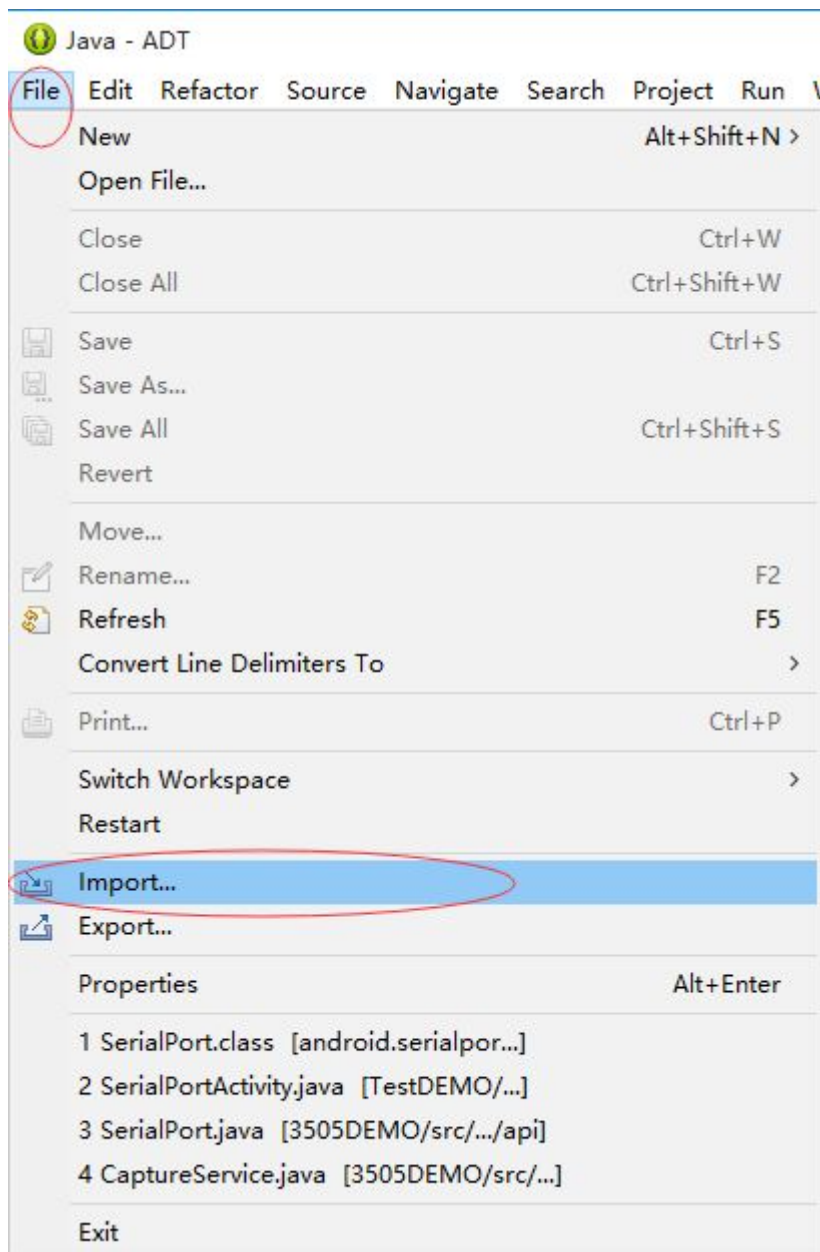


