



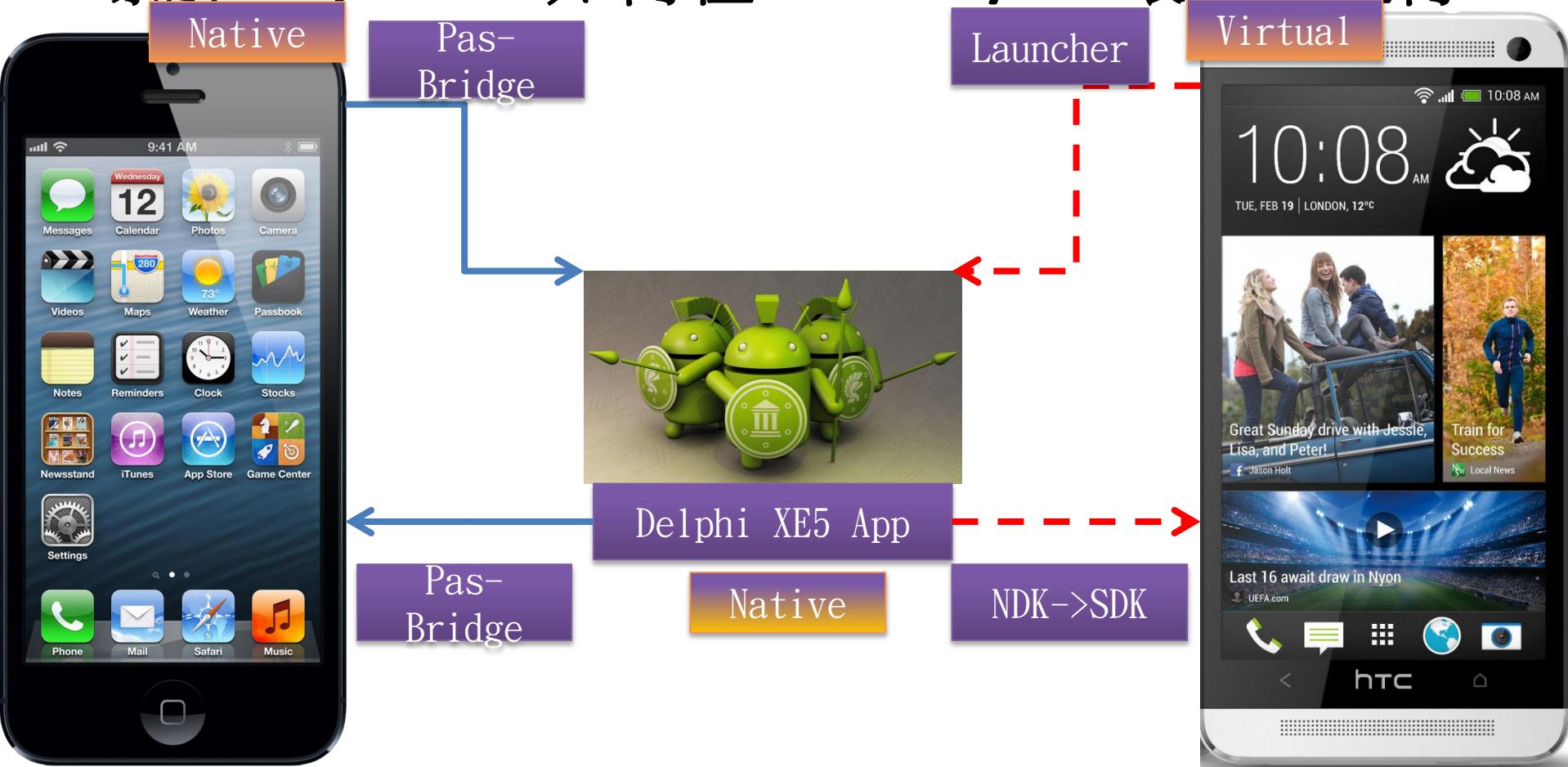
Delphi XE5深入技術研討會

系列1：深入瞭解XE5和移動開發

Agenda

- 瞭解Delphi XE5如何在Android/iOS環境中運行
- 通往Android/iOS環境的重要觀念, 類別/介面和方法
- 動態呼叫?靜態呼叫?
- 使用幾個實例來說明吧
 - Delphi XE5未封裝的API?
 - BlueTooth?
 - Bar Code?
 - 其他第3方函式庫?
- 結論

瞭解Delphi XE5如何在Android/iOS環境中運行



瞭解Delphi XE5如何在Android/iOS環境中運行

- 由於時間的限制,以下先以Android平台為說明範例。但由於其中許多的概念在iOS平台也類似,因此瞭解了Android平台之後在iOS平台也可使用類似的概念和方法。



