

首页 (http://www.hollischuang.com/) 爪哇 (http://www.hollischuang.com/archives/category/java) 框架 (http://www.hollischuang.com/archives/category/%e6%a1%86%e6%9e%b6) 综合应用

(http://www.hollischuang.com/archives/category/%e7%bb%bc%e5%90%88%e5%ba%94%e7%94%a8) 操作系统

(http://www.hollischuang.com/archives/category/%e6%93%8d%e4%bd%9c%e7%b3%bb%e7%bb%9f) 异常 (http://www.hollischuang.com/archives/category/debug)

其他 (http://www.hollischuang.com/archives/category/%e5%85%b6%e4%bb%96) Q

## Java命令学习系列(七)——javap (http://www.hollischuang.com/archives/1107)

2016-01-31 分类: Java (http://www.hollischuang.com/archives/category/java) 阅读(8952) 评论(1)

本站采用[知识共享署名-非商业性使用-相同方式共享 许可协议 (http://creativecommons.org/licenses/by-nc-sa/3.0/deed.zh)]进行许可,转载请在 正文明显处 注明原文地址

javap是jdk自带的一个工具,可以对代码<u>反编译 (http://www.hollischuang.com/archives/58)</u>,也可以查看java编译器生成的字节码。

一般情况下,很少有人使用javap对class文件进行反编译,因为有很多成熟的反编译工具可以使用,比如jad。但是,javap还可以查看 java编译器为我们生成的字节码。通过它,可以对照源代码和字节码,从而了解很多编译器内部的工作。

#### 实例

javap命令分解一个class文件,它根据options来决定到底输出什么。如果没有使用options,那么javap将会输出包,类里的protected和 public域以及类里的所有方法。 javap 将会把它们输出在标准输出上。来看这个例子,先编译( javac )下面这个类。

```
import java.awt.*;
import java.applet.*;

public class DocFooter extends Applet {
    String date;
    String email;

    public void init() {
        resize(500,100);
        date = getParameter("LAST_UPDATED");
        email = getParameter("EMAIL");
    }

    public void paint(Graphics g) {
        g.drawString(date + " by ",100, 15);
        g.drawString(email,290,15);
    }
}
```

在命令行上键入javap DocFooter后,输出结果如下

```
Compiled from "DocFooter.java"
public class DocFooter extends java.applet.Applet {
    java.lang.String date;
    java.lang.String email;
    public DocFooter();
    public void init();
    public void paint(java.awt.Graphics);
}
```

查看 2 条评论

Q

```
Compiled from "DocFooter.java"
public class DocFooter extends java.applet.Applet {
 java.lang.String date;
 java.lang.String email;
 public DocFooter();
   Code:
      0: aload_0
      1: invokespecial #1
                                           // Method java/applet/Applet."<init>":()V
      4: return
 public void init();
   Code:
      0: aload_0
      1: sipush
                        500
      4: bipush
                       100
      6: invokevirtual #2
                                            // Method resize:(II)V
      9: aload_0
     10: aload_0
     11: ldc
                                            // String LAST_UPDATED
     13: invokevirtual #4
                                            // Method getParameter:(Ljava/lang/String;)Ljava/lang/String;
     16: putfield
                                            // Field date:Ljava/lang/String;
     19: aload 0
     20: aload_0
                                            // String EMAIL
     21: ldc
                       #6
                                            // Method getParameter:(Ljava/lang/String;)Ljava/lang/String;
     23: invokevirtual #4
     26: putfield
                                            // Field email:Ljava/lang/String;
     29: return
 public void paint(java.awt.Graphics);
   Code:
      0: aload_1
                       #8
                                            // class java/lang/StringBuilder
      1: new
      4: dup
                                            // Method java/Lang/StringBuilder."<init>":()V
      5: invokespecial #9
      8: aload_0
      9: getfield
                                           // Field date:Ljava/lang/String;
     12: invokevirtual #10
                                            // Method java/Lang/StringBuilder.append:(Ljava/Lang/String;)Ljava/Lang/StringBuilder;
                                            // String by
     15: ldc
                        #11
     17: invokevirtual #10
                                            // Method java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
                                            // Method java/Lang/StringBuilder.toString:()Ljava/Lang/String;
     20: invokevirtual #12
     23: bipush
                       100
     25: bipush
                       15
     27: invokevirtual #13
                                            // Method java/awt/Graphics.drawString:(Ljava/Lang/String;II)V
     30: aload_1
     31: aload_0
     32: getfield
                        #7
                                            // Field email:Ljava/lang/String;
     35: sipush
                       290
     38: bipush
                       15
     40: invokevirtual #13
                                            // Method java/awt/Graphics.drawString:(Ljava/Lang/String;II)V
     43: return
```

