EMBARCADERO CONFERENCE

embarcadero®

Usando a JNI no FireMonkey

Jaques Nascimento

EMBARCADERO CONFERENCE 2022

Usando a JNI no FireMonkey

Com a chegada de novos equipamentos Android para automação comercial, surge a necessidade de integração com novos tipos de hardware, como impressoras, sinaleiros, telas adicionais, entre outros. Nesse momento, precisamos usar as bibliotecas fornecidas pelos fabricantes, em sua maioria escritas em Java.

Nesta palestra, entenda como usar a JNI para tornar tudo isso possível.

Usando a JNI no FireMonkey

JNI - Java Native Interface

Faz a interface entre o Delphi e o Java, nos permitindo:

- Usar classes do SO Android.

- Criarmos nossas próprias classes em Java para serem usadas a partir do de nossos Apps em Delphi.

- Fazermos uso de classes disponibilizadas por fabricantes de hardware.

Usando a JNI no FireMonkey

Caso 1: Usando a segunda tela do aparelho TecToy D2 Mini

Usando a JNI no FireMonkey

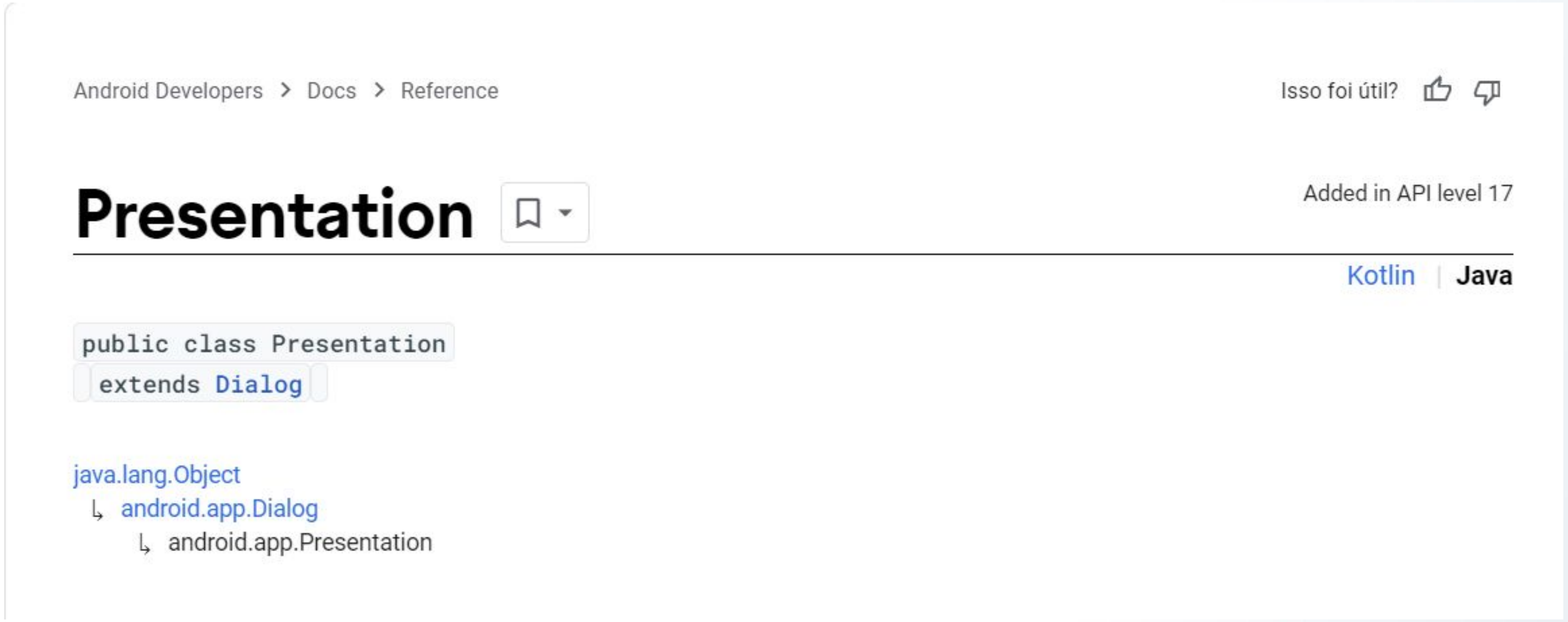
Usando a segunda tela do D2 Mini

O uso de uma segunda tela é previsto no Android, temos muitos exemplos de uso pela internet, mas a classe Presentation que é necessária, não foi incluída em nenhuma das units androidapi.JNI do Delphi, então precisamos criar uma interface para podermos usá-la.

O primeiro passo foi acessar a documentação do Android para obtermos a descrição dessa classe.

Usando a JNI no FireMonkey

https://developer.android.com/reference/android/app/Presentation



Android Developers > Docs > Reference

Isso foi útil?

Presentation

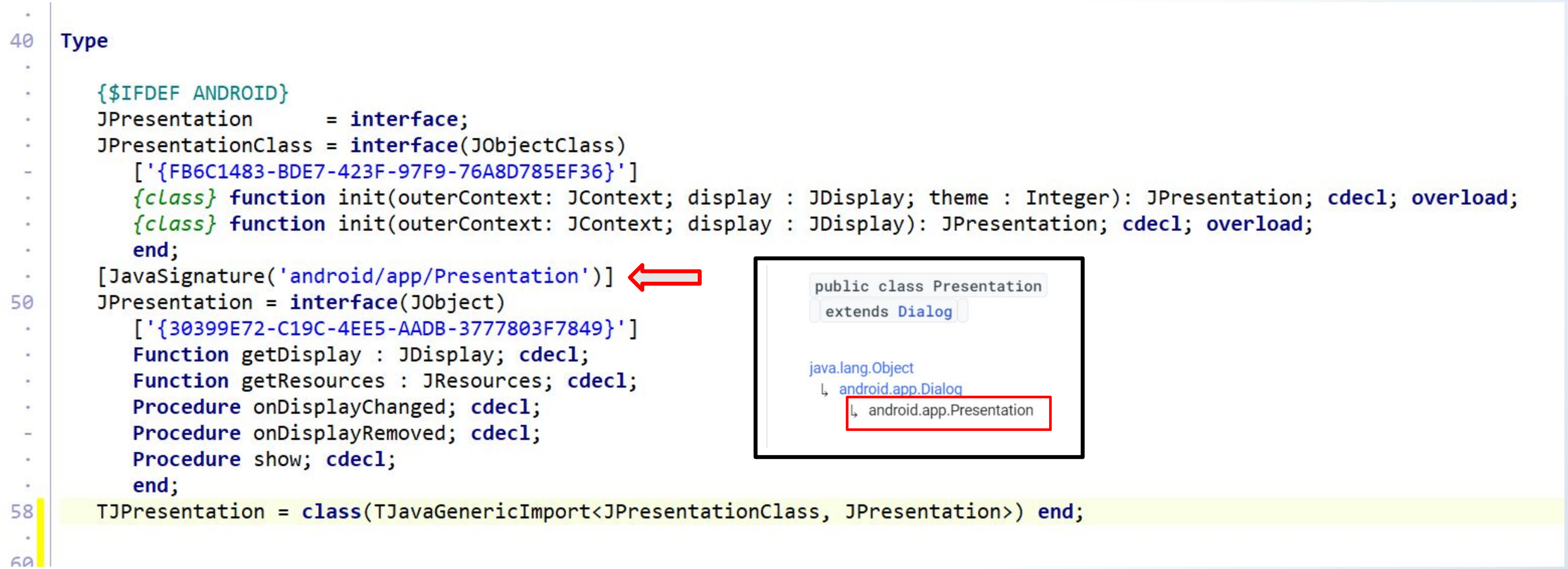
Added in API level 17

Kotlin | Java

```
public class Presentation
  extends Dialog
```

java.lang.Object
  ↳ android.app.Dialog
    ↳ android.app.Presentation

Usando a JNI no FireMonkey

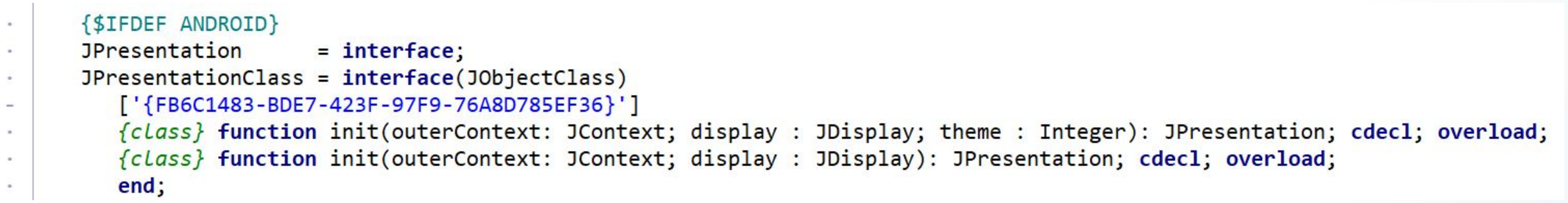


```
Type

  {$IFDEF ANDROID}
  JPresentation      = interface;
  JPresentationClass = interface(JObjectClass)
    ['{FB6C1483-BDE7-423F-97F9-76A8D785EF36}']
    {class} function init(outerContext: JContext; display : JDisplay; theme : Integer): JPresentation; cdecl; overload;
    {class} function init(outerContext: JContext; display : JDisplay): JPresentation; cdecl; overload;
    end;
  [JavaSignature('android/app/Presentation')]
  JPresentation = interface(JObject)
    ['{30399E72-C19C-4EE5-AADB-3777803F7849}']
    Function getDisplay : JDisplay; cdecl;
    Function getResources : JResources; cdecl;
    Procedure onDisplayChanged; cdecl;
    Procedure onDisplayRemoved; cdecl;
    Procedure show; cdecl;
    end;
  TJPresentation = class(TJavaGenericImport<JPresentationClass, JPresentation>) end;
```

