

```
1: // $Id: jcat.java,v 1.1 2015-12-11 17:06:45-08 - - $
2: //
3: // SYNOPSIS
4: //     jcat [filename...]
5: //
6: // DESCRIPTION
7: //     The jcat utility functions like cat(1) and copies the contents
8: //     of all files to the standard output, with error messages to
9: //     the standard error.
10: //
11: // EXIT STATUS
12: //     0 if no errors were detected.
13: //     1 if errors were detected and messages printed.
14: //
15:
16: import java.io.*;
17: import java.util.Scanner;
18: import static java.lang.System.*;
19:
20: class jcat {
21:     // Static variables keeping the general status of the program.
22:     public static final String JARNAME = get_jarname ();
23:     public static final int EXIT_SUCCESS = 0;
24:     public static final int EXIT_FAILURE = 1;
25:     public static int exit_status = EXIT_SUCCESS;
26:
27:     // A basename is the final component of a pathname.
28:     // If a java program is run from a jar, the classpath is the
29:     // pathname of the jar.
30:     static String get_jarname () {
31:         String jarpath = getProperty ("java.class.path");
32:         int lastslash = jarpath.lastIndexOf ('/');
33:         if (lastslash < 0) return jarpath;
34:         return jarpath.substring (lastslash + 1);
35:     }
36:
```

```
37:
38: // Copies a single opened file to stdout.
39: static void copylines (Scanner infile) {
40:     // Read each line from the opened file, one after the other.
41:     // Stop the loop at end of file.
42:     while (infile.hasNextLine ()) {
43:         String line = infile.nextLine ();
44:         out.printf ("%s\n", line);
45:     }
46: }
47:
48: // Open input file and copy contents to stdout.
49: static void catfile (String filename) {
50:     if (filename.equals ("-")) {
51:         copylines (new Scanner (System.in));
52:     }else {
53:         try {
54:             Scanner infile = new Scanner (new File (filename));
55:             copylines (infile);
56:             infile.close ();
57:         }catch (IOException error) {
58:             exit_status = EXIT_FAILURE;
59:             err.printf ("%s: %s\n", JARNAME, error.getMessage ());
60:         }
61:     }
62: }
63:
64: // Main function scans arguments and opens/closes files.
65: public static void main (String[] args) {
66:     if (args.length == 0) {
67:         // No files specified on the command line.
68:         catfile ("-");
69:     }else {
70:         // Iterate over each filename given on the command line.
71:         for (int argi = 0; argi < args.length; ++argi) {
72:             catfile (args[argi]);
73:         }
74:     }
75:     exit (exit_status);
76: }
77:
78: }
79:
80: //TEST// mkpspdf jcat.ps jcat.java
81:
```