



shunbinghou

专注、解决问题

博客园 :: 首页 :: 新随笔 :: 联系 :: 订阅 [XML](#) :: 管理

posts - 3, comments - 0, trackbacks - 0

## 公告

转载 **Java**中类**C/C++**的**sizeof()**操作, 知晓实例大小

在**C/C++**中有**sizeof()**操作, 可轻易获知某个类型或实例占用内存大小, **sizeof(int)** 或者 **sizeof(new TestClass)**。可是**Java**中可没有这么直观的方法可用。

因本人看过不少人写代码总爱写成

```
List userList = new ArrayList(); //注: 声明时即初始化一个空 ArrayList
userList = userDao.getAllUsers(); //注: 方法getAllUsers()中会生成一个ArrayList的
```

上面就造成平白多了一个空的 **ArrayList()**, 创建完后即刻就推向**GC**处置, 我就在想这样一个空的 **ArrayList** 会占用多少内存, 于是找来了 **Optimizeit** 观察后发现一个空的 **ArrayList** 要占去 **24b** 内存。那 **Java** 中有没有更便捷的方法呢, 于是在网上 **google "java sizeof"**, 引出不少话题。

但我觉得比较好的一段代码是 [java.sizeOf\(http://sourceforge.net/projects/sizeof/\)](http://sourceforge.net/projects/sizeof/), 需要**JDK1.5**以上的版本支持, 由它测定的空 **ArrayList** 所占内存确也是 **24b**。下载到 **SizeOf\_0\_2\_src.zip**(其中含**SizeOf.jar**, 其实就一个类 **SizeOf**), 假设解压到 **F:\Component Library\java\javasizeof**。

用法很简单, 直接看它的 **README.txt** 文件就行, 因为只有一个类, 方法也不多, 全是静态方法, 可以使用静态引入, 看看就明白, 只是现在的 **0.2** 版本推荐的方法是 **sizeOf()** 和 **deepSizeOf()**, 而不再是 **iterativeSizeOf()** 的。

它还提供了一个测试代码 **TestSizeOf** (可从中学习 **SizeOf** 的用法), 在命令行下, 进入到 **F:\Component Library\java\javasizeof** 目录, 然后执行

```
F:\Component Library\java\javasizeof>java -javaagent:SizeOf.jar
net.sourceforge.sizeof.test.TestSizeOf
```

显示结果是:

```
JAVAGENT: call premain instrumentation for class SizeOf
Starting test...
simple obj: 40.0b
```

```
int:      0.0b
long:     0.0b
char:     0.0b
double:   16.0b
boolean:  0.0b
Integer:  0.0b
empty string: 0.0b
not empty string: 0.0b
not empty string: 0.0b
simple obj: 24.0b
simple obj: 40.0b
empty list: 24.0b
10 list: 24.0b
20 list no static: 24.0b
1000 o arr: 816.0b
```

应该会惊讶一下，为什么会出现那么多 0.0b？为什么要用1.5以上版本的jdk,为什么不是用 -classpath 参数，而是 -javaagent 参数？

我们不妨把 javasizeof 的src目录中源代码导入到 eclipse中（相信大多数人都用这个的），可以看到源代码 TestSizeOf，把 SizeOf.skipFlyweightObject(true) 行注释掉，运行 TestSizeOf。

不小心的话，你应该收到 java.lang.IllegalStateException: Instrumentation is null 的异常，没错这个 SizeOf 用到了 JDK1.5 后新加入的 java.lang.instrument.Instrumentation 接口，所以您需要为 TestSizeOf 设置 VM arguments

**-javaagent:"F:\Component Library\java\jvasizeof\SizeOf.jar"**

（注意到在 SizeOf.jar包中的 META-INF\MANIFEST.MF中有 Premain-Class: net.sourceforge.sizeof.SizeOf，这就是 -javaagent: 所在意的。指向SizeOf.jar的路径中有空格的话一定要用双引号引起来）

运行后结果就是：

```
JAVAGENT: call premain instrumentation for class SizeOf
Starting test...
simple obj: 40.0b
int: 16.0b
long: 16.0b
char: 16.0b
double: 16.0b
boolean: 16.0b
Integer: 16.0b
empty string: 24.0b
not empty string: 24.0b
not empty string: 24.0b
simple obj: 24.0b
simple obj: 40.0b
empty list: 24.0b
10 list: 24.0b
20 list no static: 24.0b
```

1000 o arr: 816.0b

对于现在的输出值细细去体会吧，也许不符合您早已形成的较为稳固的想法，或许你发现确实存在出入。

SizeOf\_0\_2\_src.zip包中的 README.txt 中部分内容：

java.SizeOf is a simple java agent what can be used to calculate the memory size of java objects.

The agent is implemented with the java.lang.instrument introduced with java 5.

Here is a simple howto:

1) use the class SizeOf in your code to test the size of your java object:

```
//configuration steps
SizeOf.skipStaticField(true);
SizeOf.setMinSizeToLog(10);

//calculate object size
SizeOf.iterativeSizeOf(<your object>)
```

2) start the jvm with the argument: -javaagent:<path to>/SizeOf.jar

To avoid the dependencies of your code to SizeOf the best use of the agent is in conjunction with aspect.

posted on 2011-09-16 17:50 侯顺兵 阅读(579) 评论(0) 编辑 收藏

[刷新页面](#) [返回顶部](#)

(评论功能已被博主禁用)

找优秀程序员，就在博客园

[博客园首页](#) [博问](#) [新闻](#) [闪存](#) [程序员招聘](#) [知识库](#)

Powered by:



Copyright ©2013 侯顺兵