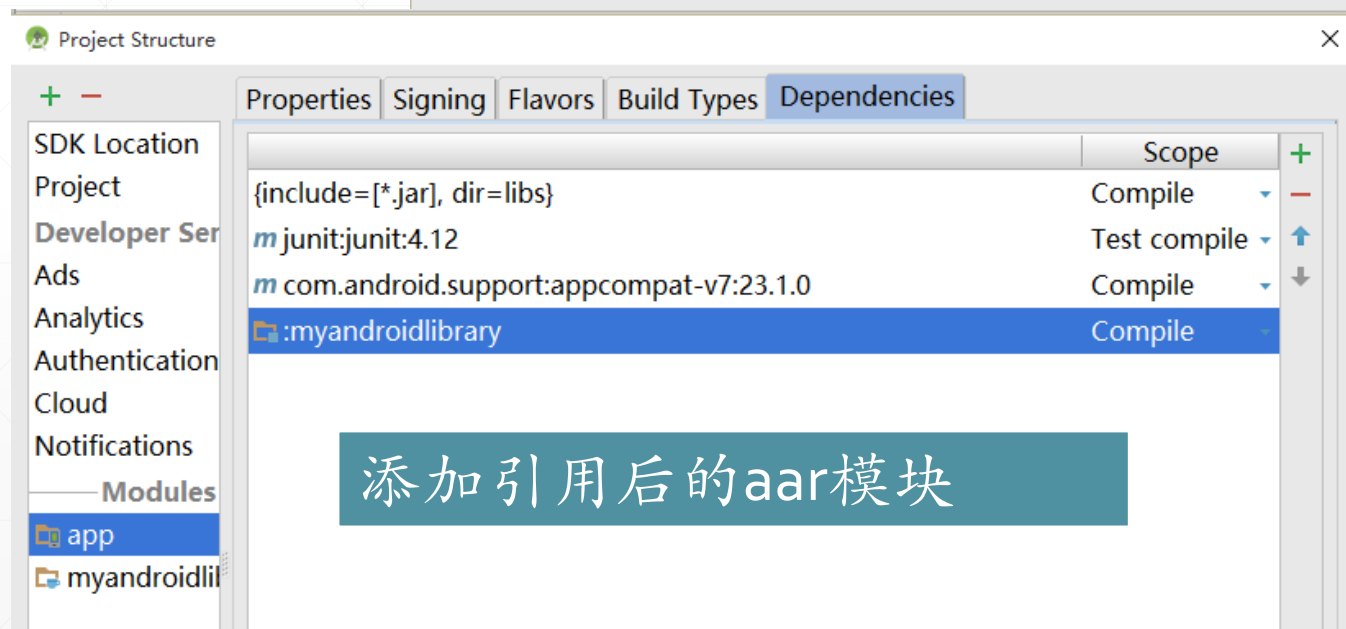
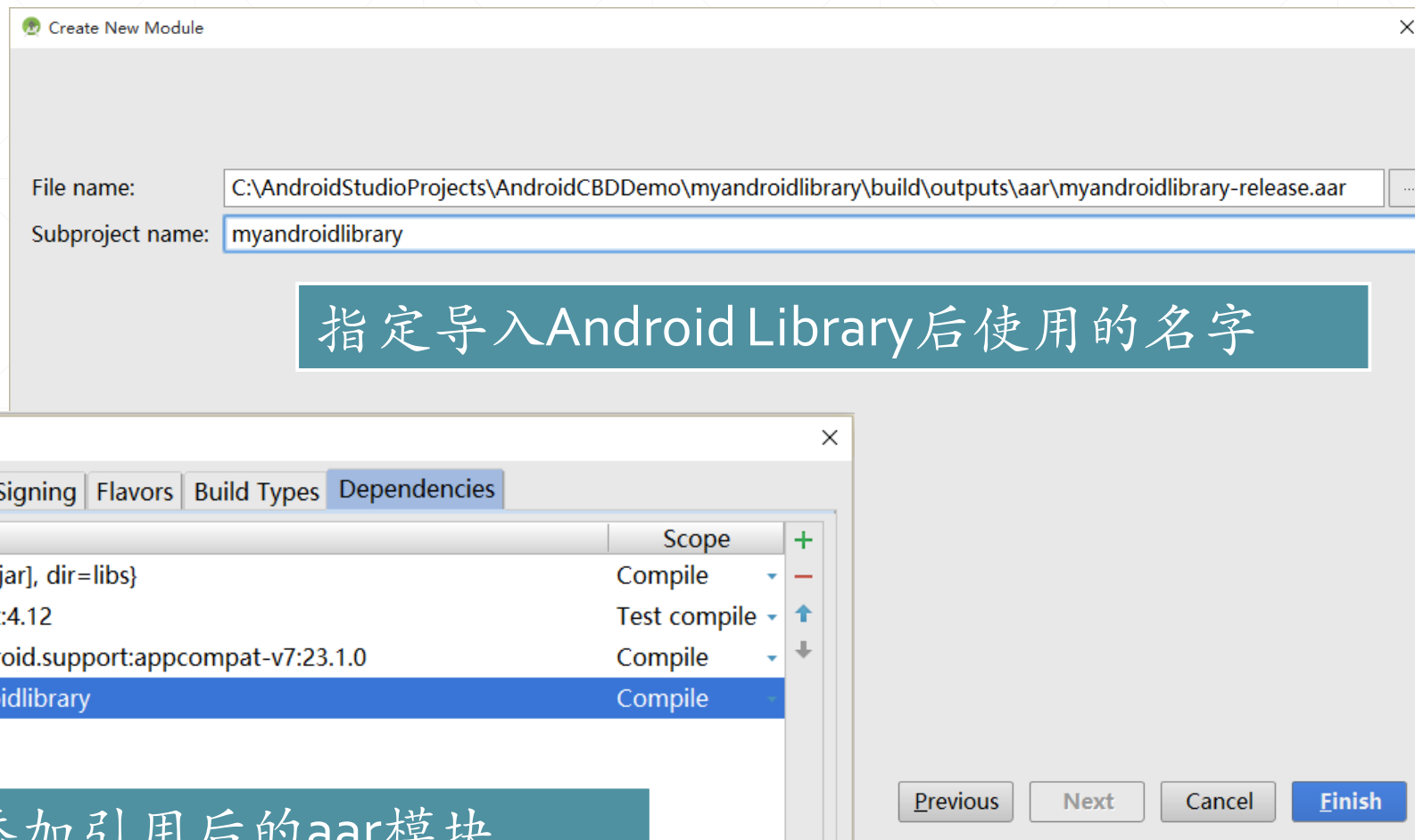


Android Library会生成一个文件名后缀为.aar的可重用组件。

新的Android项目可以直接地导入这个文件，以直接重用此Android Library中的代码。



寻找到生成的aar文件



测试以aar方式导入的Android Library

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    infoTextView = (TextView) findViewById(R.id.tvInfo);
    // 引用Android Library Module中的组件
    int cpuCores = AndroidSystemHelper.getNumCores();
    infoTextView.setText("本机CPU核数: "+cpuCores);
}
```

