**[JNI 数据类型转换](http://dolphin-ygj.iteye.com/blog/519492)**

**博客分类：**

* [java综合](http://dolphin-ygj.iteye.com/category/11710)

[JNI](http://www.iteye.com/blogs/tag/JNI)[Java](http://www.iteye.com/blogs/tag/Java)[C](http://www.iteye.com/blogs/tag/C)[C++](http://www.iteye.com/blogs/tag/C++)[C#](http://www.iteye.com/blogs/tag/C%23)

Java代码 [复制代码](http://andilyliao.iteye.com/blog/494255)

1. #include <jni.h>
2. #include "com\_test\_Test.h"
3. #include <stdio.h>
4. #include <stdlib.h>
5. #include <strings.h>
7. //获取字符串
8. JNIEXPORT **void** JNICALL Java\_com\_test\_Test\_sayHello(JNIEnv \* env, jobject obj,
9. jstring s) {
11. **char** \* str = (**char** \*) (\*env)->GetStringUTFChars(env, s, 0);
12. printf("\n c-string: hello - %s", str);
14. }
16. //获取，返回int
17. JNIEXPORT jint JNICALL Java\_com\_test\_Test\_add(JNIEnv \* env, jobject obj,
18. jint a, jint b) {
19. **int** i = a;
20. **int** j = b;
21. printf("\n c-int: %d - %d", i, j);
22. **char** str[256];
23. sprintf(str, "%d", i + j);
24. **return** (jint) i + j;
25. }
27. //获取，返回float
28. JNIEXPORT jfloat JNICALL Java\_com\_test\_Test\_getFloat(JNIEnv \* env, jobject obj,
29. jfloat f) {
30. **float** fl = f;
31. printf("\n c-float: %3f", fl);
33. fl = 200.555;
34. **return** (jfloat) fl;
35. }
37. //获取，返回 double
38. JNIEXPORT jdouble JNICALL Java\_com\_test\_Test\_getDouble(JNIEnv \* env,
39. jobject obj, jdouble dou) {
40. **double** d = dou;
41. printf("\n c-double: %3f", d);
43. d = 800.88;
44. **return** (jdouble) d;
45. }
47. //获取，返回boolean
48. JNIEXPORT jboolean JNICALL Java\_com\_test\_Test\_getBoolean(JNIEnv \* env,
49. jobject obj, jboolean bool) {
51. unsigned **char** b = bool;
52. printf("\n c-boolean: %lu  ", b);
53. **if** (b) {
54. printf("true");
55. } **else** {
56. printf("false");
57. }
59. b = 1;
60. **return** (jboolean) b;
61. }
63. //获取，返回 string
64. JNIEXPORT jstring JNICALL Java\_com\_test\_Test\_getString(JNIEnv \* env,
65. jobject obj, jstring s) {
66. **char** \* st = (**char** \*) (\*env)->GetStringUTFChars(env, s, 0);
67. printf("\n c-string:  %s", st);
69. **char** \* str = "hello wang wu!";
70. jstring rtn;
71. rtn = (\*env)->NewStringUTF(env, str);
72. **return** rtn;
73. }
75. //获取，返回 string []
76. JNIEXPORT jobjectArray JNICALL Java\_com\_test\_Test\_getStringArray(JNIEnv \* env,
77. jobject obj, jobjectArray arr) {
78. **int** len = (\*env)->GetArrayLength(env, arr);
79. printf("\n c-stringArray :");
80. **int** i = 0;
81. **for** (i = 0; i < len; i++) {
82. jobject obj = (\*env)->GetObjectArrayElement(env, arr, i);
83. jstring str = (jstring) obj;
84. **const** **char** \* szStr = (\*env)->GetStringUTFChars(env, str, 0);
85. printf(" %d-%s ", i, szStr);
86. (\*env)->ReleaseStringChars(env, str, szStr);
87. }
88. //-----返回----
90. jstring str;
91. jobjectArray args = 0;
92. jsize size = 5;
93. **char**\* sa[] = { "Hello,", "world!", "zhang", "san", "yuang" };
94. **int** j = 0;
95. jclass objClass = (\*env)->FindClass(env, "java/lang/String");
96. args = (\*env)->NewObjectArray(env, size, objClass, 0);
97. **for** (j = 0; j < size; j++) {
98. str = (\*env)->NewStringUTF(env, sa[j]);
99. (\*env)->SetObjectArrayElement(env, args, j, str);
100. }
101. **return** args;
102. }
104. -----------------------------------------------------------------
106. 下面是Test.java
108. **package** com.test;
110. **public** **class** Test {
111. **private** **native** **void** sayHello(String str);
113. **private** **native** **int** add(**int** a, **int** b);
115. **private** **native** **float** getFloat(**float** f);
117. **private** **native** **double** getDouble(**double** d);
119. **private** **native** **boolean** getBoolean(**boolean** b);
121. **private** **native** String getString(String str);
123. **private** **native** String[] getStringArray(String[] sa);
125. **static** {
126. System.loadLibrary("Test");
127. }
129. **public** **static** **void** main(String[] args) {
130. Test test = **new** Test();
131. test.sayHello("zhang san");
132. System.out.println("int     --> " + test.add(10, 20));
133. System.out.println("float   --> " + test.getFloat((**float**) 20.123));
134. System.out.println("double  --> " + test.getDouble(100.369));
135. System.out.println("boolean --> " + test.getBoolean(**true**));
136. System.out.println("string  --> " + test.getString("wang Wu"));
138. String[] ss = { "hello", "-", "zhang", "san" };
139. Object[] obj = test.getStringArray(ss);
140. System.out.print("string[]-->  ");
141. **for** (Object object : obj) {
142. System.out.print(object + " ");
143. }
145. }
147. }