Class 4/5 – Exceptions, I/O

Exercise 1:

From the RPG game we created in the previous class:
1. Create first a simple Jar file (making sure to add sources as well),
2. Then create an executable Jar file with the same name.
Double click each one to see what happens, why do they not run? Try through a console next. Also check their contents. What are the differences between the two?

Exercise 2:

Write a program for dividing two integers, that you pass as an argument at run time (using Eclipse configurations, or by running through a console). Add exception handling for improper input, and give appropriate information to the user (can also print the call trace).

Exercise 3:

We will now upgrade yet again our RPG. If you want to keep previous versions of your code, create a new package called class5.ioRPG and copy your classes there.

Part one
1. Instead of hardcoding the characters of our game, we will ask the user to input them at the command line. Create a new class responsible for input of characters.
2. In the class create a function that asks the user to input the parameters of a character in turn (e.g., first name, then profession, then health, then mana). Explain in the text you use to prompt your user what is the format you are expecting. For now just repeat back to the user what their input is.

Part two
1. Extend your input class, by checking how appropriate the input that the user is (i.e., if it follows the expected format). Use the appropriate read functions and string manipulation functions to help you check, or throw appropriate exceptions.
2. If you are sure your user input is correct, create the character and add them to your RPG!

Part three
Extend your input, such that the user can create new characters continuously until they input a special sequence (e.g., when they write “done“). Run the RPG with all the new characters.

Exercise 4:

We will upgrate a bit more our RPG, by reading characters from a file (instead of hardcoding them, or of asking the user for direct input). Create a new file with your characters (one character per line), read them, create their respective object in your program and then ran your RPG.

Note, for each line representing a character, you can separate attributes by “,“. This creates a csv (comma separated values) text file that you can save as “characters.csv“

While your program is running, keep a record of the elimination of players, and of the winner. Do this by writing this information into a new file on the disk.

The first exercises are adapted from Evan Jones, Adam Marcus, and Eugene Wu. 6.092 Introduction to Programming in Java, January IAP 2010. (Massachusetts Institute of Technology: MIT OpenCourseWare), http://ocw.mit.edu (Accessed).