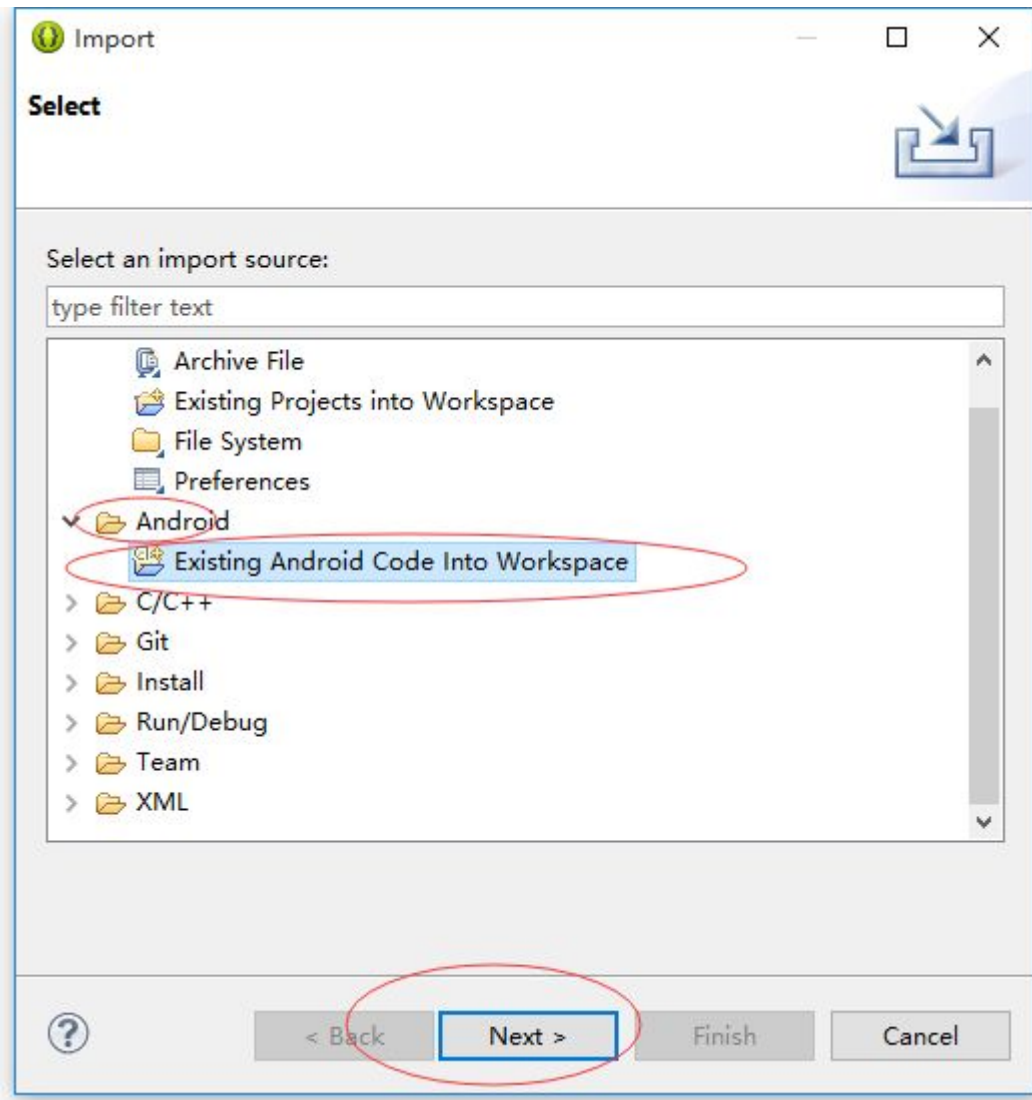
项目说明:

