

```
1: import java.io.File;
2: import java.util.ArrayDeque;
3:
4: /**
5:  * Calculates the total size of a file system object and displays result.
6:  * @author 1828799
7:  *
8:  */
9: public class TotalSize
10: {
11:     public static void main(String[] args)
12:     {
13:         // Stack to keep track of the files
14:         ArrayDeque<File> fileStack = new ArrayDeque<File>();
15:
16:         int totalSize = 0; // The total size
17:
18:         boolean hasUnreadableFile = false; // Boolean for end display message
19:
20:         File currentFile = null; // The current file
21:
22:         File[] directoryFiles = null; // Array to hold array of
23:
24:         // ERROR CHECK: No argument passed in
25:         if (args.length == 0)
26:         {
27:             System.out.println("Please enter a file path.");
28:             System.exit(0);
29:         }
30:
31:         // Initialize the current file to be the root passed in
32:         currentFile = new File(args[0]);
33:
34:         // Push the first file onto the stack
35:         fileStack.push(currentFile);
36:
37:         while (!fileStack.isEmpty()) // While stack is not empty
38:         {
39:             // Pop the stack for the most recent file
40:             currentFile = fileStack.pop();
41:
42:             if (currentFile.canRead()) // If the file/directory is readable
43:             {
44:
45:                 // Add size of directory/file
46:                 totalSize = (int) (totalSize + currentFile.length());
47:
48:                 if (currentFile.isDirectory())
49:                 {
50:                     // Access the files in the directory
51:                     directoryFiles = currentFile.listFiles();
52:
53:                     // If it is an accessible directory
54:                     if (directoryFiles != null)
55:                     {
56:                         for (File aFile: directoryFiles)
57:                         {
58:                             fileStack.push(aFile); // Push files onto the stack
59:                         }
60:                     }
61:                 }
62:             }
63:         }
64:     }
65: }
```

```
61:         else // The directory is not accessible
62:         {
63:             // Set boolean true to later display message
64:             hasUnreadableFile = true;
65:         }
```

```
66:         }
67:     }
68:     else // Else there is an unreadable file
69:     {
70:         hasUnreadableFile = true;
71:     }
72: }
73:
74: // Display size in bytes to screen
75: System.out.println("Total size: " + totalSize + " bytes");
76:
77: if (hasUnreadableFile) // If there are unreadable files display message
78: {
79:     System.out.println("(Note: some files/directories were unreadable)");
80: }
81: }
82: }
```