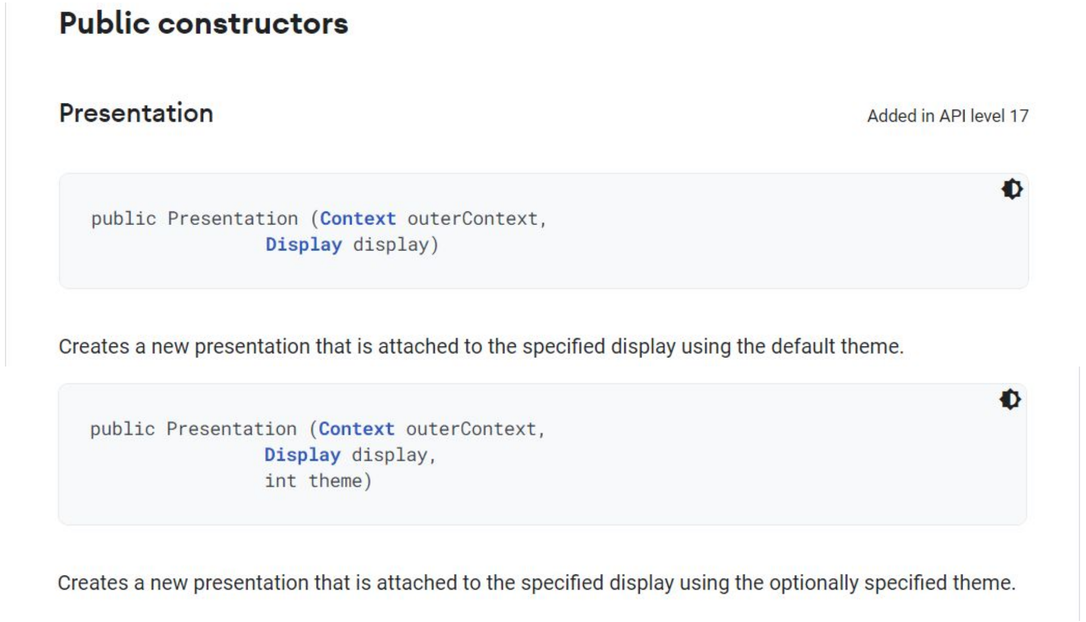
```
public class Presentation
extends Dialog

java.lang.Object
  ↳ android.app.Dialog
      ↳ android.app.Presentation
```

Usando a JNI no FireMonkey

Constructors



```
{$IFDEF ANDROID}
JPresentation      = interface;
JPresentationClass = interface(JObjectClass)
  ['{FB6C1483-BDE7-423F-97F9-76A8D785EF36}']
  {class} function init(outerContext: JContext; display : JDisplay; theme : Integer): JPresentation; cdecl; overload;
  {class} function init(outerContext: JContext; display : JDisplay): JPresentation; cdecl; overload;
  end;
```



Public constructors

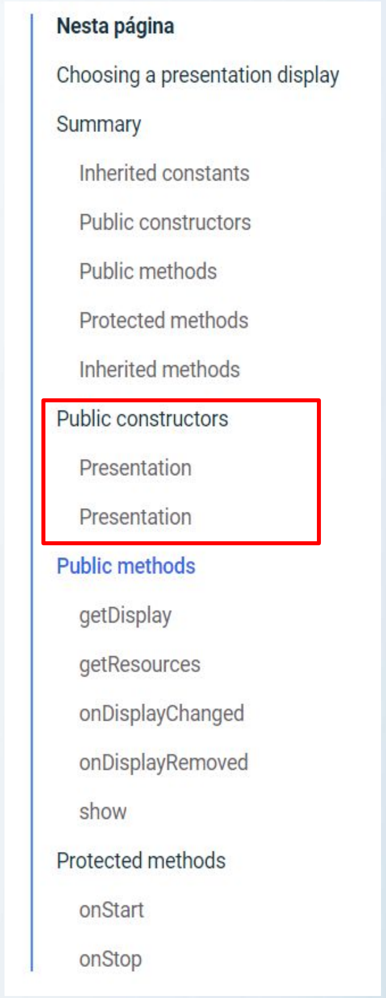
Presentation — Added in API level 17

```
public Presentation (Context outerContext,
                Display display)
```

Creates a new presentation that is attached to the specified display using the default theme.

```
public Presentation (Context outerContext,
                Display display,
                int theme)
```

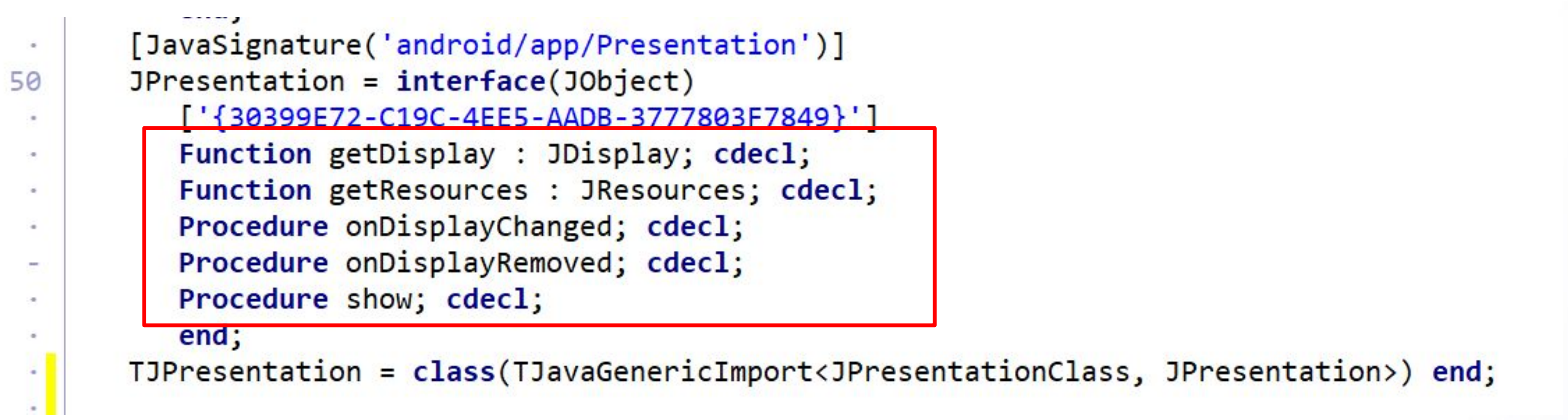
Creates a new presentation that is attached to the specified display using the optionally specified theme.



