NAME

javap − Disassembles one or more class files.

SYNOPSIS

javap [*options*] *classfile*...

options

The command−line options. See Options.

classfile

One or more classes separated by spaces to be processed for annotations such as DocFooter.class. You can specify a class that can be found in the class path, by its file name or with a URL such as **file:///home/user/myproject/src/DocFooter.class**.

DESCRIPTION

The **javap** command disassembles one or more class files. The output depends on the options used. When no options are used, then the **javap** command prints the package, protected and public fields, and methods of the classes passed to it. The **javap** command prints its output to **stdout**.

OPTIONS

−help
−−help
−?

Prints a help message for the **javap** command.

−version

Prints release information.

−l

Prints line and local variable tables.

−public

Shows only public classes and members.

−protected

Shows only protected and public classes and members.

−private
−p

Shows all classes and members.

−J*option*

Passes the specified option to the JVM. For example:

```
javap −J−version
javap −J−Djava.security.manager −J−Djava.security.policy=MyPolicy MyClassName
```

For more information about JVM options, see the command documentation.

−s

Prints internal type signatures.

−sysinfo

Shows system information (path, size, date, MD5 hash) of the class being processed.

−constants

Shows **static final** constants.

−c

Prints disassembled code, for example, the instructions that comprise the Java bytecodes, for each of the methods in the class.

−verbose

Prints stack size, number of locals and arguments for methods.

–classpath *path*
Specifies the path the **javap** command uses to look up classes. Overrides the default or the **CLASSPATH** environment variable when it is set.

–bootclasspath *path*
Specifies the path from which to load bootstrap classes. By default, the bootstrap classes are the classes that implement the core Java platform located in **jre/lib/rt.jar** and several other JAR files.

–extdir *dirs*
Overrides the location at which installed extensions are searched for. The default location for extensions is the value of **java.ext.dirs**.

EXAMPLE

Compile the following **DocFooter** class:

```
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);
    }
}
```

The output from the **javap DocFooter.class** command yields the following:

```
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);
}
```

The output from **javap –c DocFooter.class** command yields the following:

```
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           #3                  // String LAST_UPDATED
      13: invokevirtual #4                  // Method
 getParameter:(Ljava/lang/String;)Ljava/lang/String;
      16: putfield      #5                  // Field date:Ljava/lang/String;
      19: aload_0
      20: aload_0
      21: ldc           #6                  // String EMAIL
      23: invokevirtual #4                  // Method
 getParameter:(Ljava/lang/String;)Ljava/lang/String;
      26: putfield      #7                  // Field email:Ljava/lang/String;
      29: return

    public void paint(java.awt.Graphics);
      Code:
       0: aload_1
       1: new           #8                  // class java/lang/StringBuilder
       4: dup
       5: invokespecial #9                  // Method
 java/lang/StringBuilder."<init>":()V
       8: aload_0
       9: getfield      #5                  // Field date:Ljava/lang/String;
      12: invokevirtual #10                 // Method
 java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
      15: ldc           #11                 // String  by
      17: invokevirtual #10                 // Method
 java/lang/StringBuilder.append:(Ljava/lang/String;)Ljava/lang/StringBuilder;
      20: invokevirtual #12                 // Method
 java/lang/StringBuilder.toString:()Ljava/lang/String;
      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
}
```

SEE ALSO

- java(1)
- javac(1)
- javadoc(1)
- jdb(1)
- jdeps(1)