|  |  |  |
| --- | --- | --- |
| **函数** | **Java 数组类型** | **本地类型** |
| GetBooleanArrayElements | jbooleanArray | jboolean |
| GetByteArrayElements | jbyteArray | jbyte |
| GetCharArrayElements | jcharArray | jchar |
| GetShortArrayElements | jshortArray | jshort |
| GetIntArrayElements | jintArray | jint |
| GetLongArrayElements | jlongArray | jlong |
| GetFloatArrayElements | jfloatArray | jfloat |
| GetDoubleArrayElements | jdoubleArray | jdouble |

在Java存在两种数据类型： 基本类型 和 引用类型 ，大家都懂的 。

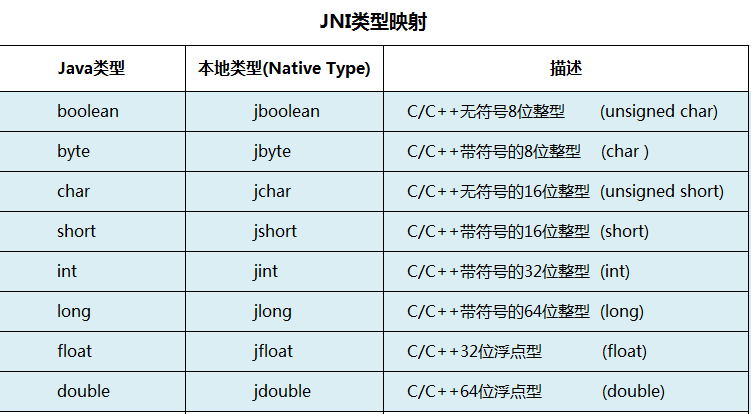
    在JNI的世界里也存在类似的数据类型，与Java比较起来，其范围更具严格性，如下：

        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：

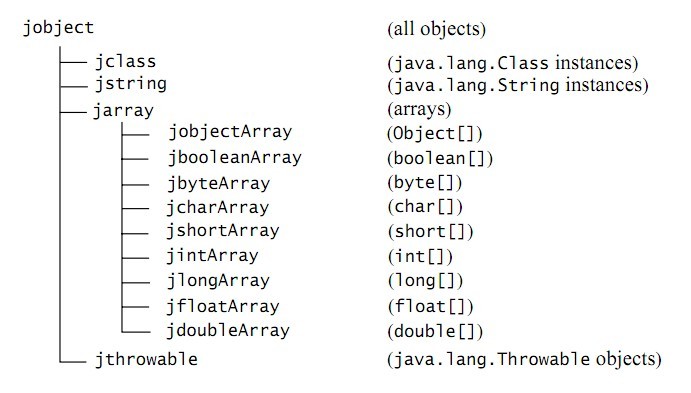


**这些基本数据类型都是可以在Native层直接使用的**

 2、**reference types** (引用数据类型)映射参见下表）



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：



**数组类型**的描述符则为，则为：  [ + 其类型的域描述符        (后文说明)

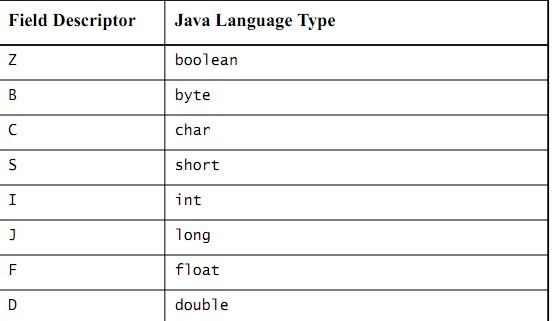
           例如：int [ ]     其描述符为[I

                  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：



**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，如果是利用javah工具的话，这些都不需要我们手动编写对应的类型转换，如果不能用javah工具，就只能手动的

  进行类型转换了。

1、void

java的void与JNI的void是一致的。

2、基本数据类型



3、对象类型





4、数组类型