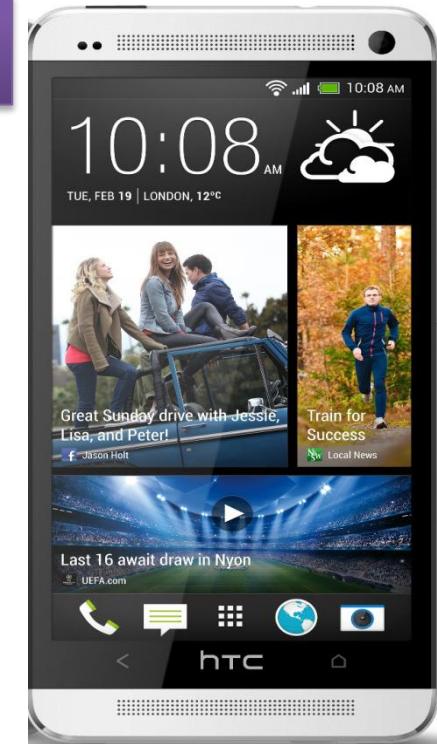
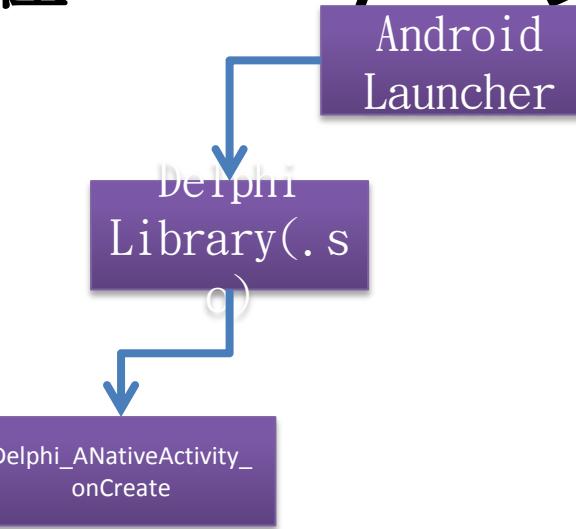
瞭解Delphi XE5如何在Android/iOS環境中運行



Delphi XE5 App

Delphi原生環境

JNI



Java虛擬環境

embarcadero®

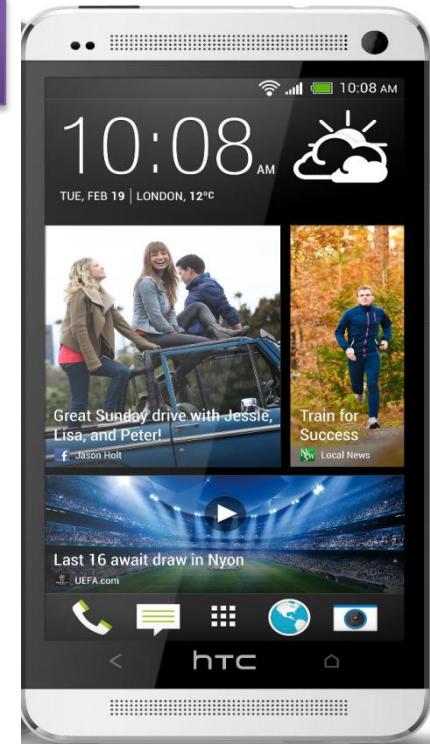
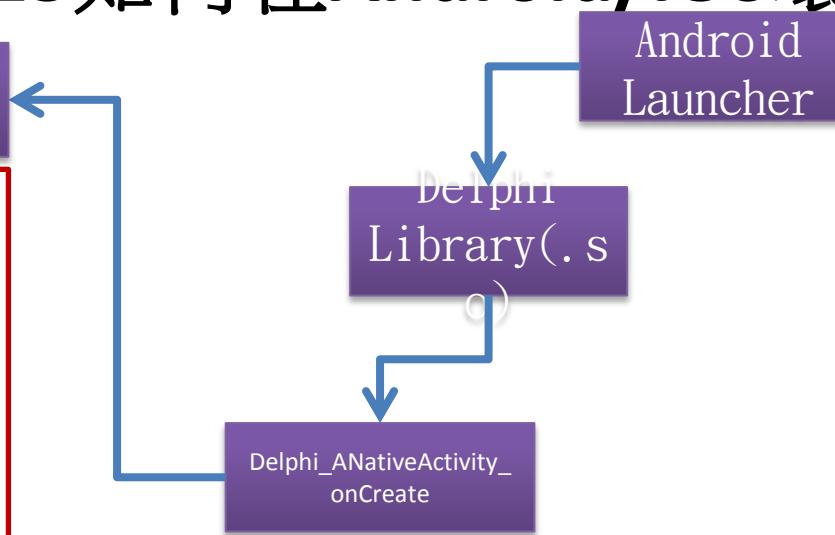
瞭解Delphi XE5如何在Android/iOS環境中運行