上面输出的内容就是字节码。

### 用法摘要

```
-help 帮助
-l 输出行和变量的表
-public 只输出public方法和域
-protected 只输出public和protected类和成员
-package 只输出包,public和protected类和成员,这是默认的
-p -private 输出所有类和成员
-s 输出内部类型签名
-c 输出分解后的代码,例如,类中每一个方法内,包含java字节码的指令,
-verbose 输出栈大小,方法参数的个数
-constants 输出静态final常量
```

## 总结

javap可以用于反编译和查看编译器编译后的字节码。平时一般用 javap -c 比较多,该命令用于列出每个方法所执行的JVM指令,并显示每个方法的字节码的实际作用。可以通过字节码和源代码的对比,深入分析java的编译原理,了解和解决各种Java原理级别的问题。

【公告】版权声明 (http://www.hollischuang.com/转载说明)

(全文完)



#### 欢迎关注HollisChuang微信公众账号

如未加特殊说明,此网站文章均为原创,转载必须注明出处。HollisChuang's Blog (http://www.hollischuang.com) » Java命令学习系列(七)——javap (http://www.hollischuang.com/archives/1107)

标签: Java命令学习系列 (http://www.hollischuang.com/archives/tag/java%e5%91%bd%e4%bb%a4%e5%ad%a6%e4%b9%a0%e7%b3%bb%e5%88%97)

分享到: 更多 (<sup>3</sup>)

## 相关推荐

- Java开发必须掌握的线上问题排查命令 (http://www.hollischuang.com/archives/1561)
- Java命令学习系列 ( 六 ) ——jinfo (http://www.hollischuang.com/archives/1094)
- Java命令学习系列 ( 五 ) ——jhat (http://www.hollischuang.com/archives/1047)
- Java命令学习系列(四)——jstat (http://www.hollischuang.com/archives/481)
- Java命令学习系列 ( 三 ) ——Jmap (http://www.hollischuang.com/archives/303)
- Java命令学习系列 (零) ——常见命令及Java Dump介绍 (http://www.hollischuang.com/archives/308)
- Java命令学习系列 ( 二 ) ——Jstack (http://www.hollischuang.com/archives/110)
- Java命令学习系列 ( ) ——Jps (http://www.hollischuang.com/archives/105)

登录

来说两句吧...

评论 **2** 人参与, **2** 条评论

最新评论



代号而己

2017年6月4日 15:24

可以问一下在IDEA里配置javap的后,右键执行时总是报"找不到类"的错误,大概是什么原因?

回复 3



偲

2016年2月4日 17:46

_	恒
1111	ᅜ

HollisChuang正在使用畅言 (http://changyan.kuaizhan.com/)

# HollisChuang's Blog

联系我 (http://mail.qq.com/cgi-bin/qm\_share?t=qm\_mailme&email=-JSTkJCVj5\_UiZ2Sm7yNjdKfk5E)

关于我 (/sample-page)

© 2018 HollisChuang's Blog (http://www.hollischuang.com) 网站地图 (http://www.hollischuang.com/sitemap.xml)|阿里云 (http://www.aliyun.com)|Alibaba.com (http://www.alibaba.com)| [thez ] (http://www.cnzz.com/stat/website.php?web\_id=1256611093)