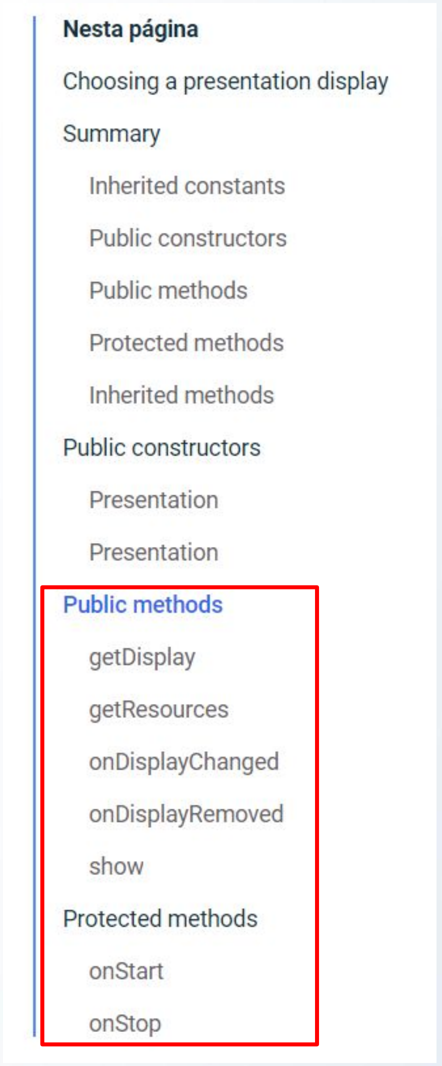
cdecl - Tipo de chamada padrão no linux

Usando a JNI no FireMonkey

Public / Protected methods



```
  [JavaSignature('android/app/Presentation')]
  JPresentation = interface(JObject)
    ['{30399E72-C19C-4EE5-AADB-3777803F7849}']
    Function getDisplay : JDisplay; cdecl;
    Function getResources : JResources; cdecl;
    Procedure onDisplayChanged; cdecl;
    Procedure onDisplayRemoved; cdecl;
    Procedure show; cdecl;
    end;
  TJPresentation = class(TJavaGenericImport<JPresentationClass, JPresentation>) end;
```



Nesta página

Choosing a presentation display

Summary

- Inherited constants
- Public constructors
- Public methods
- Protected methods
- Inherited methods

Public constructors

- Presentation
- Presentation

Public methods

- getDisplay
- getResources
- onDisplayChanged
- onDisplayRemoved
- show

Protected methods

- onStart
- onStop

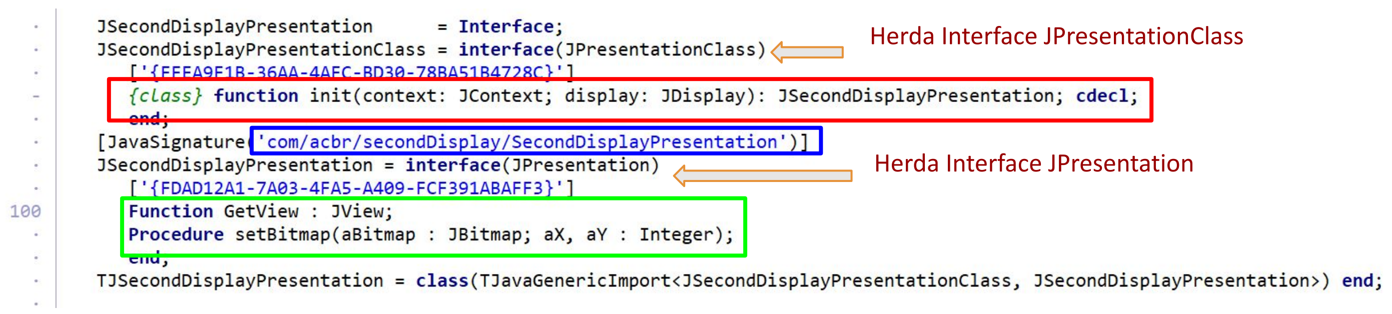
Usando a JNI no FireMonkey

O próximo passo foi criar uma classe em java para fazer a apresentação de um bitmap na segunda tela.

```
package com.acbr.secondDisplay;

import android.app.Activity;
...

public class SecondDisplayPresentation extends Presentation {

     private myView PresentationView;

     public SecondDisplayPresentation(Context outerContext, Display display) {
          super(outerContext, display);
          PresentationView = new myView(outerContext);
          setContentView(PresentationView);
          }

     public View GetView()
          {
          return this.PresentationView;
          }

     public void setBitmap(Bitmap aBitmap, int aX, int aY) {
          PresentationView.setBitmap(aBitmap, aX, aY);
          }

     private class myView extends View {

          private Bitmap BitmapView;
          private int X, Y;

          public myView(Context context) {
          super(context);
          }

          public void setBitmap(Bitmap aBitmap, int aX, int aY) {
               BitmapView = aBitmap;
               X = aX;
               Y = aY;
               invalidate();
               }

          @Override
          protected void onDraw(Canvas canvas) {
               if (BitmapView != null) {
                    Paint myPaint = new Paint();
                    canvas.drawBitmap(BitmapView, X, Y, myPaint);
                    }
               }
     }

}
```

Usando a JNI no FireMonkey

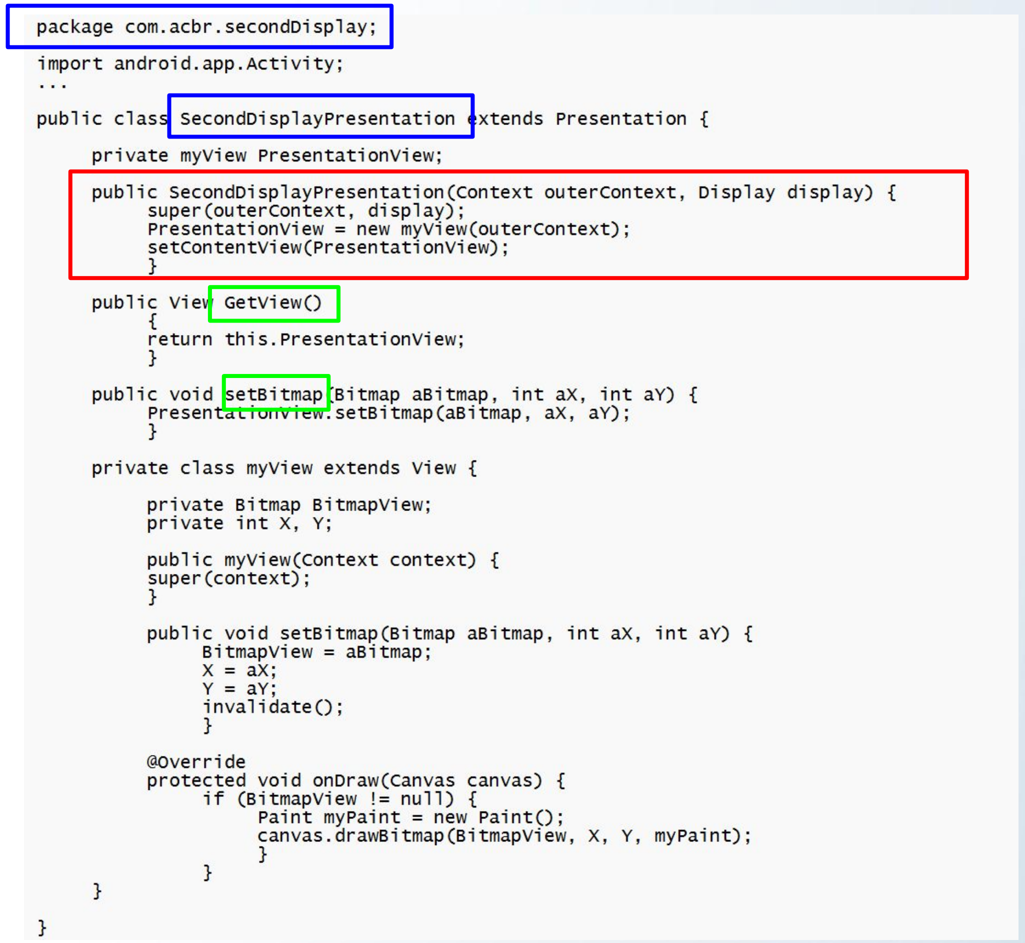
Com a classe Java pronta, criamos a interface para ela ser usada pelo Delphi.



```
JSecondDisplayPresentation      = Interface;
JSecondDisplayPresentationClass = interface(JPresentationClass)
   ['{FFFA9F1B-36AA-4AFC-BD30-78BA51B4728C}']
   {class} function init(context: JContext; display: JDisplay): JSecondDisplayPresentation; cdecl;
   end;
[JavaSignature('com/acbr/secondDisplay/SecondDisplayPresentation')]
JSecondDisplayPresentation = interface(JPresentation)
   ['{FDAD12A1-7A03-4FA5-A409-FCF391ABAFF3}']
   Function GetView : JView;
   Procedure setBitmap(aBitmap : JBitmap; aX, aY : Integer);
   end;
TJSecondDisplayPresentation = class(TJavaGenericImport<JSecondDisplayPresentationClass, JSecondDisplayPresentation>) end;
```