com/embcadero/firemonkey/FMXNativeActivity



Delphi XE5 App

Delphi原生環境



Java虛擬環境

embarcadero®

瞭解Delphi XE5如何在Android/iOS環境中運行

```

procedure
Java_com_example_jni_MainAppClass_MyNativeMethod(Env: PJNIEnv;
This: JObject); cdecl;
var
JavaClass: jclass;
JavaMethodID: jmethodID;
begin
{ resolve the call and method we want to call }
JavaClass := Env^.GetObjectClass(Env, This);
JavaMethodID := Env^.GetMethodID(Env, JavaClass, 'showMessage', '()V');

{ call the Java method }
Env^.CallVoidMethod(Env, This, JavaMethodID);
end;

exports
Java_com_example_hellojni_HelloJni_MyNativeMethod;

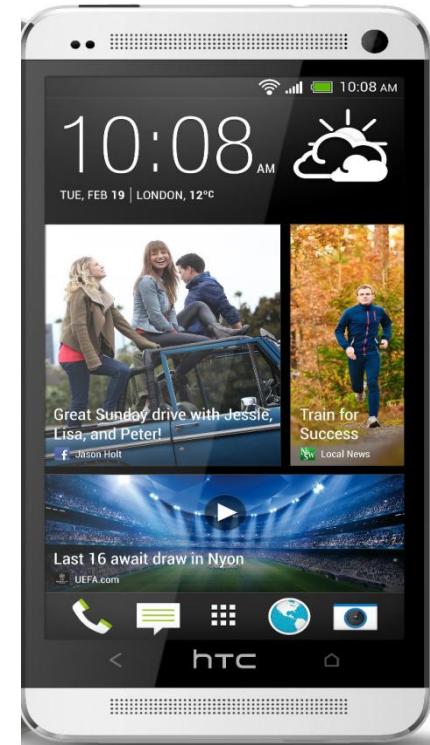
```



Delphi XE5 App



Delphi原生環境



Java虛擬環境

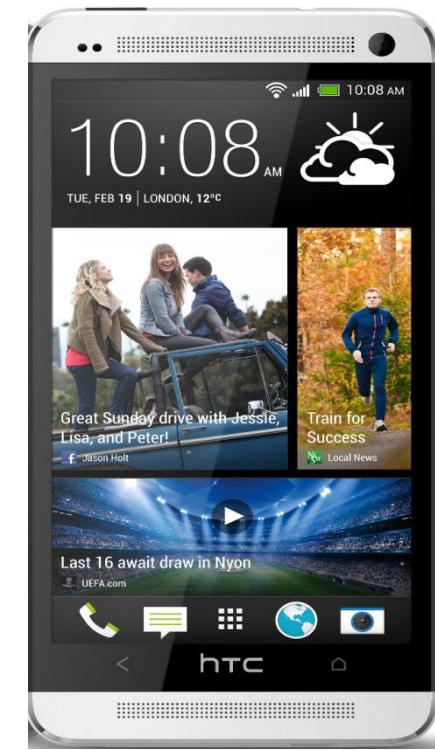
瞭解Delphi XE5如何在Android/iOS環境中運行



Delphi原生環境



- 靜態呼叫
 - 用Delphi語言定義Java API, 再使用Delphi呼叫
- 動態呼叫
 - 直接載入Java Class呼叫



Java虛擬環境

embarcadero

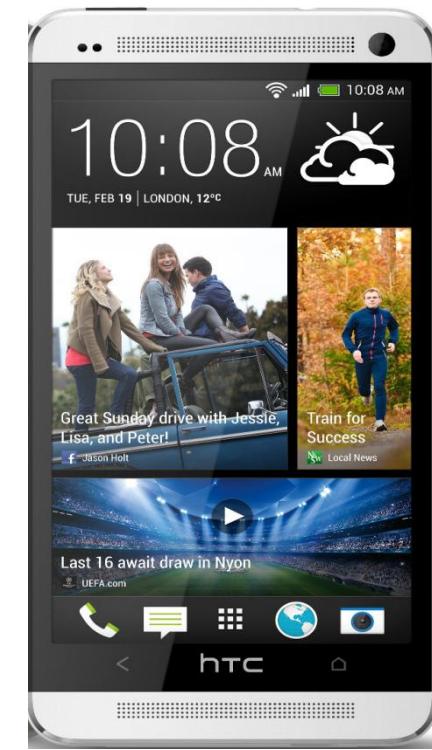
瞭解Delphi XE5如何在Android/iOS環境中運行



Delphi XE5 App

Delphi原生環境

您需要瞭解一些重要觀念,
類別/介面和方法



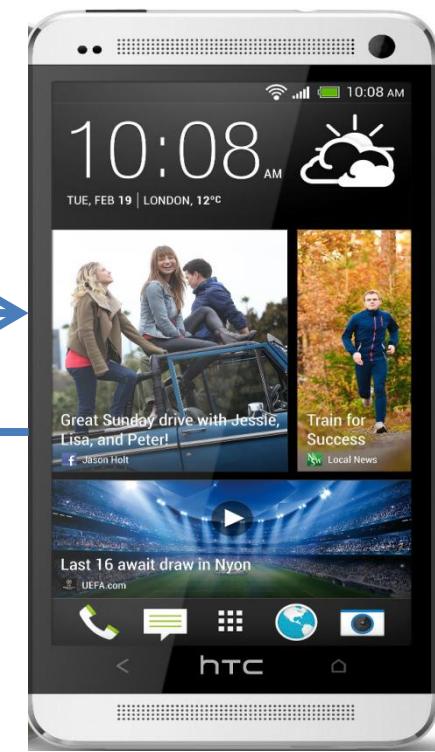
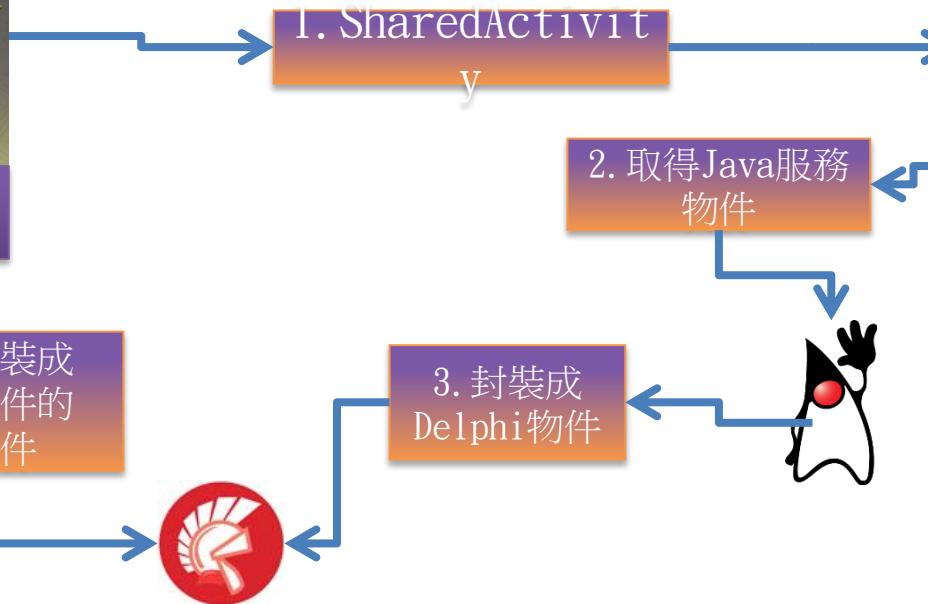
Java虛擬環境