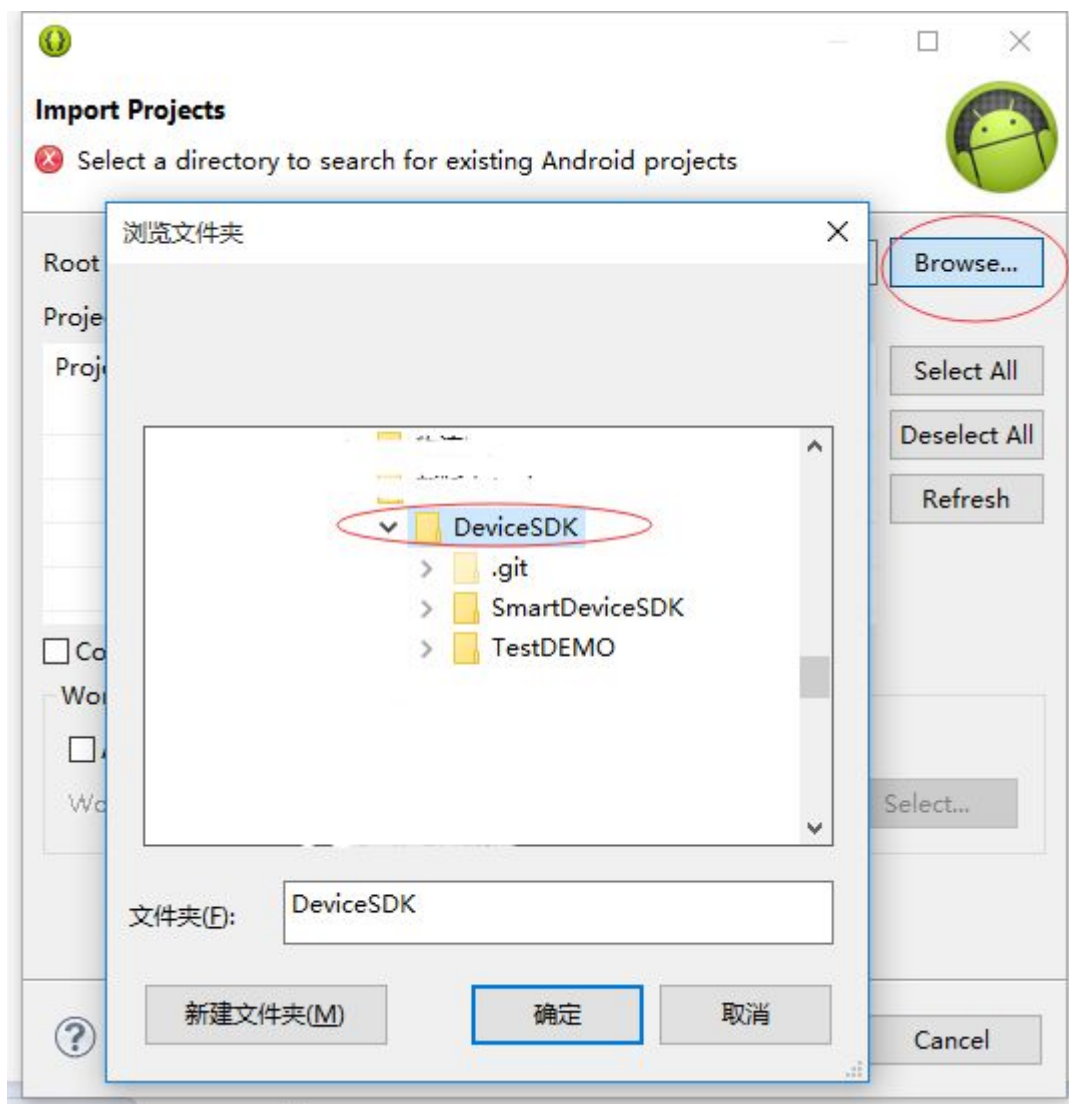
导入选择 DeviceSDK 文件夹，导入两个项目，导入后效果如下

- > > SmartDeviceSDK [DeviceSDK master]
- > > TestDEMO [DeviceSDK master]

Eclipse 导入方式: File->Import->Android->Existing Android Code Into Workspace->Chose 'DeviceSDK'->Finish







SDK 包说明:

android.serialport.api

串口操作接口，主要用到 SerialPort 类

SerialPort 函数说明:

函数	说明	参数	备注
SerialPort(String, int)	实例化，并自动打开串口，启动接收返回数据线程	String 串口设备名称，如 /dev/ttySAC1 int 串口波特率，如 115200	
SerialPort()	实例化		需 要 调 用 open(String, int) 打开串口，并使用

			mFileInputStream 接收返回数据
setOnSerialportDataReceived(SerialPortDataReceived)	注册返回数据接收事件		如果注册该函数，需要使用 <code>SerialPort(String, int)</code> 方法实例化
open(String, int)	打开串口	String 串口设备名称，如 <code>/dev/ttySAC1</code> int 串口波特率，如 <code>115200</code>	
open()	打开串口		
closePort()	关闭串口，释放资源		
Write(byte[])	发送串口数据	16 进制数据	
Write(String)	发送串口数据	字符数据，以 UTF-8 编码发送	
setOnSerialportDataReceived(SerialPortDataReceived)	注册返回数据监听事件		

com.smartdevicesdk.adapter

UI 适配器管理

com.smartdevicesdk.camerascanner

摄像头扫描接口

使用方法：

//需要注册权限

```
<uses-feature android:name="android.hardware.camera" />
```

//打开扫描窗口

```
Intent intent = new Intent(this, ZBarScannerActivity.class);
startActivityForResult(intent, ZBAR_SCANNER_REQUEST);
```