Herda Interface JPresentationClass

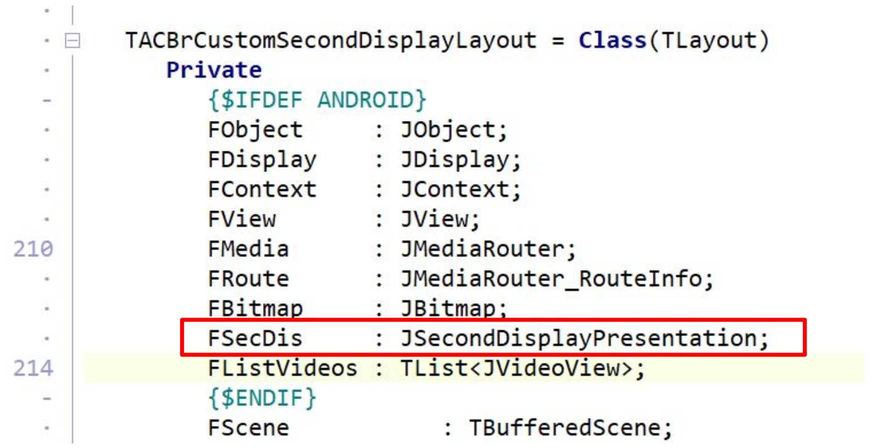
Herda Interface JPresentation



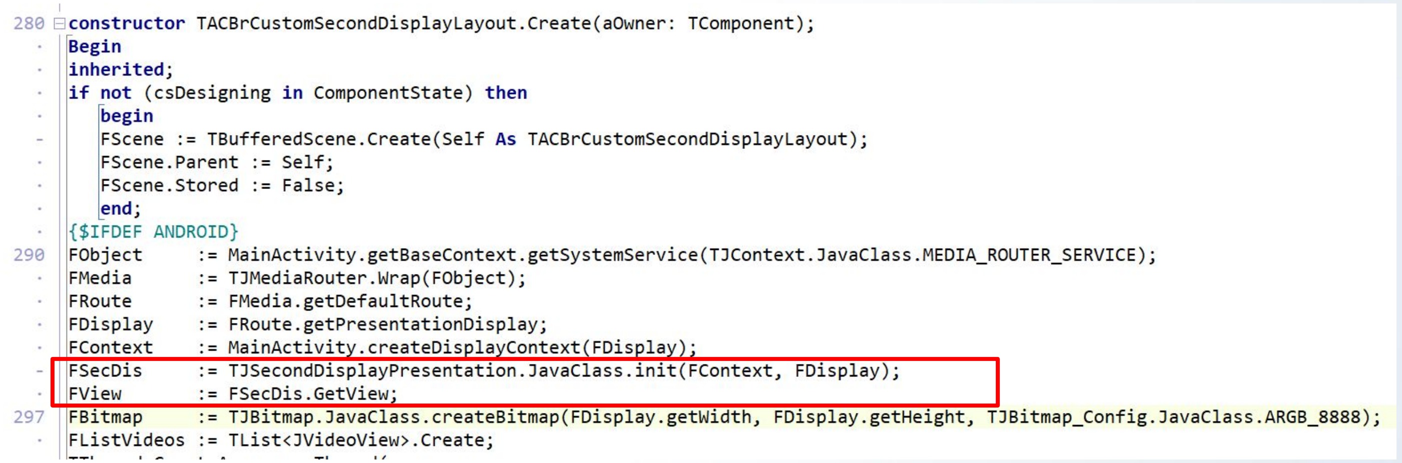
```
package com.acbr.secondDisplay;

import android.app.Activity;
...

public class SecondDisplayPresentation extends Presentation {

     private myView PresentationView;

     public SecondDisplayPresentation(Context outerContext, Display display) {
          super(outerContext, display);
          PresentationView = new myView(outerContext);
          setContentView(PresentationView);
          }

     public View GetView()
          {
          return this.PresentationView;
          }

     public void setBitmap(Bitmap aBitmap, int aX, int aY) {
          PresentationView.setBitmap(aBitmap, aX, aY);
          }

     private class myView extends View {

          private Bitmap BitmapView;
          private int X, Y;

          public myView(Context context) {
          super(context);
          }

          public void setBitmap(Bitmap aBitmap, int aX, int aY) {
               BitmapView = aBitmap;
               X = aX;
               Y = aY;
               invalidate();
               }

          @Override
          protected void onDraw(Canvas canvas) {
               if (BitmapView != null) {
                    Paint myPaint = new Paint();
                    canvas.drawBitmap(BitmapView, X, Y, myPaint);
                    }
               }
     }

}
```

Usando a JNI no FireMonkey

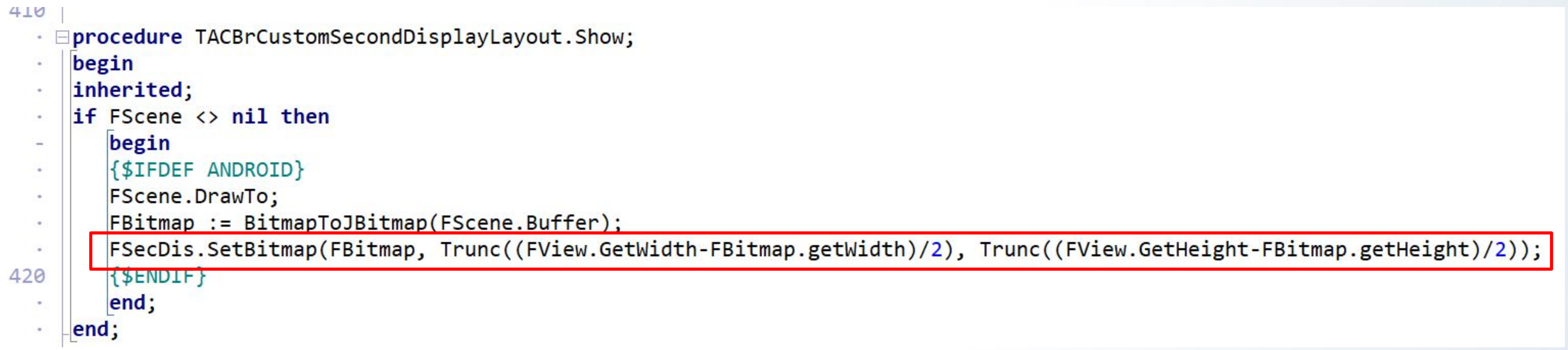
Usando as classes Java no Delphi



```
TACBrCustomSecondDisplayLayout = Class(TLayout)
   Private
      {$IFDEF ANDROID}
      FObject     : JObject;
      FDisplay    : JDisplay;
      FContext    : JContext;
      FView       : JView;
      FMedia      : JMediaRouter;
      FRoute      : JMediaRouter_RouteInfo;
      FBitmap     : JBitmap;
      FSecDis     : JSecondDisplayPresentation;
      FListVideos : TList<JVideoView>;
      {$ENDIF}
      FScene           : TBufferedScene;
```



```
constructor TACBrCustomSecondDisplayLayout.Create(aOwner: TComponent);
Begin
inherited;
if not (csDesigning in ComponentState) then
   begin
   FScene := TBufferedScene.Create(Self As TACBrCustomSecondDisplayLayout);
   FScene.Parent := Self;
   FScene.Stored := False;
   end;
{$IFDEF ANDROID}
FObject     := MainActivity.getBaseContext.getSystemService(TJContext.JavaClass.MEDIA_ROUTER_SERVICE);
FMedia      := TJMediaRouter.Wrap(FObject);
FRoute      := FMedia.getDefaultRoute;
FDisplay    := FRoute.getPresentationDisplay;
FContext    := MainActivity.createDisplayContext(FDisplay);
FSecDis     := TJSecondDisplayPresentation.JavaClass.init(FContext, FDisplay);
FView       := FSecDis.GetView;
FBitmap     := TJBitmap.JavaClass.createBitmap(FDisplay.getWidth, FDisplay.getHeight, TJBitmap_Config.JavaClass.ARGB_8888);
FListVideos := TList<JVideoView>.Create;
```