embarcadero

通往Android/iOS環境的重要觀念, 類別/介面和方法

- 一般來說在Delphi的原生世界中我們通常會需要Android(iOS)進行下面的工作
 - 使用Android(iOS)的服務
 - 呼叫Android(iOS)的API
 - 使用和呼叫Android(iOS)的第3方函式庫或是自己撰寫的Java(Objective-C)程式碼
- 讓我們討論如何完成上述的工作

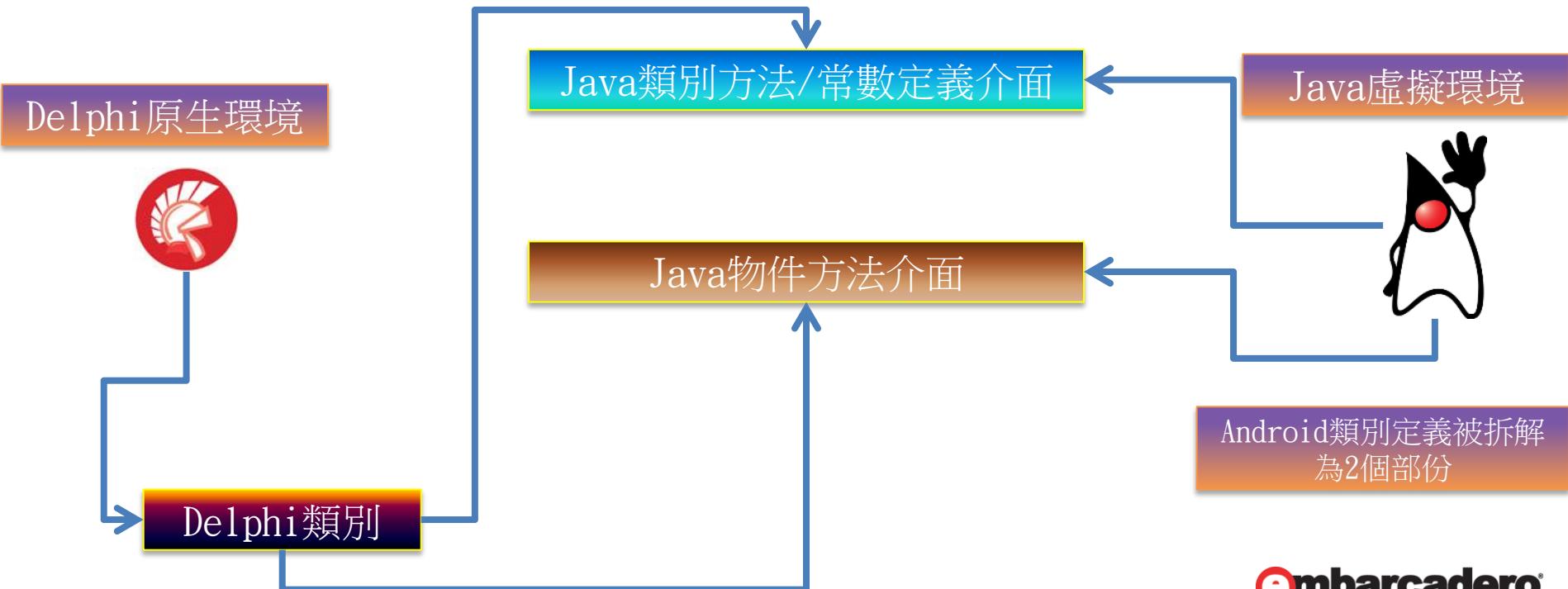
通往Android/iOS環境的重要觀念, 類別/介面和方法



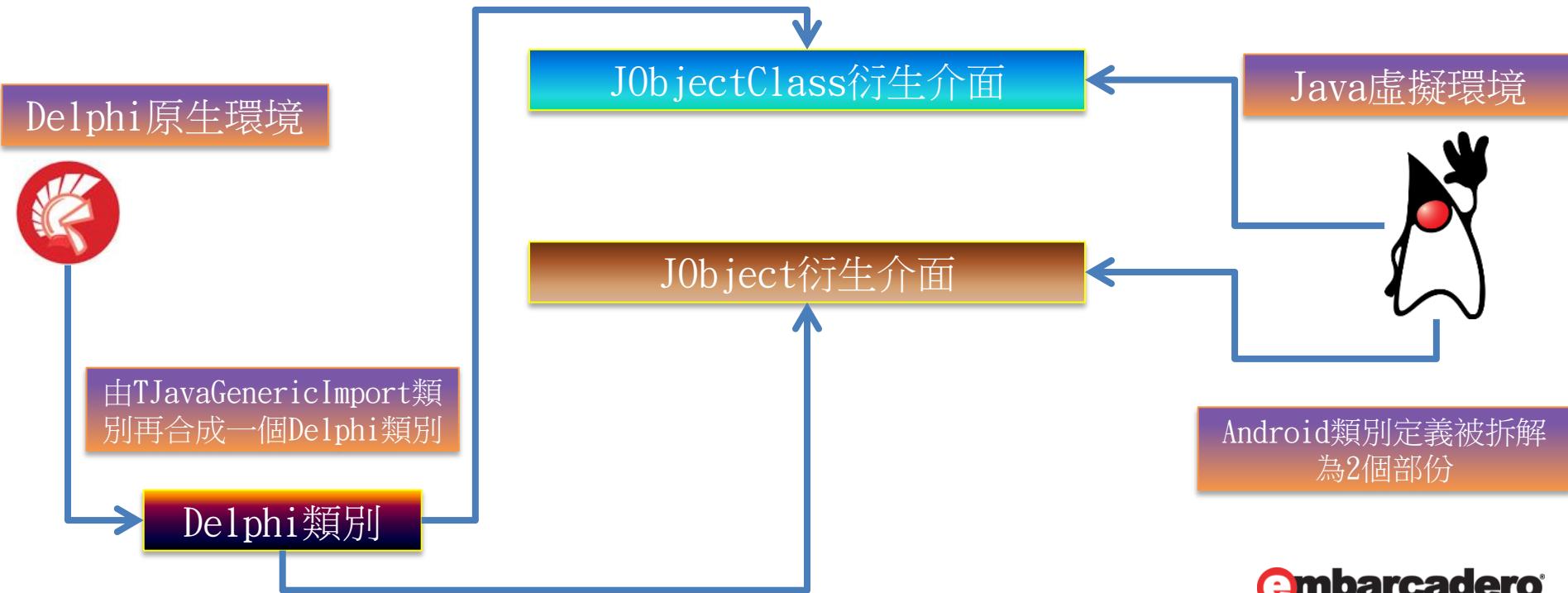
通往Android/iOS環境的重要觀念, 類別/介面和方法

全域物件, 類別	定義程式單元	說明
SharedActivity	FMX.Helpers.Android	Delphi的JNI Activity, 存取Android環境的橋樑
TJavaGenericImport	Androidapi.JNIBridge	可封裝Java物件成Delphi物件, 稍後有更多的說明
MainActivity	FMX.Platform.Android	FireMonkey Activity
SharedActivityContext	FMX.Helpers.Android	和SharedActivity一樣, 但提供不同的介面服務

通往Android/iOS環境的重要觀念, 類別/介面和方法



通往Android/iOS環境的重要觀念, 類別/介面和方法



通往Android/iOS環境的重要觀念, 類別/介面和方法

- 範例1
 - Android Vibrate
 - <http://developer.android.com/reference/android/content/Context.html>

通往Android/iOS環境的重要觀念, 類別/介面和方法

- 範例1
 - <http://developer.android.com/reference/android/os/Vibrator.html>

定義	方法	說明
abstract void	cancel()	Turn the vibrator off.
abstract boolean	hasVibrator()	Check whether the hardware has a vibrator.
abstract void	vibrate(long[] pattern, int repeat)	Vibrate with a given pattern.
abstract void	vibrate(long milliseconds)	Vibrate constantly for the specified period of time.

通往Android/iOS環境的重要觀念, 類別/介面和方法

Delphi原生環境



```
TJVibrator =  
class(TJavaGenericImpo  
rt<JVibratorClass,  
JVibrator>) end;
```

Delphi類別

JVibratorClass
=interface(JObjectClass)

```
[JavaSignature(' android/os/Vibrator' )]  
JVibrator = interface(JObject)  
[ '{82BDC8BC-22A3-4EAD-99AF-5DA70739B086}' ]  
{Methods}  
procedure cancel; cdecl;  
function hasVibrator: Boolean; cdecl;  
procedure vibrate(milliseconds: Int64);  
cdecl; overload;  
procedure vibrate(pattern:  
TJavaArray<Int64>; repeat_: Integer);  
cdecl; overload;  
end;
```

