

AP COMPUTER SCIENCE A – Scanner CONNECTED TO A FILE

The following method opens a Scanner object connected to an input file `fileName`.

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

.
.
.
/*
 * Open a Scanner connected to fileName.
 */
private static Scanner open ( String fileName ) {
    /**/
    File f;
    Scanner rv;
    /**/
    f=new File(fileName);
    rv=null;
    try {
        rv=new Scanner(f);
    }
    catch ( FileNotFoundException fnfe ) {
        System.out.println();
        System.out.println( "Could not open \"" + fileName + "\" for input." );
        System.exit(0);
    }
    /**/
    return rv;
}
```

AP COMPUTER SCIENCE A – Scanner CONNECTED TO A FILE

The program `ScannerTest.java` reads in the file `scanner_test.txt` listed here.

```
Billy    Bob    Thornton
58       23     7
13.87    17.82  21.95
```

Note that in the first line of the file: Billy is preceded by 2 spaces; Bob is preceded by 5 spaces; and Thornton, by 3 spaces.

- 1) Run `ScannerTest` via `java ScannerTest 1`. This will tokenize the first line of the input file using the `tokenizer1` method, which uses the `s.nextLine()` method of class `Scanner`. The second line of the input file is tokenized by the `tokenizerInt` method, which uses the `s.nextInt()` method of class `Scanner`. Finally, the third line of the input file is tokenized by the `tokenizerDouble` method, which uses the `s.nextDouble()` method of class `Scanner`. Note that unlike the `BufferedReader` class, the `Scanner` methods do not throw any exceptions that need to be caught, making the `Scanner` class more convenient to use. In any case, the output to the console is

```
tokenizer1:
tokens = 11
""
""
"Billy"
""
""
""
""
"Bob"
""
""
"Thornton"

tokenizerInt:
tokens = 3
"58"
"23"
"7"

tokenizerDouble:
tokens = 3
"13.87"
"17.82"
"21.95"
```

Note that the `tokenizer1` method, which uses `split` of class `String`, generates empty tokens, so it is not desirable to use. On the other hand, methods `tokenizerInt` and `tokenizerDouble`, which use, respectively, methods `nextInt` and `nextDouble` of class `Scanner`, work as desired.

AP COMPUTER SCIENCE A – Scanner CONNECTED TO A FILE

- 2)** Run `ScannerTest` via `java ScannerTest 2`. This will tokenize the first line of the input file using the `tokenizer2` method, which uses the `s.next()` method of class `Scanner`. The output to the console is

```
tokenizer2:
tokens = 3
"Billy"
"Bob"
"Thornton"

tokenizerInt:
tokens = 3
"58"
"23"
"7"

tokenizerDouble:
tokens = 3
"13.87"
"17.82"
"21.95"
```

Here, the first line of the input file is tokenized as desired.

- 3)** Starting with program `MatrixMultiply1`, write a program `MatrixMultiply3`, which can read in the files `ab3.txt` or `ab4.txt`, by:
- a)** Removing the static `BufferedReader` methods `open`, `readLineTokens`, `readLine` and `close`.
 - b)** Putting in method `open` from `ScannerTest`.
 - c)** Changing the `import` statements appropriately.
 - d)** Modifying method `readMatrix` to use methods `nextInt()` and `nextDouble()` of class `Scanner`.