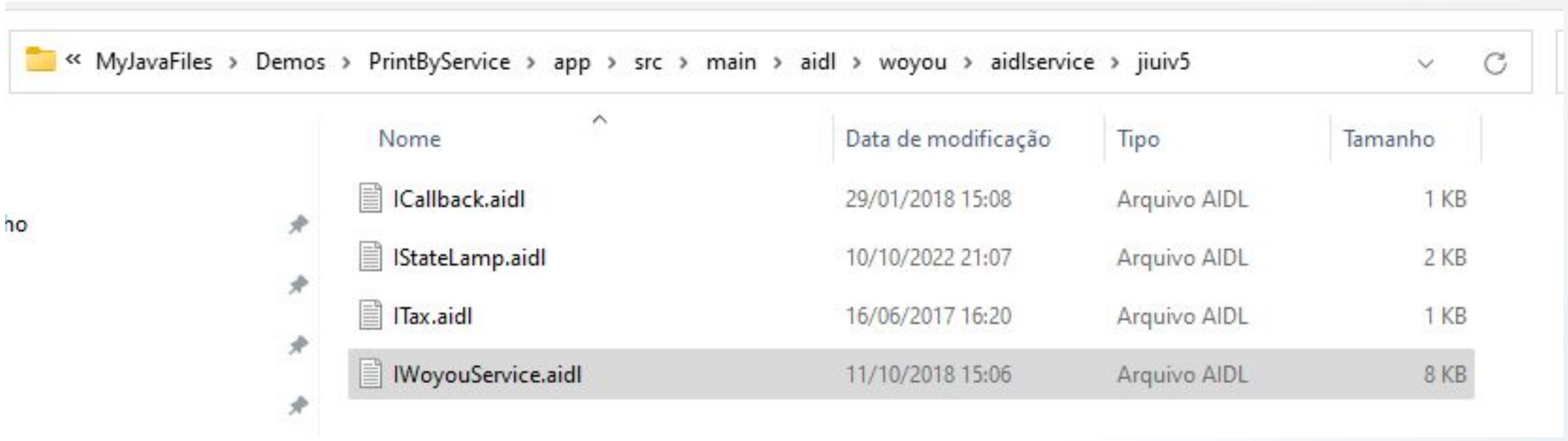


```
procedure TACBrCustomSecondDisplayLayout.Show;
begin
inherited;
if FScene <> nil then
   begin
   {$IFDEF ANDROID}
   FScene.DrawTo;
   FBitmap := BitmapToJBitmap(FScene.Buffer);
   FSecDis.SetBitmap(FBitmap, Trunc((FView.GetWidth-FBitmap.getWidth)/2), Trunc((FView.GetHeight-FBitmap.getHeight)/2));
   {$ENDIF}
   end;
end;
```

Usando a JNI no FireMonkey

Caso 2: Usando a impressora do aparelho TecToy D2 Mini

No D2 Mini, o “driver” da impressora é distribuído no formato AIDL (Android Interface Definition Language). Para serem usados, precisam ser compilados para classes em java.

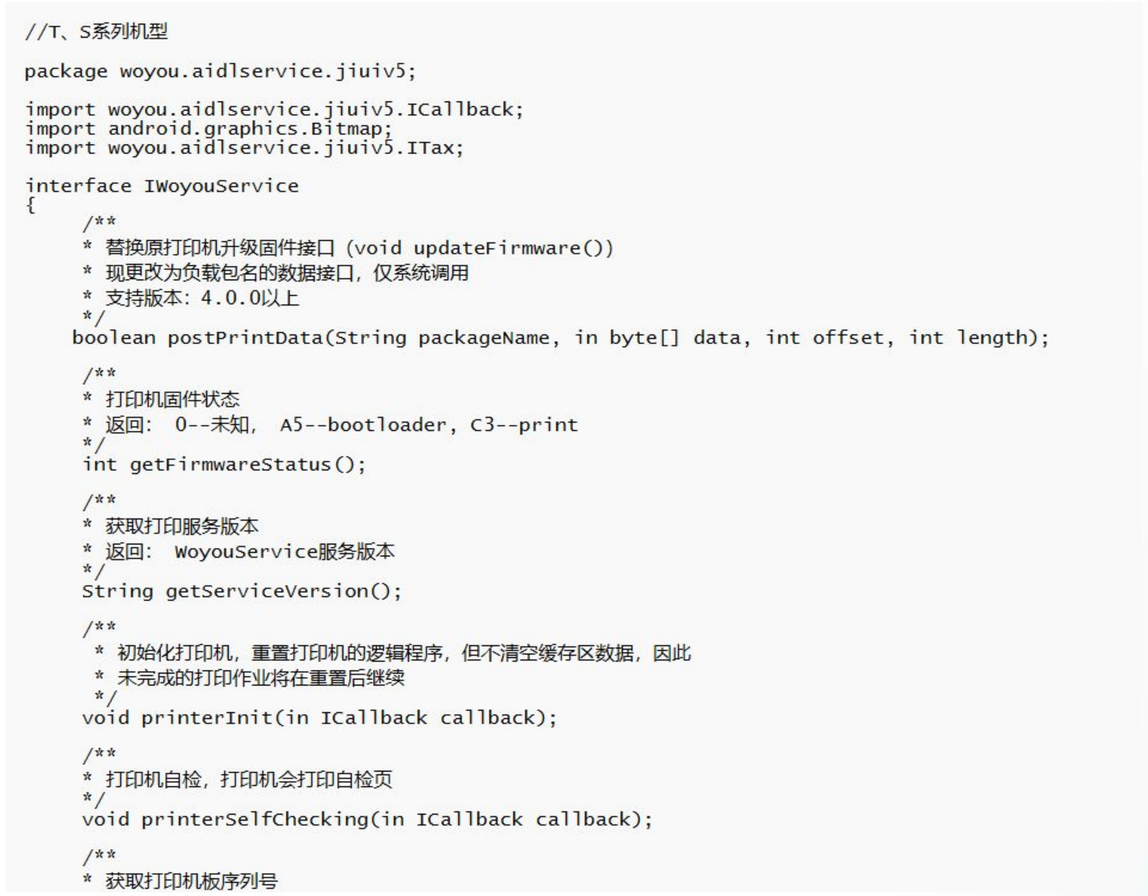


« MyJavaFiles › Demos › PrintByService › app › src › main › aidl › woyou › aidlservice › jiuiv5

Nome	Data de modificação	Tipo	Tamanho
ICallback.aidl	29/01/2018 15:08	Arquivo AIDL	1 KB
IStateLamp.aidl	10/10/2022 21:07	Arquivo AIDL	2 KB
ITax.aidl	16/06/2017 16:20	Arquivo AIDL	1 KB
IWoyouService.aidl	11/10/2018 15:06	Arquivo AIDL	8 KB

Usando a JNI no FireMonkey

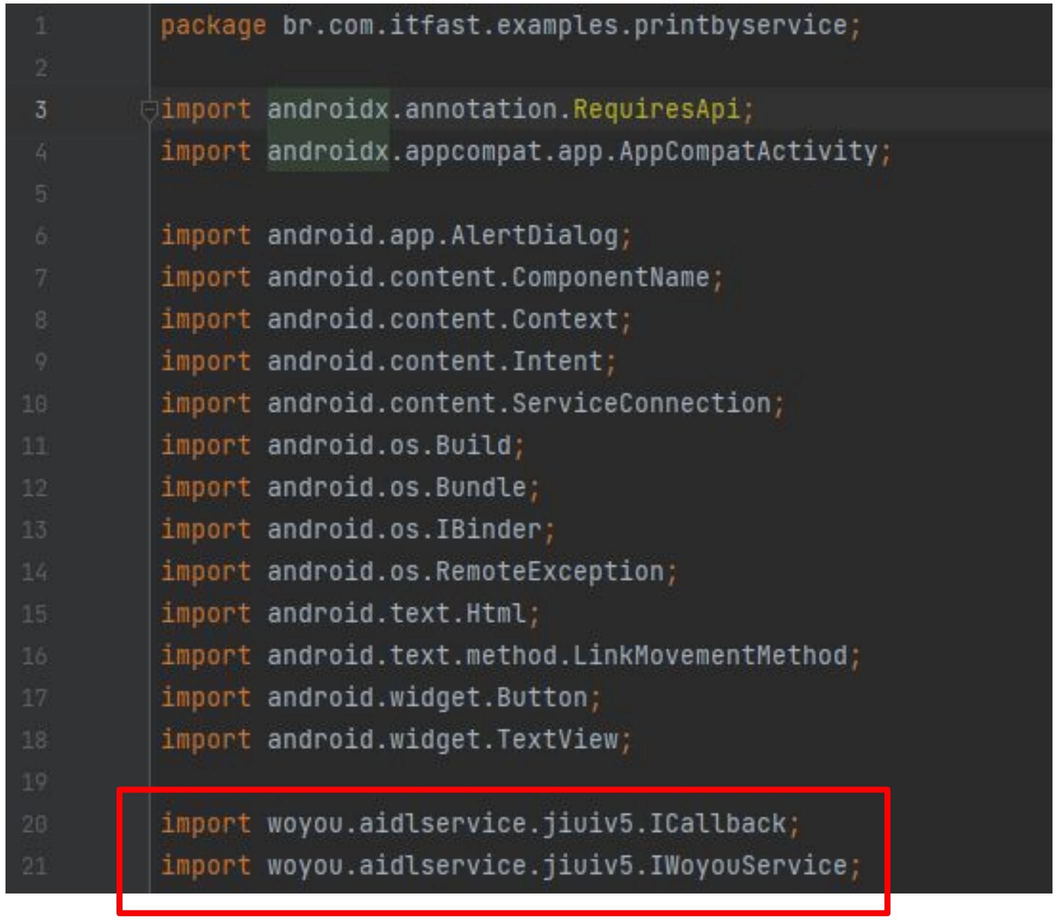
Exemplo de AIDL do serviço de impressão do D2 Mini