Java虛擬環境



Android類別定義被拆解
為2個部份

通往Android/iOS環境的重要觀念, 類別/介面和方法

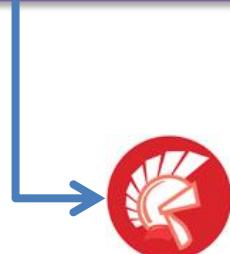
全域物件, 類別	定義程式單元	說明
TJavaGenericImport	Androidapi.JNIBridge	合成Java成Delphi的類別
ILocalObject	Androidapi.JNIBridge	遠端和本地Java物件實作的 介面
JObjectClass	Androidapi.JNI.JavaTypes	代表Java類別的類別
JObject	Androidapi.JNI.JavaTypes	代表Java物件的類別

通往Android/iOS環境的重要觀念, 類別/介面和方法

Delphi原生環境



Delphi XE5 App

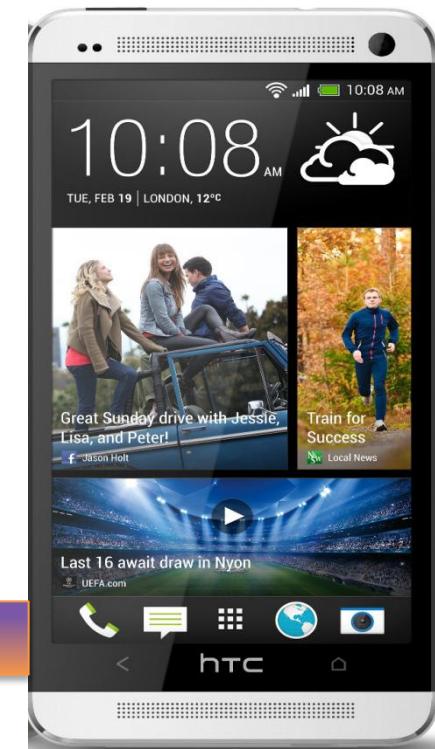


如何呼叫?

I. SharedActivit
y
getSystemService
e



Java Vibrat
e物件



Java虛擬環境

embarcadero

通往Android/iOS環境的重要觀念, 類別/介面和方法

- TJavaGenericImport<C: IJavaClass; T: IJavaInstance> = class(TJavaImport)
 - strict private
 - class var FInstanceVTable: TJavaVtable;
 - class var FClassVTable: TJavaVTable;
 - class var FJavaClass: C;
 - class var FClsID: Pointer;
 - class function GetJavaClass: C; static;
 - class function GetInstanceVTable(ClsID: Pointer; ObjID: Pointer): TJavaVTable;
 - class function Alloc(var ObjID: Pointer): T; overload;
 - public
 - class function Wrap(P: Pointer): T;
 - ///
 - /// Provides the Java class object for this type. You can use the
 - /// resulting interface to call Java class methods, rather than
 - /// instance methods.
 - class property JavaClass: C read GetJavaClass;

通往Android/iOS環境的重要觀念, 類別/介面和方法

```
///  
/// Both import objects and local objects implement this interface. We use it  
/// as a common means of getting to key data when dealing with marshaling  
parameters.  
ILocalObject = interface(IJava)  
  [IID_ILocalObject_Name]  
    /// Returns the Java instance ID of this object.  
    function GetObjectID: Pointer;  
end;
```

通往Android/iOS環境的重要觀念, 類別/介面和方法

```
public abstract Object getSystemService (String name)
```

```
function getSystemService(name: JString): JObject; cdecl;
```

```
class function Wrap(P: Pointer): T;
```

```
Wrap((VibratorObj as ILocalObject).GetObjectID);
```

通往Android/iOS環境的重要觀念, 類別/介面和方法

- 範例1
 - Android Vibrate

Delphi XE5未封裝的API?

- 以Android Environment為範例
 - <http://developer.android.com/reference/android/os/Environment.html>

Delphi XE5未封裝的API?

- 使用
 - JObjectClass
 - JObject
 - TJavaGenericImport
- 3個類別, 加上
 - GUID
 - JavaSignature屬性

Delphi XE5未封裝的API?

- 範例2
 - <http://developer.android.com/reference/android/os/Environment.html>
 - Android Environment

Delphi XE5未封裝的API?

- 範例3
 - [http://developer.android.com/reference/android/bluetooth
h/BluetoothDevice.html](http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html)
 - Android Bluetooth

中場休息

動態呼叫? 靜態呼叫?

- 靜態呼叫
 - 上面討論的呼叫方法可視為用Delphi靜態呼叫Java(Object-C)的方法,
 - 即直接用Delphi語法呼叫Java(Object-C)
- 動態呼叫
 - 動態載入Java使用Method signature呼叫

如何呼叫您的Java程式碼

- 觀察您的XE5 Android App
 - Classes.dex
 - Dexdump classes.dex > xe5debugdexdump.txt
- 封裝您的Java程式碼到APK中
- 觀察APK中的classes.dex
- 再使用動態呼叫
- <http://wwwpclviewer.com/android/androidJNI.html>

如何呼叫您的Java程式碼

- 動態呼叫步驟

打包你的Java程式碼成APK

使用Winzip/Winrar取出其中的
classes.dex

使用**DEXDUMP**觀察你的Java
函式的Prototype

使用XE5的Deployment
Wizard部署你的Java APK

使用TPath找到你部署的APK

使用Delphi程式碼動態載入
Java類別和呼叫

如何呼叫您的Java程式碼

- 動態呼叫程式碼

全域物件, 類別	定義程式單元
SharedActivityContext	FMX.Helpers.Android
TJDexClassLoader	Androidapi.JNI.Dalvik
TJNIResolver	Androidapi.JNIBridge
JNINativeInterface_	Androidapi.Jni

