Week 15 Suggested Problem: Final Project

Published | Assign To | Edit

This is the suggested practice problem for this week. It has been built to incorporate the objectives that have been covered in this week's lecture and reading. You're free to modify the requirements or build something else entirely, and I'll judge the project based on what you've been able to build.

Overview

If you're missing points on earlier Learning Portfolio Check-In Assignments, it may be best to concentrate on creating work to fulfill those objectives. If you're caught up with those points, or need an original project to raise some 4/5's to 5/5's, then this may help.

Suggested Problem: Character Manager

Build a Character Manager for a simplified Role Playing Game. In the character manager you should be able to create characters, display character stats, have specific characters take actions, save characters to file and load characters from file.

An example run of the program may look like this:

```
Welcome to the Manager!  You have no character selected.  Press 1 to create a new character, 2 to se
lect an existing character,
3 to load a character from file, 4 to save a character to file, 5 to perform an action, or 6 to exi
t.
>> 2
There are currently 2 characters available:
Chad the Barbarian - 1
Chelsey the Wizard - 2
enter the associated number to choose a character
>> 2
Welcome to the Manager!  You have Chelsey the Wizard selected.  Press 1 to create a new character, 2
to select an existing character,
3 to load a character from file, 4 to save a character to file, 5 to perform an action, or 6 to exi
t.
>> 5
What would you like to do?
1 - roll for attack
2 - roll for damage
3 - roll for physical save
4 - roll for magical save
5 - go back to main menu
>> 1
Chelsey rolls a 12 - 3 for a total of 9.
What would you like to do?
1 - roll for attack
2 - roll for damage
3 - roll for physical save
4 - roll for magical save
5 - go back to main menu
```

```
>> 5
```

And so on...

Requirements

- Users must be able to:
  - Create a new Character from a selection of 4 fantasy classes
    - On creating a character, the user is presented with a set of random numbers and is asked to place each number at a different stat in the character
  - Save existing characters to a text file
  - Load characters from a text file into the program
  - Select from multiple loaded characters in the main menu
  - Perform character actions with individual characters
- Loaded characters must be stored in some kind of list
- When a character is created, it's added to the "loadedCharacters" list
- At least 4 Fantasy Classes should be available
  - Examples: Wizard, Archer, Acrobat, Thief, Barbarian, etc. (the classes you use are your choice)
  - Each class should have at least 5 stats that can be set
    - accuracy
    - attack
    - physical defense
    - magical defense
  - All classes should share a common interface:
    - Should have a method to roll for damage that combines a random number generator with the attack stat
    - Should have a method to roll for attack that combines a random number generator with the accuracy stat
    - Should have a method to roll for physical defense that combines a random number generator with the physical defense stat
    - Should have a method to roll for magical defense that combines a random number generator with the magical defense stat
    - You're free to implement this shared interface either through an Interface or via Class Inheritance
- When a character is saved, the user can enter the name of the file to save to
- When a character is loaded, the user can enter the name of the file to load
- The app does not crash at any point and handles bad input gracefully

Hints

- Aside from the classes necessary for your Fantasy Classes (Barbarian, Wizard, etc.), you can also build other classes and method to modularize your project
  - You may have a CharacterManager class that contains your list of characters and provides methods to select different ones
  - You might also have a GameManager class that has methods for the main options you have, simplifying the complexity of your code in your main while loop
  - These are random suggestions that you're welcome to use, completely ignore or use your own ideas of useful classes
- On saving your characters to file...
  - One way for you to save your characters is to not really save your characters to file, but to save their details to file and then recreate them based on the contents of a file
    - To write a character to file, you would just write their name and stats in a text file, with each piece of information being on a specific line in that file
    - To read a character from file, you would read the text file and save all of the details into temporary variables, then create a new instance of the object using those variables. It's not the original object, but it works essentially the same.
  - If you do want to save the object itself, then the answer is serialization. We didn't cover this in class, but you can write Serializable objects to file and read them back again
    - Tutorials on this:
      - **https://www.tutorialspoint.com/java/java_serialization.htm (https://www.tutorialspoint.com/java/java_serialization.htm)**
      - **https://www.baeldung.com/java-serialization (https://www.baeldung.com/java-serialization)**
    - If this is confusing, then concentrate on the other method listed above
    - Remember that when you read a Serialized object back into your code, it's of type Object, so you'll have to cast it to the appropriate class
      - This might be useful for that:
        - **https://www.geeksforgeeks.org/java-instanceof-and-its-applications/ (https://www.geeksforgeeks.org/java-instanceof-and-its-applications/)**

Related Challenges

- Build a larger working project that properly utilizes looping structures, polymorphism, encapsulation and 3 of the following: arrays, arraylists, file I/O, exception handling, recursion, graphical user interfaces

Points	None
Submitting	Nothing

Due	For	Available from	Until
-	Everyone	-	-

+ Rubric