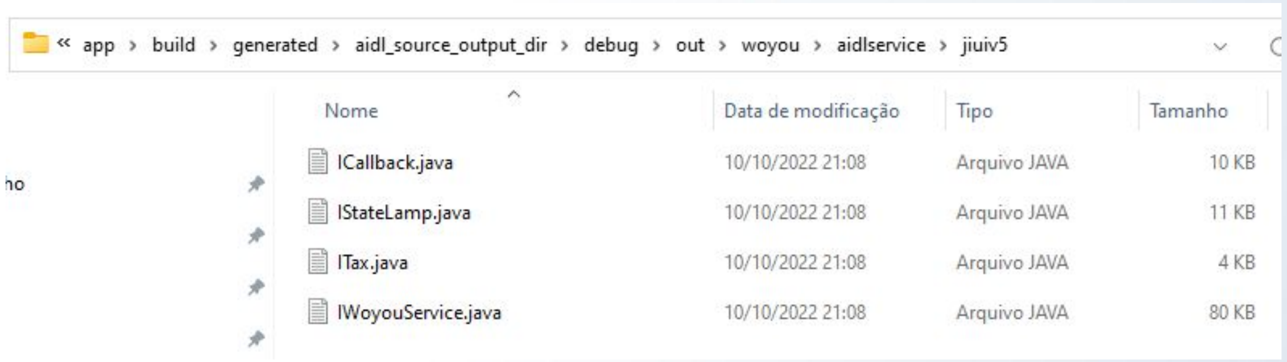
```
//T、S系列机型

package woyou.aidlservice.jiuiv5;

import woyou.aidlservice.jiuiv5.ICallback;
import android.graphics.Bitmap;
import woyou.aidlservice.jiuiv5.ITax;

interface IWoyouService
{
     /**
     * 替换原打印机升级固件接口 (void updateFirmware())
     * 现更改为负载包名的数据接口，仅系统调用
     * 支持版本：4.0.0以上
     */
    boolean postPrintData(String packageName, in byte[] data, int offset, int length);

     /**
     * 打印机固件状态
     * 返回： 0--未知， A5--bootloader, C3--print
     */
     int getFirmwareStatus();

     /**
     * 获取打印服务版本
     * 返回： WoyouService服务版本
     */
     String getServiceVersion();

     /**
      * 初始化打印机，重置打印机的逻辑程序，但不清空缓存区数据，因此
      * 未完成的打印作业将在重置后继续
      */
     void printerInit(in ICallback callback);

     /**
     * 打印机自检，打印机会打印自检页
     */
     void printerSelfChecking(in ICallback callback);

     /**
     * 获取打印机板序列号
```

Usando a JNI no FireMonkey

Exemplo de AIDL do serviço de impressão do D2 Mini


```java
package br.com.itfast.examples.printbyservice;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.app.AlertDialog;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Build;
import android.os.Bundle;
import android.os.IBinder;
import android.os.RemoteException;
import android.text.Html;
import android.text.method.LinkMovementMethod;
import android.widget.Button;
import android.widget.TextView;

import woyou.aidlservice.jiuiv5.ICallback;
import woyou.aidlservice.jiuiv5.IWoyouService;
```

Utilizando o Android Studio para compilar um fonte java que importa os arquivos AIDL, as interfaces são criadas na pasta build do projeto: *C:\MyJavaFiles\Demos\PrintByService\app\build\generated\aidl_source_output_dir*

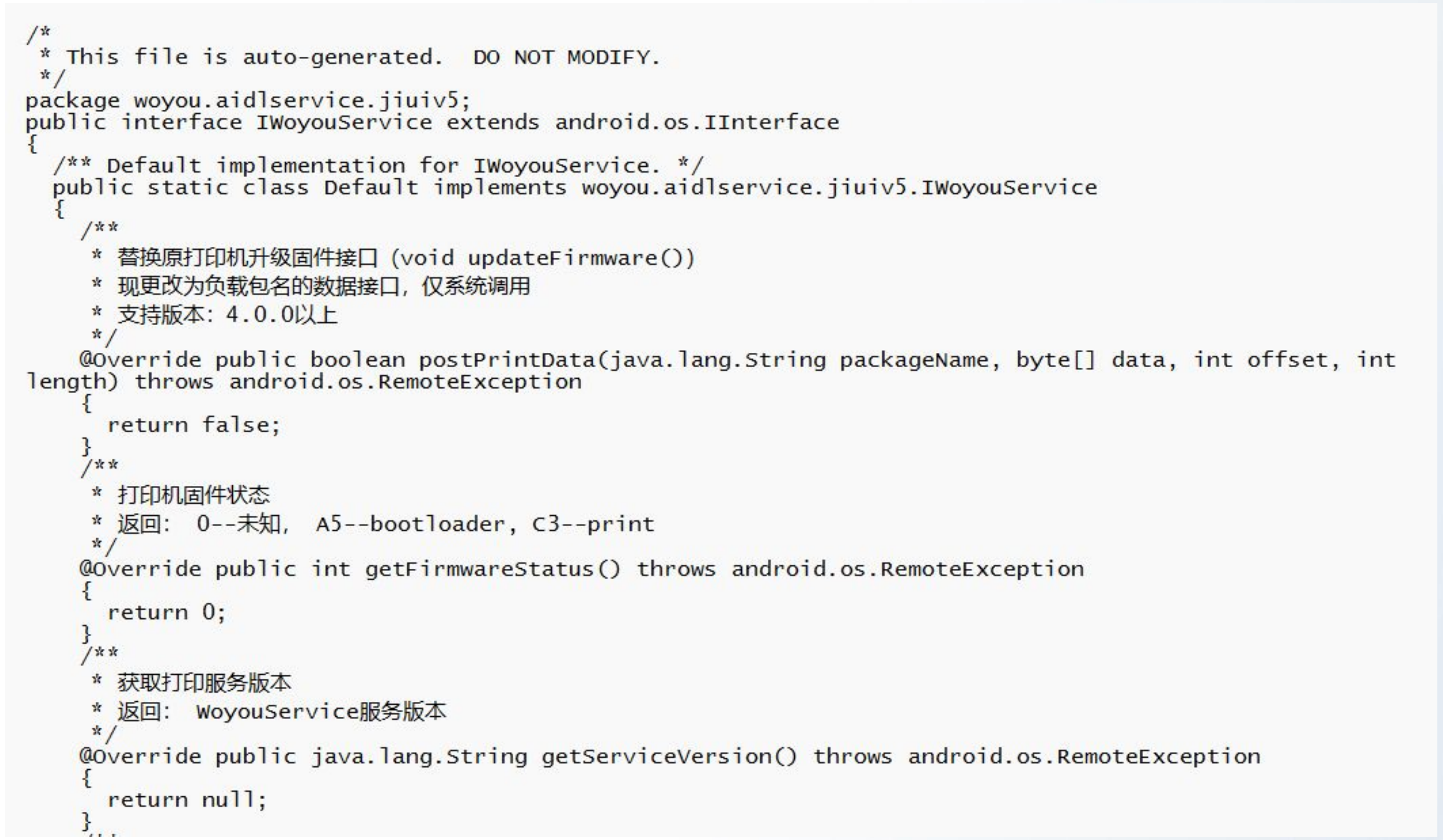

« app › build › generated › aidl_source_output_dir › debug › out › woyou › aidlservice › jiuiv5

Nome	Data de modificação	Tipo	Tamanho
ICallback.java	10/10/2022 21:08	Arquivo JAVA	10 KB
IStateLamp.java	10/10/2022 21:08	Arquivo JAVA	11 KB
ITax.java	10/10/2022 21:08	Arquivo JAVA	4 KB
IWoyouService.java	10/10/2022 21:08	Arquivo JAVA	80 KB

