

qinjuning、lets go


编程珠玑 JVM Android爱好者QQ群: 55945620

目录视图

摘要视图

RSS 订阅

个人资料



qinjuning

访问: 1416763次

积分: 9419

等级: BLOG 5

排名: 第884名

原创: 43篇 转载: 0篇

译文: 3篇 评论: 1026条


通知

本文博客欢迎转载, 请保留出处。谢谢合作

武汉Android联盟QQ群

Android爱好者QQ群: 55945620, 希望热爱武汉Android同胞能够加入。


博客专栏



Android框架浅析

文章: 10篇

阅读: 593426



Android技巧拾取

文章: 17篇

阅读: 490589

文章分类

Andoird技巧拾取 (27)

Andoird框架浅析 (12)

Java技巧拾取 (3)

Linux学习笔记 (1)

总结:难得糊涂 (3)

专家和你聊Web, 速来报名

微信开发学习路线高级篇上线

免费公开课平台正式上线啦

恭喜July新书上市

## JNI学习积累之二 ---- 数据类型映射、域描述符说明

分类: Java技巧拾取

2012-05-25 22:31

14095人阅读

评论(2)

收藏

举报

jni

primitive

reference

types

string

java

目录(?)

[+]

本文原创, 转载请注明出处: <http://blog.csdn.net/qinjuning>

在Java存在两种数据类型: 基本类型 和 引用类型, 大家都懂的。

在JNI的世界里也存在类似的数据类型, 与Java比较起来, 其范围更具严格性, 如下:

1、primitive types ----基本数据类型, 如: int、float、char等基本类型

2、reference types----引用类型, 如: 类、实例、数组。

特别需要注意: 数组 ----- 不管是对对象数组还是基本类型数组, 都作为reference types存在。

1、primitive types (基本数据类型)映射参见下表:

### JNI类型映射

Java类型	本地类型(Native Type)	描述
boolean	jboolean	C/C++无符号8位整型 (unsigned char)
byte	jbyte	C/C++带符号的8位整型 (char )
char	jchar	C/C++无符号的16位整型 (unsigned short)
short	jshort	C/C++带符号的16位整型 (short)
int	jint	C/C++带符号的32位整型 (int)
long	jlong	C/C++带符号的64位整型 (long)
float	jfloat	C/C++32位浮点型 (float)
double	jdouble	C/C++64位浮点型 (double)

这些基本数据类型都是可以在Native层直接使用的。

2、reference types (引用数据类型)映射参见下表

<http://blog.csdn.net/qinjuning/article/details/7599796>

1/5

阅读排行

Android中View绘制流程I

(157373)

Android中Context详解 --

(150847)

Android中获取应用程序(

(96211)

Android框架浅析之锁屏(I

(94181)

Android中Preference的使

(64947)

Android中滑屏初探 ---- s

(50035)

Android中将布局文件/Vie

(46719)

Android中滑屏实现 ----手:

(46337)

Andriod中绘(画)图 ----Car

(45347)

Android中获取正在运行

(43643)

评论排行

Android中View绘制流程I

(106)

Android中Context详解 --

(68)

Android中滑屏实现 ----手:

(64)

Android中获取应用程序(

(63)

毕业半年，点滴在心中

(62)

Android框架浅析之锁屏(I

(59)

Android中获取应用程序(

(52)

Android学习进阶路线导航

(47)

Android中将布局文件/Vie

(42)

Android中Preference的使

(41)

文章存档

2014年12月 (1)

2013年04月 (1)

2013年03月 (1)

2013年01月 (1)

2012年11月 (1)

展开

最新评论

Android中View绘制流程以及inva

woshimiaoqingren: 好人一生平安

Android中measure过程、WRAP

u010588886: 好厉害的感觉

Android中Context详解 ---- 你所不

jj1234588: thanks!

Android中获取应用程序(包)的信

happy\_horse: 真棒

Android中Preference的使用以及

zy000000001: 十分的到

Android中Context详解 ---- 你所不

qq\_29877767: @leehong2005:xxxactivity.this

Android中measure过程、WRAP

u010236416: 非常好

毕业半年，点滴在心中

qq\_18212135: 最近很迷茫，本

Android中获取应用程序(包)的信

清澈@Cherry: 楼主辛苦了，学习

Android中Context详解 ---- 你所不

l5741T: 谢谢分享! 学习了

Java类型	Native Type	描述
Object	jobject	任何Java对象，或者没有对应java类型的对象
Class	jclass	Class类对象
String	jstring	字符串对象
Object[]	jobjectArray	任何对象的数组
boolean[]	jbooleanArray	布尔型数组
byte[]	jbyteArray	比特型数组
char[]	jcharArray	字符型数组
short[]	jshortArray	短整型数组
int[]	jintArray	整型数组
long[]	jlongArray	长整型数组
float[]	jfloatArray	浮点型数组
double[]	jdoubleArray	双浮点型数组
特殊类型： void	void	n/a

