Computer Science 212
Object-Oriented Programming in Java

Lab 8

Aim: StringTokenizers and two-dimension arrays.

1. Open Eclipse, set up your private workspace on the H: drive, and create a Java Project for Lab8.

2. A Java program (Tokens.java) has been provided on the public Z: drive under the folder *Lab8*. Import this program as well as TextFileInput.java into your Lab8 project in Eclipse:
   a. Right click on the **src** folder under *Lab8* in your list of Eclipse projects.
   b. Choose *Import*, expand *General* by clicking on the + sign, choose *File System*
   c. Click on *Next*, then *Browse* and go to the Z: drive and select the folder Lab8 and click OK. Click the box next to *Tokens.java* (the file you want to import) and click Finish.

   Also, import the file twodimension.txt into the *Lab8* folder (**not** the src folder) from the Z: drive.

3. In Eclipse, double click the file *Tokens.java*. It should open the file in a tab. Look at line 27 (if you don't see line numbers, right-click in the small column between the scroll bar and the Java source code, and select Show Line Numbers.) The statement

   ```
   myTokens = new StringTokenizer(line,",");
   ```

   creates a new *StringTokenizer* object using a String *line* and a *delimiter* (separator) which in the case is the comma (","). The StringTokenizer will contain each of the substrings (tokens) which are separated by the delimiter. So, if *line* is read from the input file as:

   ```
   "cat,rat,dog,hog,fish,rabbit,horse"
   ```

   then *myTokens* references a StringTokenizer object containing seven strings ("cat","rat","dog",…)

   Observe line 33:

   ```
   animals = new String[myTokens.countTokens()];
   ```

   An array is created to store the strings from the *StringTokenizer*. Notice that the method *countTokens* returns the number of tokens in the string (in this case, 7).

   Finally, observe line 40:

   ```
   animals[i]=myTokens.nextToken();
   ```

   The method *nextToken* returns the next token stored in the StringTokenizer. The first call, in this case, will return "cat", the next "rat" and so on. Notice that the *while* loop terminates when the method *hasMoreTokens* returns false (the last token has been read).

With the cursor somewhere inside the tab with the source code, right click and choose *Run As Java Application*. The program should run, and the output should appear at the bottom in the *Console* tab.

Run the program.

4. Now we will use the StringTokenizer in conjunction with the two-dimension array program from Lab 7. The input file in lab 7 had each number for the array on a separate line. A new data file (*twodimension8.txt*) is in the Lab8 folder on the Z: drive. Import this file in the the *Lab8* project in Eclipse and open it.

   The format of the file is:

   ```
   <number of rows>,<number of columns>
   <number>,<number>,…,<number>
   . . .
   <number>,<number>,…,<number>
   ```

So the input file:

```
3,4
12,45,3,18
7,65,34,8
19,56,9,27
```

Creates the array:

	0	*1*	*2*	*3*
0	12	45	3	18
1	7	65	34	8
2	19	56	9	27

5. Import the *TwoDimArray.java* program from the Lab7 Z: drive folder into the *src* folder for Lab 8.
   Modify the program so that it
   - reads the first line of the input file, tokenizes it to get the number of rows and columns,
   - creates a two-dimension array of *integers* of the proper dimensions,
   - reads the rest of the file, tokenizing each line and storing the number in the array (remember to use *parseInt*).