Usando a JNI no FireMonkey

Exemplo de AIDL do serviço de impressão do D2 Mini

Exemplo da Interface em Java criada na compilação



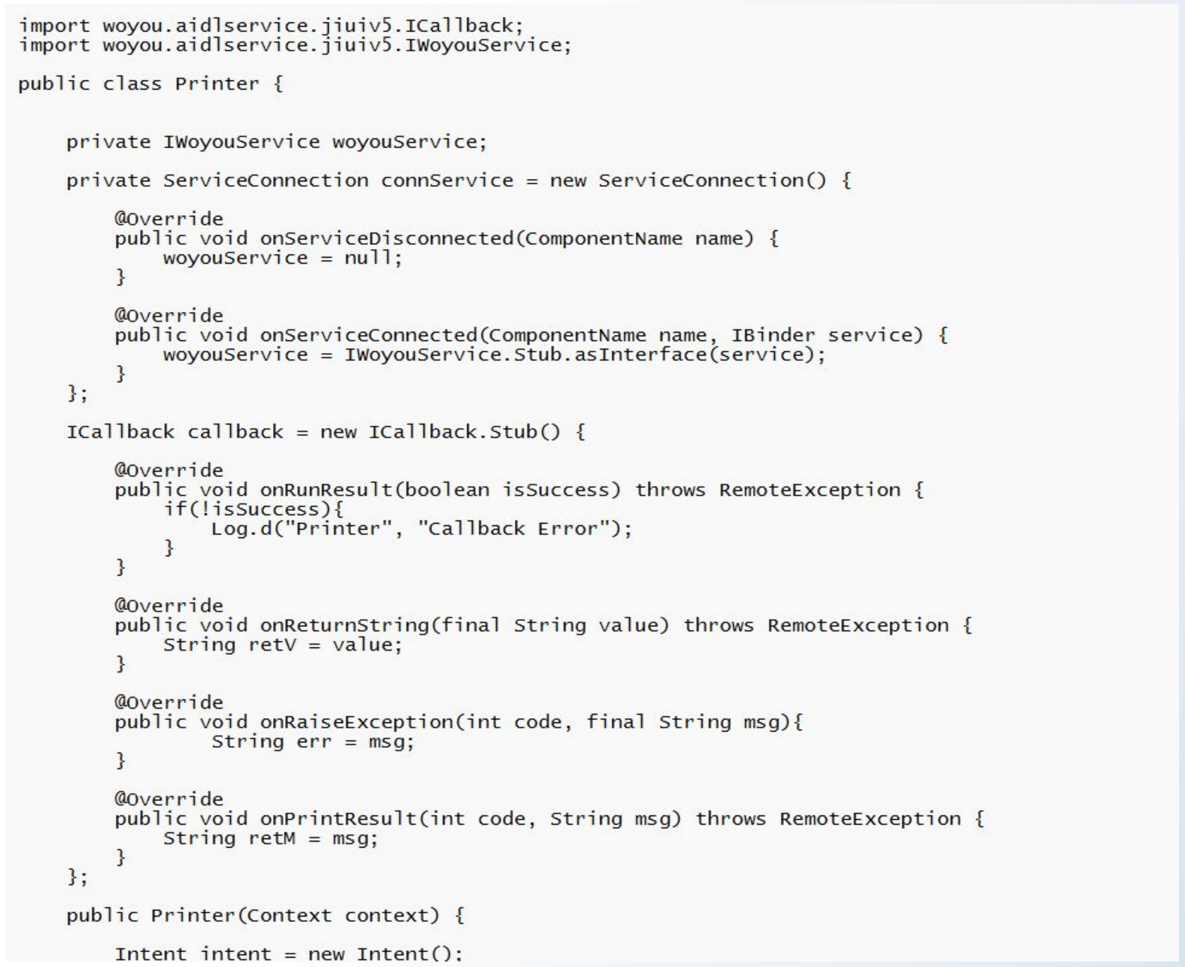
```
/*
 * This file is auto-generated.  DO NOT MODIFY.
 */
package woyou.aidlservice.jiuiv5;
public interface IWoyouService extends android.os.IInterface
{
  /** Default implementation for IWoyouService. */
  public static class Default implements woyou.aidlservice.jiuiv5.IWoyouService
  {
    /**
     * 替换原打印机升级固件接口 (void updateFirmware())
     * 现更改为负载包名的数据接口，仅系统调用
     * 支持版本：4.0.0以上
     */
    @Override public boolean postPrintData(java.lang.String packageName, byte[] data, int offset, int
length) throws android.os.RemoteException
    {
      return false;
    }
    /**
     * 打印机固件状态
     * 返回：  0--未知，  A5--bootloader, C3--print
     */
    @Override public int getFirmwareStatus() throws android.os.RemoteException
    {
      return 0;
    }
    /**
     * 获取打印服务版本
     * 返回：  WoyouService服务版本
     */
    @Override public java.lang.String getServiceVersion() throws android.os.RemoteException
    {
      return null;
    }
```

Usando a JNI no FireMonkey

Exemplo de AIDL do serviço de impressão do D2 Mini

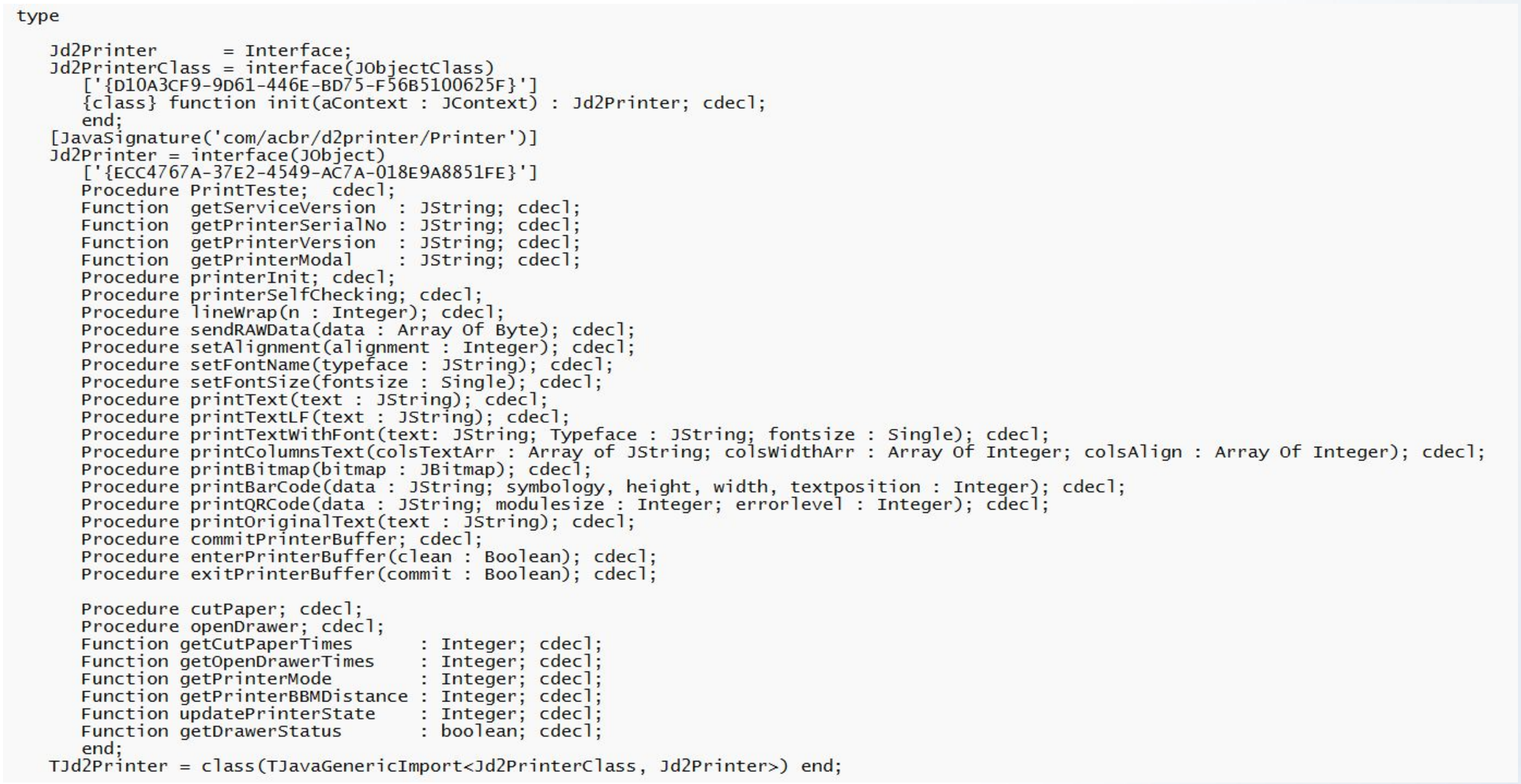
O próximo passo é criar uma classe Java com os métodos descritos na interface.