如何呼叫您的Java程式碼

- 動態呼叫程式碼

全域物件, 類別	說明
Androidapi.JNI.JavaTypes	Java型態定義
Androidapi.Jni	JNI型態, 介面定義

如何呼叫您的Java程式碼

使用TJDexClassLoader動態
載入你的APK

使用TJDexClassLoader動態
載入你的Java Class

使用TJNIResolver取得你的
Java方法的原型定義

使用Androidapi.JNI.JavaTypes和
Androidapi.Jni中的型態定義和傳入參數
值

使用JNINativeInterface呼叫你
的Java方法

Delphi XE5未封裝的API?

- 範例4
 - pDelphiCallJava

瞭解您的Android App!

- AndroidManifest.template.xml
- AndroidManifest.xml

瞭解您的Android App!

```
<activity android:name="com.embarcadero.firemonkey.FMXNativeActivity"  
        android:label="pEnvironmentDemo"  
        android:configChanges="orientation|keyboardHidden">      <!-- Tell NativeActivity the name  
        of our .so -->    <meta-data android:name="android.app.lib_name"  
        android:value="pEnvironmentDemo" />    <intent-filter>          <action  
        android:name="android.intent.action.MAIN" />          <category  
        android:name="android.intent.category.LAUNCHER" />    </intent-filter>    </activity>  
<receiver android:name="com.embarcadero.firemonkey.notifications.FMXNotificationAlarm" />
```

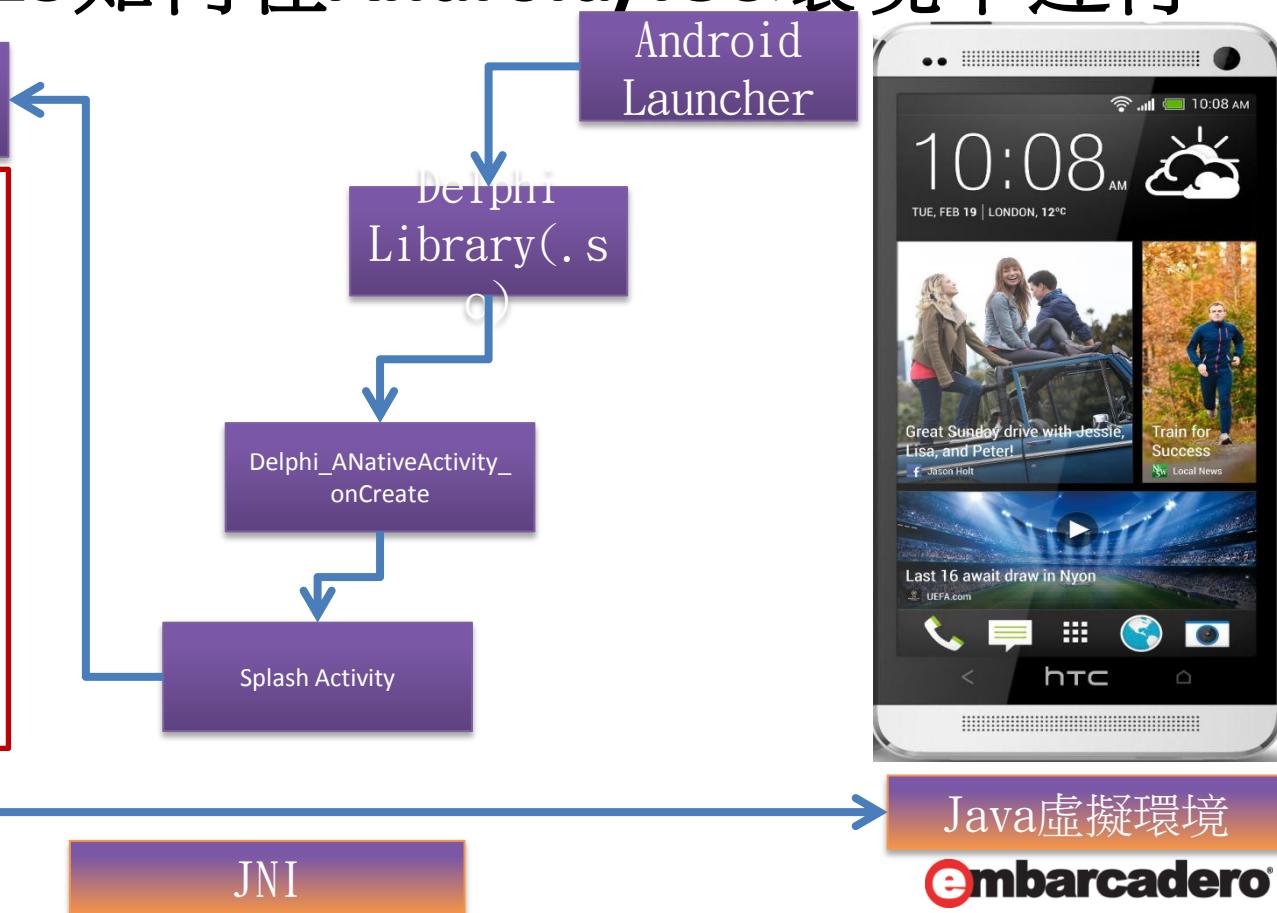
瞭解Delphi XE5如何在Android/iOS環境中運行

com/embcadero/firemonkey/FMXNativeActivity



Delphi XE5 App

Delphi原生環境



XE5和BarCode

- <https://code.google.com/p/zxing/>
- <http://blog.naver.com/simonsayz>
- <https://code.google.com/p/zxing/wiki/ScanningViaIntent>
- <http://edn.embarcadero.com/article/43269>
- <http://blog.blong.com/>
- <http://wwwpclviewer.com/android/androidJNI.html>