//接收扫描返回数据

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    if (resultCode == RESULT_OK) {
        // Scan result is available by making a call to
        // data.getStringExtra(ZBarConstants.SCAN_RESULT)
        // Type of the scan result is available by making a call to
        // data.getStringExtra(ZBarConstants.SCAN_RESULT_TYPE)
        String str = "Scan Result:\r\n"
            + data.getStringExtra(ZBarConstants.SCAN_RESULT);
    }
}
```

```

str += "\r\nScan Result Type:\r\n"
        + data.getIntExtra(ZBarConstants.SCAN_RESULT_TYPE, 0);
// The value of type indicates one of the symbols listed in Advanced
// Options below.

textView.setText(str);

} else if (resultCode == RESULT_CANCELED) {
    textView.setText("Camera unavailable");
}
}

```

com.smartdevicesdk.database
数据库 SQLite 接口

com.smartdevicesdk.device
不同 Android 设备管理，功能选择,主要用于设备默认串口选择，了解设备串口对应关系后，无需调用此接口

com.smartdevicesdk.fingerprint
指纹模块接口，调用方法请参见 TestDEMO 中 FingerActivity 类

com.smartdevicesdk.idcard
二代身份证接口

使用方法：
初始化接口 IDCardHelper(String, int)，传入串口名称与波特率后，调用 getIDCard()即可获取二代证信息

com.smartdevicesdk.media
多媒体声音播放接口

com.smartdevicesdk.printer
打印机接口(串口)

使用方法：
主要用到 PrinterClassSerialPort 类

函数	说明	参数	备注
PrinterClassSerialPort(Context, Handler)	初始化	Context 上下文实例 Handler 打印机返回数据句柄	
open()	打开设备		
close()	关闭设备		
printText(String)	发送文字数据打印，默认 GBK 编码		
printImage(Bitmap)	发送图片数据打印		
printUnicode(String)	发送 Unicode 打印数据		
write(byte[])	发送 16 进制数据		打印指令数据，可以使用此函数
device : String	串口设备名称		

baudrate : int	串口波特率		
----------------	-------	--	--

com.smartdevicesdk.psam
接触式 PSAM 卡接口

函数	说明	参数	备注
OpenCard(int[], int)	打开设备	参数: [in]int slotno,卡槽编号, 如果传入 0, 自动适配第一个可打开卡的卡槽 [out]unsigned long * fd 传出设备句柄	
CloseCard(long)	关闭设备	参数: [in]unsigned long fd 传入要关闭的设备句柄 返回值: 正确为 0, 错误为非 0	
ResetCard(long, byte[], int[])	设备复位	参数: [in]unsigned long fd 传入要关闭的设备句柄 [out]unsigned char *atr 传出设备复位信息 [in/out]int *atrLen 传出设备复位信息长度 返回值: 正确为 0, 错误为非 0	
CardApdu(long, byte[], int, byte[], int[])	发送命令	参数: [in]unsigned long fd 传入设备句柄 [in]unsigned char *apdu 要发送的 apdu 指令 [in]int apduLength 要发送的 apdu 指令长度 [out]unsigned char*response 返回数据内容 [in/out]int* respLength 返回数据长度 返回值: 正确为 0, 错误为非 0	
CheckCard(long)	检查卡片在位状态	参数: [in]unsigned long fd 传入检测的设备句柄 返回值: 正确为 0, 错误为非 0	功能未启用

com.smartdevicesdk.scanner
一维二维扫描模块接口

函数	说明	参数	备注
ScannerHelper(Context, String, int, Handler)	实例化	Context 上下文实例 String 串口设备名称 int 串口波特率 Handler 接收扫描数据句柄	
Close()	关闭模块		
scan()	启动扫描		

com.smartdevicesdk.stripcard

磁条卡接口(I2C 通讯)

调用 **Stripcardhelper.ReadCard()** 即可获取刷卡后的数据, 详细参见 TestDEMO 中 MagneticCardActivity

com.smartdevicesdk.ui

UI 控制接口

com.smartdevicesdk.utils

字符串函数处理接口