Após a criação da parte Java, escrevemos a interface em Delphi conforme vimos anteriormente.



```
import woyou.aidlservice.jiuiv5.ICallback;
import woyou.aidlservice.jiuiv5.IWoyouService;

public class Printer {

    private IWoyouService woyouService;

    private ServiceConnection connService = new ServiceConnection() {

        @Override
        public void onServiceDisconnected(ComponentName name) {
            woyouService = null;
        }

        @Override
        public void onServiceConnected(ComponentName name, IBinder service) {
            woyouService = IWoyouService.Stub.asInterface(service);
        }
    };

    ICallback callback = new ICallback.Stub() {

        @Override
        public void onRunResult(boolean isSuccess) throws RemoteException {
            if(!isSuccess){
                Log.d("Printer", "Callback Error");
            }
        }

        @Override
        public void onReturnString(final String value) throws RemoteException {
            String retV = value;
        }

        @Override
        public void onRaiseException(int code, final String msg){
                String err = msg;
        }

        @Override
        public void onPrintResult(int code, String msg) throws RemoteException {
            String retM = msg;
        }
    };

    public Printer(Context context) {

        Intent intent = new Intent();
```

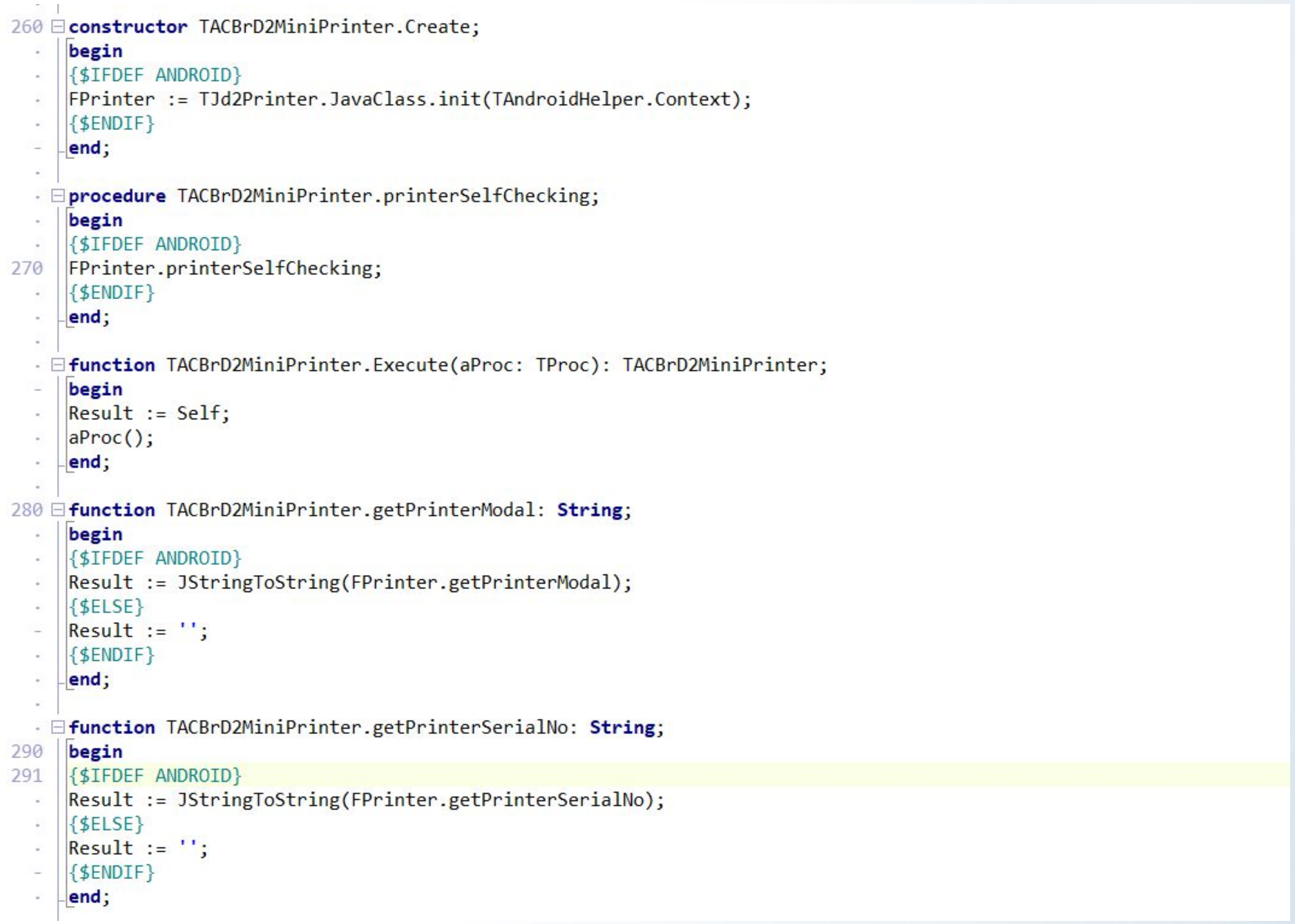
Usando a JNI no FireMonkey



```
type

   Jd2Printer      = Interface;
   Jd2PrinterClass = interface(JObjectClass)
      ['{D10A3CF9-9D61-446E-BD75-F56B5100625F}']
      {class} function init(aContext : JContext) : Jd2Printer; cdecl;
      end;
   [JavaSignature('com/acbr/d2printer/Printer')]
   Jd2Printer = interface(JObject)
      ['{ECC4767A-37E2-4549-AC7A-018E9A8851FE}']
      Procedure PrintTeste;  cdecl;
      Function  getServiceVersion  : JString; cdecl;
      Function  getPrinterSerialNo : JString; cdecl;
      Function  getPrinterVersion  : JString; cdecl;
      Function  getPrinterModal    : JString; cdecl;
      Procedure printerInit; cdecl;
      Procedure printerSelfChecking; cdecl;
      Procedure lineWrap(n : Integer); cdecl;
      Procedure sendRAWData(data : Array Of Byte); cdecl;
      Procedure setAlignment(alignment : Integer); cdecl;
      Procedure setFontName(typeface : JString); cdecl;
      Procedure setFontSize(fontsize : Single); cdecl;
      Procedure printText(text : JString); cdecl;
      Procedure printTextLF(text : JString); cdecl;
      Procedure printTextWithFont(text: JString; Typeface : JString; fontsize : Single); cdecl;
      Procedure printColumnsText(colsTextArr : Array of JString; colsWidthArr : Array Of Integer; colsAlign : Array Of Integer); cdecl;
      Procedure printBitmap(bitmap : JBitmap); cdecl;
      Procedure printBarCode(data : JString; symbology, height, width, textposition : Integer); cdecl;
      Procedure printQRCode(data : JString; modulesize : Integer; errorlevel : Integer); cdecl;
      Procedure printOriginalText(text : JString); cdecl;
      Procedure commitPrinterBuffer; cdecl;
      Procedure enterPrinterBuffer(clean : Boolean); cdecl;
      Procedure exitPrinterBuffer(commit : Boolean); cdecl;

      Procedure cutPaper; cdecl;
      Procedure openDrawer; cdecl;
      Function getCutPaperTimes      : Integer; cdecl;
      Function getOpenDrawerTimes    : Integer; cdecl;
      Function getPrinterMode        : Integer; cdecl;
      Function getPrinterBBMDistance : Integer; cdecl;
      Function updatePrinterState    : Integer; cdecl;
      Function getDrawerStatus       : boolean; cdecl;
      end;
   TJd2Printer = class(TJavaGenericImport<Jd2PrinterClass, Jd2Printer>) end;
```

Usando a JNI no FireMonkey

Para finalizar, escrevemos a classe que faz uso da interface.



```
constructor TACBrD2MiniPrinter.Create;
begin
{$IFDEF ANDROID}
FPrinter := TJd2Printer.JavaClass.init(TAndroidHelper.Context);
{$ENDIF}
end;

procedure TACBrD2MiniPrinter.printerSelfChecking;
begin
{$IFDEF ANDROID}
FPrinter.printerSelfChecking;
{$ENDIF}
end;

function TACBrD2MiniPrinter.Execute(aProc: TProc): TACBrD2MiniPrinter;
begin
Result := Self;
aProc();
end;

function TACBrD2MiniPrinter.getPrinterModal: String;
begin
{$IFDEF ANDROID}
Result := JStringToString(FPrinter.getPrinterModal);
{$ELSE}
Result := '';
{$ENDIF}
end;

function TACBrD2MiniPrinter.getPrinterSerialNo: String;
begin
{$IFDEF ANDROID}
Result := JStringToString(FPrinter.getPrinterSerialNo);
{$ELSE}
Result := '';
{$ENDIF}
end;
```

Usando a JNI no FireMonkey

Podemos concluir que com um conhecimento básico de Java e sabendo criar a interface JNI no delphi, podemos utilizar qualquer recurso disponível no Android.

@ jaquesrnj@gmail.com

in https://www.linkedin.com/in/jaques-nascimento-júnior-00a88114a/

OBRIGADO