XE5和BarCode

- 現在(XE5)在iOS中做BarCode很簡單(因為都是Native),但在Android中比較麻煩
- 因為...

XE5和BarCode

```
{  
    Intent intent = new Intent("com.google.zxing.client.android.SCAN");  
    intent.putExtra("SCAN_MODE", "QR_CODE_MODE");  
    startActivityForResult(intent, 0);  
}  
  
public void onActivityResult(int requestCode, int resultCode, Intent intent) {  
    if (requestCode == 0) {  
        if (resultCode == RESULT_OK) {  
            String contents = intent.getStringExtra("SCAN_RESULT");  
            String format = intent.getStringExtra("SCAN_RESULT_FORMAT");  
            // Handle successful scan  
            Toast toast = Toast.makeText(this, "Content:" + contents + " Format:" + format , Toast.LENGTH_LONG);  
            toast.setGravity(Gravity.TOP, 25, 400);  
            toast.show();  
        } else if (resultCode == RESULT_CANCELED) {  
            // Handle cancel  
            Toast toast = Toast.makeText(this, "Scan was Cancelled!", Toast.LENGTH_LONG);  
            toast.setGravity(Gravity.TOP, 25, 400);  
            toast.show();  
        }  
    }  
}
```

XE5和BarCode

- `JActivity = interface(JContextThemeWrapper)`
 - ...
 - `procedure startActivities(intents: TJavaObjectArray<JIntent>); cdecl;`
`overload;`
 - `procedure startActivities(intents: TJavaObjectArray<JIntent>; options:`
`JBundle); cdecl; overload;`
 - `procedure startActivity(intent: JIntent); cdecl; overload;`
 - `procedure startActivity(intent: JIntent; options: JBundle); cdecl;`
`overload;`
 - `procedure startActivityForResult(intent: JIntent; requestCode: Integer);`
`cdecl; overload;`
 - `procedure startActivityForResult(intent: JIntent; requestCode: Integer;`
`options: JBundle); cdecl; overload;`

XE5和BarCode

- procedure TForm1.Button1Click(Sender: TObject);
- var
- anIntent : JIntent;
- begin
- anIntent := TJIntent.Create;
- anIntent.putExtra('SCAN_MODE', 'QR_CODE_MODE');
- SharedActivity.startActivityForResult(anIntent, 0);
- end;

瞭解Delphi XE5如何在Android/iOS環境中運行

com/embarcadero/firemonkey/FMXNativeActivity



Delphi XE5 App

Delphi原生環境

實作繼承類別並override
onActivityResult

取出

呼叫

在你的FireMonkey Form中
動態加入onActivityResult
方法

打包到classes.dex中

部署新的classes.dex

修改AndroidManifest.xml

Delphi XE5未封裝的API?

- 範例5
 - BarCode Test

Delphi XE5和iOS

- 最後再印證一下iOS
- https://developer.apple.com/library/ios/documentation/Cocoa/Reference/Foundation/Classes/NSNumber_Class/Reference/Reference.html#/apple_ref/occ/clm/NSNumber/numberWithBool:

Delphi XE5和iOS

- NSNumberClass = interface(NSValueClass)
 - ['{82392D79-32AF-4380-9389-667299F94C07}']
 - {class} function numberWithBool(value: Boolean): Pointer; cdecl;
 - ...
- End;

- NSNumber = interface(NSValue)
 - ['{547076CE-9409-4389-B449-47DA1D9718EF}']
 - function boolValue: Boolean; cdecl;
 - ...
- End;

- TNSNumber = class(TOCGenericImport<NSNumberClass, NSNumber>) end;

Delphi XE5和iOS

- function SharedApplication: UIApplication;
- begin
- Result :=
 UIApplication.Wrap(UIApplication.OCClass.shared
 Application);
- end;

Delphi XE5和iOS

- 範例6
 - piOSNumberDemo

```
procedure TForm2.Button1Click(Sender: TObject);
var
  nsNumb: nsNumber;
  nsNFor: NSNumberFormatter;
  nsStr: NSString;
begin
  nsnumb := TNSNumber.Wrap(TNSNumber.OCClass.numberWithInteger(12345)); // Length
  = 5;
  nsNFor := TNSNumberFormatter.Create;
  nsStr := nsNFor.stringFromNumber(nsnumb);

  ShowMessage(NSStrToStr(nsStr));
end;
```

Delphi For Android/iOS平台會持續發展

- 封裝更多的Android/iOS API
- Delphi/C++更容易的呼叫Java/Objective-C
- 更多針對移動平台和穿戴式設備的功能



謝謝參加