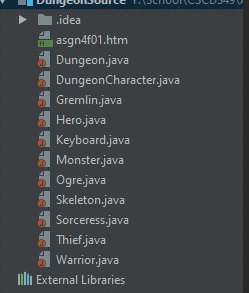
DungeonSource Refactoring

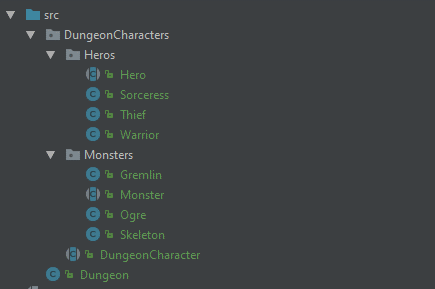
1. Create Packages for the Hero, Monster characters and their parent classes

Packages make the code much easier to maintain and navigate. Now I know exactly what are monsters and what are heroes and can add new classes without making it harder to navigate

* 1. Before

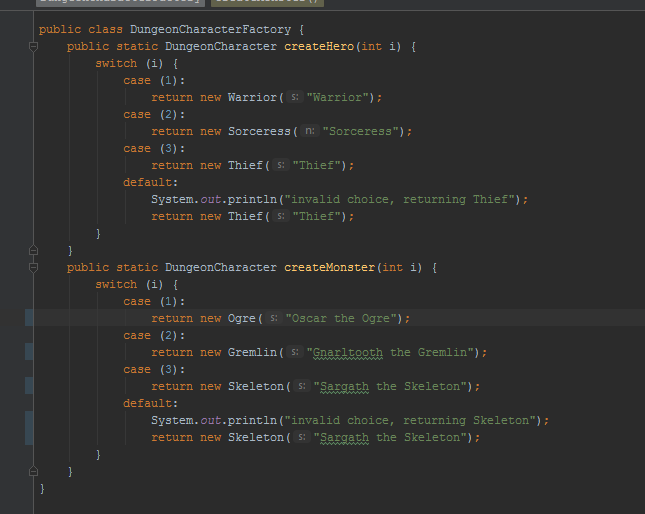


* 1. After



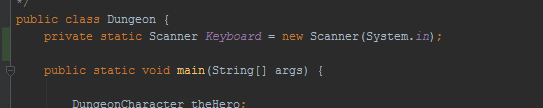
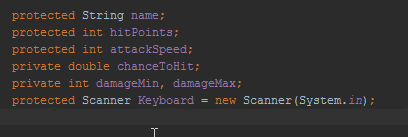
1. Created DungeonCharacterFactory

Separates the user/tester from knowing what kind of dungeon character they have. All they need to know is they are or are fighting against a dungeon character created based on input of number. The factory handles the rest. This will allow for runtime changes to characters (or character abilities based on my further refactoring of the code)

* 1. 

