Melody Kinyon

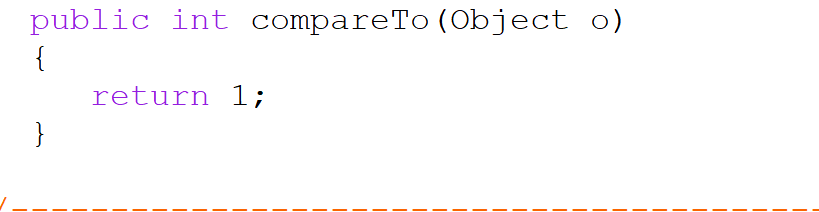
Corbin Smith

Jordan Fisher

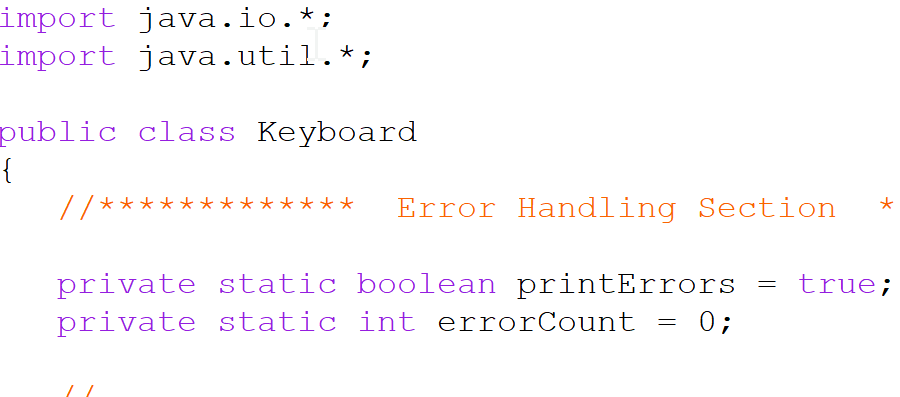
CSCD349 Team Based Assignment: Refactor

Refactor Ideas-

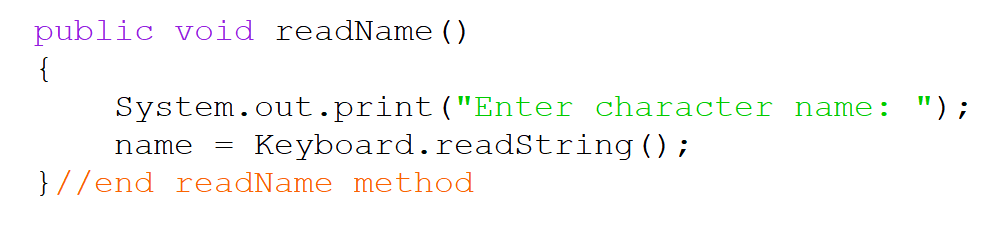
1. Implement a Hero and Monster Factory- ***This was successfully implemented in our current code!***
2. Remove the unused Comparable Interface in the DungeonCharacter abstract class as the compareTo was not meaningfully implemented(Refused Bequest).



1. Remove the keyboard class and use the scanner Object instead



1. Remove the readName() method from the Hero Class and move it to the main, as it made more sense to set it there and then send the name to the HeroFactory to make our heroes.



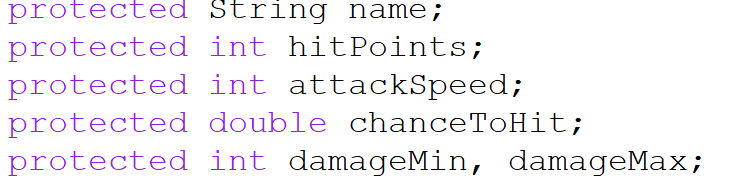
1. Rename methods and fields to be more intent revealing as some of the names were confusing

***We renamed the DungeonCharacters to be GameCharacters and refined the names of the data fields***

1. Create an Attack interface as all DungeonCharacters(renamed GameCharacters) have the ability to attack, so we have a regular attack and a special attack there. It just made good sense to push it up to a superclass.(Single Responsibility);

***We did accomplish this successfully!***

1. Remove the prompt after each attack and only prompt the user to play again only after the battle is concluded. This was more of an annoyance as the game play went on and it made the most sense to change it in this way. ***This feature is fully working now***
2. Mainstream the display of the statistics for the players as it felt clunky- ***We did implement this in the main, and only report a missed hit from the individuals classes.***
3. Change accessibility modifiers throughout so that things are private.



1. We moved the getters and setters from the individual Hero and Monster classes and put them into the GameCharacter class instead- ***We got this change accomplished.***