Jennifer Swofford  
APC 390  
Assignment4: Custom Exceptions & Tests

## Existing Java Exceptions

I didn’t change the existing java exceptions I had in the program so as not to re-invent the wheel. However, I did update the program to make sure that I displayed messages to the end-user in the JOptionPane messageDialogue (rather than the console). I also updated all those exceptions to print the stack trace in the console for the developer.

## FileExtensionException - saveTable

Saving the table without a file extension doesn’t throw an error or cause a crash, but I’ve decided to make sure the user is required to save the file with a file extension. I’ve considered also ensuring the file is saved as a .txt, but ultimately decided to allow the user to have more flexibility on the file type.

Like the existing java exceptions, I display user messaging in the JOptionPane messsageDialogue and the stacktrace in the console for the developer.

## InvalidFileStructureException - loadTable

As mentioned in the CrashReport file, there are a number of general file structure issues that can cause the program to crash when the user loads a table. I ultimately decided to have a general exception handle these issues. The exception message advises the user to check that they are using the correct file type (stadium vs city) or that they chose the right menu option for their file. Additionally, there is instruction for the user that they should have their row elements comma separated with spaces.   
As with all the other exceptions, I’ve also decided to print the stacktrace in the console for the developer.

## Tests:

1. Save file as cities.txt
   1. Expected & Actual Outcome: File saved successfully
2. Save file as cities.2.txt (Option 2)
   1. Expected & Actual Outcome: FileExtensionException thrown
3. Save file as cities.2.txt (Option 7)
   1. Expected & Actual Outcome: FileExtensionException thrown
4. Load a city file (nospace.txt) where the row elements don’t have spaces after the commas
   1. Expected & Actual Outcome: InvalidFileStructureException thrown
5. Load a stadium file (nospaceStadium.txt) where the row elements don’t have spaces after the commas
   1. Expected & Actual Outcome: InvalidFileStructureException thrown
6. Load a stadium file that has blank lines after the header
   1. Expected & Actual Outcome: InvalidFileStructureException thrown
7. Load a city file (cities.txt) when the program expects a stadium file (Option 6)
   1. Expected & Actual Outcome: InvalidFileStructureException thrown
8. Load a stadium file (stadia.txt) when the program expects a city file (Option 1)
   1. Expected & Actual Outcome: InvalidFileStructureException thrown
9. Load an empty file (empty.txt) for city & stadium options (Option 1 & 6)
   1. Expected & Actual Outcome: NoSuchElementException thrown
10. Load a file with > 100 rows (100City.txt. & 100Stadia.txt) for both city & stadium (Option 1 & 6)
    1. Expected & Actual Outcome: ArrayIndexOutOfBoundsException