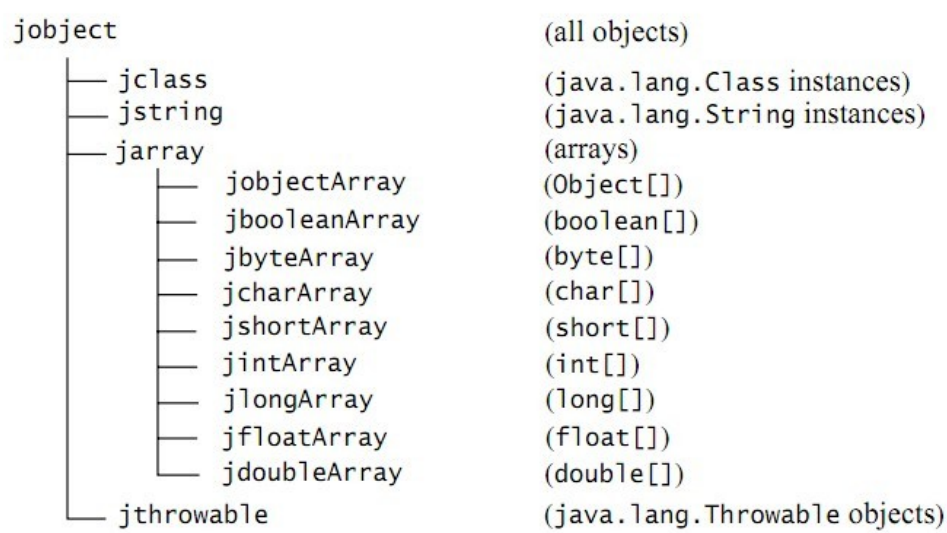
- 注意：
- 1、引用数据类型则不能直接使用，需要根据JNI函数进行相应的转换后，才能使用

2、多维数组(包括二维数组)都是引用类型，需要使用 **jobjectArray** 类型存取其值；

例如：二维整型数组就是指向一位数组的数组，其声明使用方式如下：

```
[java] view plain copy print ?
01. //获得一维数组 的类引用，即jintArray类型
02. jclass intArrayClass = env->FindClass("[I");
03. //构造一个指向jintArray类一维数组的对象数组，该对象数组初始大小为dimion
04. jobjectArray obejctIntArray = env->NewObjectArray(dimion ,intArrayClass , NULL);
05. ...//具体操作
```

另外，关于引用类型的一个继承关系如下，我们可以对具有父子关系的类型进行转换：



类描述符

类描述符是类的完整名称（包名+类名），将原来的 . 分隔符换成 / 分隔符。

例如：在java代码中的java.lang.String类的类描述符就是java/lang/String

其实，在实践中，我发现可以直接用该类型的域描述符取代，也是可以成功的。

```
例如：    jclass intArrCls = env->FindClass("java/lang/String")
          等同于    jclass intArrCls = env->FindClass("Ljava/lang/String;")
```

数组类型的描述符则为，则为： [ + 其类型的域描述符 （后文说明）

```
例如：
int [ ]  其描述符为[I
float [ ] 其描述符为[F
String [ ] 其描述符为[Ljava/lang/String;
```

域描述符

1、基本类型的描述符已经被定义好了，如下表所示：

Field Descriptor	Java Language Type
Z	boolean
B	byte
C	char
S	short
I	int
J	long
F	float
D	double

2、引用类型的描述符

一般引用类型则为 L + 该类型类描述符 + ; （注意，这儿的分号“;”只得是JNI的一部分，而不是我们汉语中的分段，下同）

```
例如：String类型的域描述符为 Ljava/lang/String;
```

对于数组，其为： [ + 其类型的域描述符 + ;

```
int[ ]  其描述符为[I
float[ ] 其描述符为[F
String[ ] 其描述符为[Ljava/lang/String;
Object[ ]类型的域描述符为[Ljava/lang/Object;
```

多维数组则是 n个[ +该类型的域描述符 ,N代表的是几维数组。例如：

```
int [ ][ ] 其描述符为[[I
float[ ][ ] 其描述符为[[F
```

方法描述符

将参数类型的域描述符按照申明顺序放入一对括号中后跟返回值类型的域描述符，规则如下：(参数的域描述符的叠加)返回

类型描述符。对于，没有返回值的，用V(表示void型)表示。举例如下：

Java层方法	JNI函数签名
String test ( )	Ljava/lang/String;

```
int f (int i, Object object)          (ILjava/lang/Object;)I
void set (byte[ ] bytes)              ([B)V
```

在编程时，如果具有的话，这些都不需要我们手动编写对应的类型转换，如果不能使用javah工具，就只能进行类型转换了。

下一篇将会以一个具体的例子来具体掌握JNI。

分析一个使用Android NDK 开发的系列博客，觉得挺不错的：

《 Android的NDK开发(3)—— JNI数据类型的详解》

版权声明：本文为博主原创文章，未经博主允许不得转载。

- 上一篇
- JNI学习积累之一 ---- 常用函数大全
- 下一篇
- JNI学习积累之三 ---- 操作JNI函数以及复杂对象传递

顶

4

踩

0

主题推荐

数据类型

jni

微软

猜你在找

- Android必备的Java基础知识 (三)
- javablibrarypath在哪
- Android底层技术：Java层系统服务(Android Service)
- android NDK的androidmk文件的详解
- Android必备的Java基础知识
- Android漫游记2---ELF可执行文件格式
- Java基础核心技术：面向对象编程(day05-day07)
- PCM数据格式
- Android入门实战教程
- Linux shell 下处理 二进制



查看评论

2楼 yfsyy 2015-06-01 10:59发表



引用“hkllkh009”的评论：

方法描述符

将参数类型的域描述符按照申明顺序放入一对括号中后跟返回值类型的域描述符， ...

楼上是正确的。

对于数组，其为：[ + 其类型的域描述符 + ；  
int[ ] 其描述符为[I  
float[ ] 其描述符为[F  
String[ ] 其描述符为[Ljava/lang/String;