

NOTE: SCENARIO IS WHAT THE PUZZLE TAKER SEES.

SCENARIO:

See the [J01-File.createNewFile](#) puzzle.

Code:

```
01  // OMITTED: Import whatever is needed.
02  public final class FileUtils {
03      /**
04       * This method copies the contents of the specified source file
05       * to the specified destination file. The source file and the
06       * directory holding the destination file have to exist.
07       * If 'overwrite' argument is true, then the method will
08       * overwrite the destination file if it exists.
09       *
10       * @param source
11       *         the path to an existing file
12       * @param destination
13       *         the path of a destination file
14       * @param overwrite
15       *         whether the copy method should overwrite existing
16       *         destination file
17       * @return true, if the operation succeeded, otherwise false
18       * @throws NullPointerException if source or destination is null
19       *         IOException if an I/O error occurs
20       */
21      public static boolean copy (File source, File destination,
22                                  boolean overwrite) throws IOException {
23          InputStream input = null;
24          OutputStream output = null;
25          try {
26              if (!overwrite && destination.exists())
27                  return false;
28
29              if (destination.createNewFile() || overwrite) {
30                  input = new FileInputStream(source);
31                  output = new FileOutputStream(destination);
32
33                  // OMITTED: By using 'input' and 'output', read all bytes
34                  // from 'source' and write them into 'destination'. At the
35                  // end, 'source' and 'destination' contain the same data.
36
37                  return true;
38              }
39              return false;
40          }
```

```
41      // OMITTED: In finally block, close all open streams;  
42    }  
43  }
```

Questions:

See the [J01-File.createNewFile](#) puzzle.

NOTE: ANSWER IS TO BE SHOWN TO THE PUZZLE TAKER AT THE END OF THE SESSION.

ANSWER:

a

See the [J01-File.createNewFile](#) puzzle.

NOTE: THE REST OF THIS DOCUMENT CONTAINS EXTRA INFORMATION FOR THE PROJECT RESEARCHERS. IT IS NOT TO BE SHOWN TO THE PUZZLE TAKERS.

TAGS:

java, file-class, io-operation, file-operation, ignoring-return-value, old-api

CATEGORIES:

Blindspot - NO

Type - File

Number of distinct functions - 4

Number of total functions - 4

Blindspot function - NA

Function call omitted - NA

Blindspot type - NA

Number of Parameters in the blindspot function - NA

Cyclomatic complexity - 3

No Blindspot, File, 2 functions, 0 parameters

NAME:

File class, createNewFile method - An abstract representation of file or directory pathnames, creates a new file.

MORE INFORMATION:

To see an incorrect way to use the API, look at the [J01-File.createNewFile](#) puzzle.

REFERENCES:

1. [File#createNewFile](#)