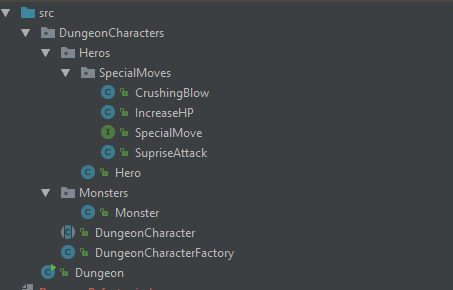
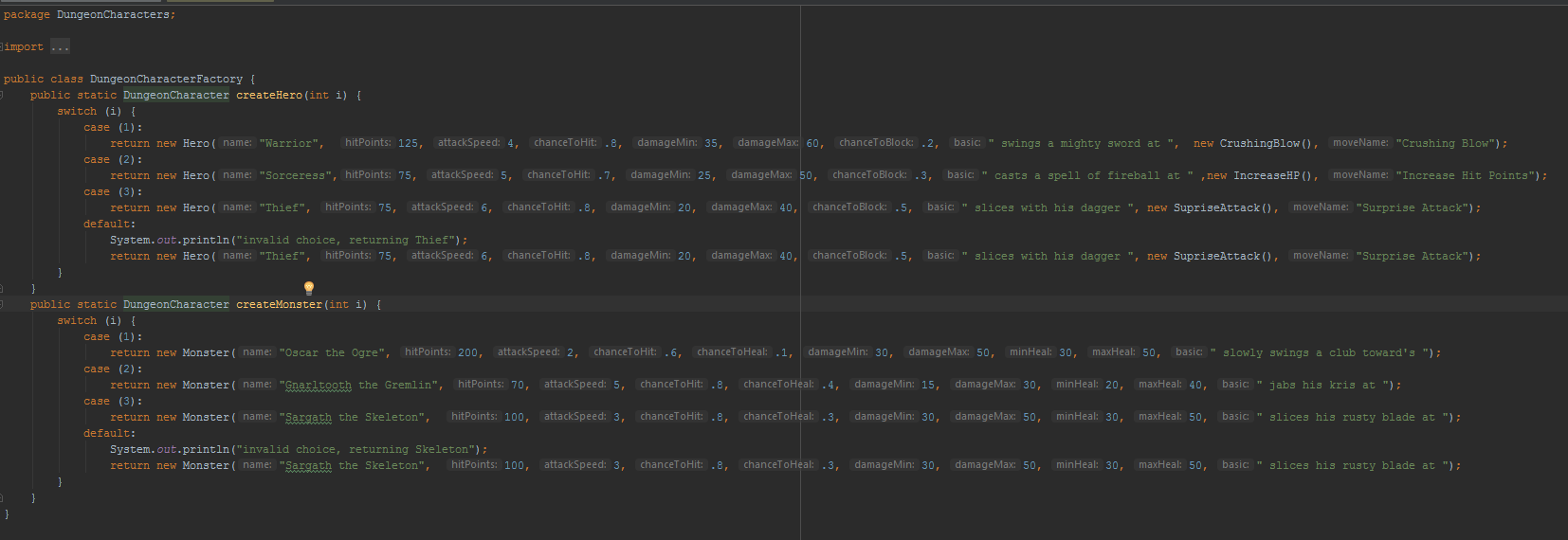
1. Use Scanner instead of Keyboard class now

Scanner is a trusted tool that, since its built into the api, should function better than the keyboard class provided.

* 1. 
  2. 

1. Refactored concrete child classes of monster and hero to no longer need them, changed factory to create generic DungeonCharacters of both hero and monster that could be changed at run time and made to do whatever the user wanted and created special moves to distinguish different characters (those too could be applied to any character created)

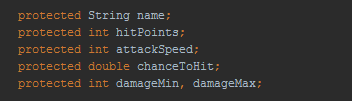
This was the best choice of refactor as now the game is completely customizable and can add new characters from the factory. No new classes need be created for new monsters or heroes. Only new special moves will need to be created (also covers part of the final project requirements)

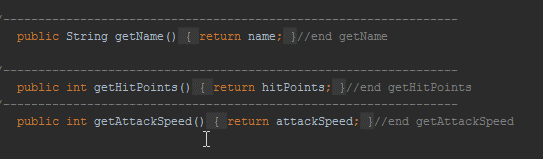
* 1. 
  2. 

1. Changed the visibility of methods and variables as necessary throughout the refactor process for my code to work

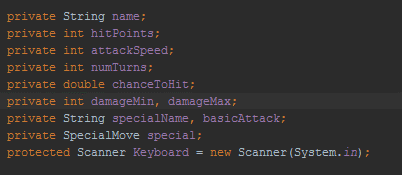
I did this because as the structure of the program changed so did the visibility needs, moving from all protected to most private.

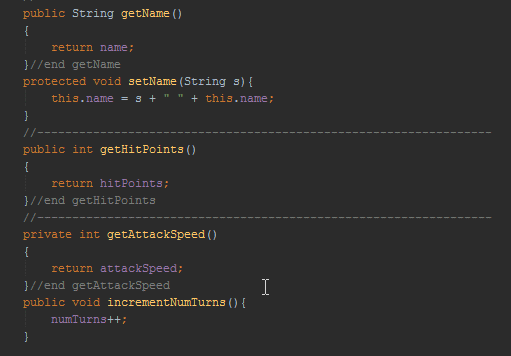
* 1. Before





* 1. After



 Added setName() and incrementNumTurns() to make name private