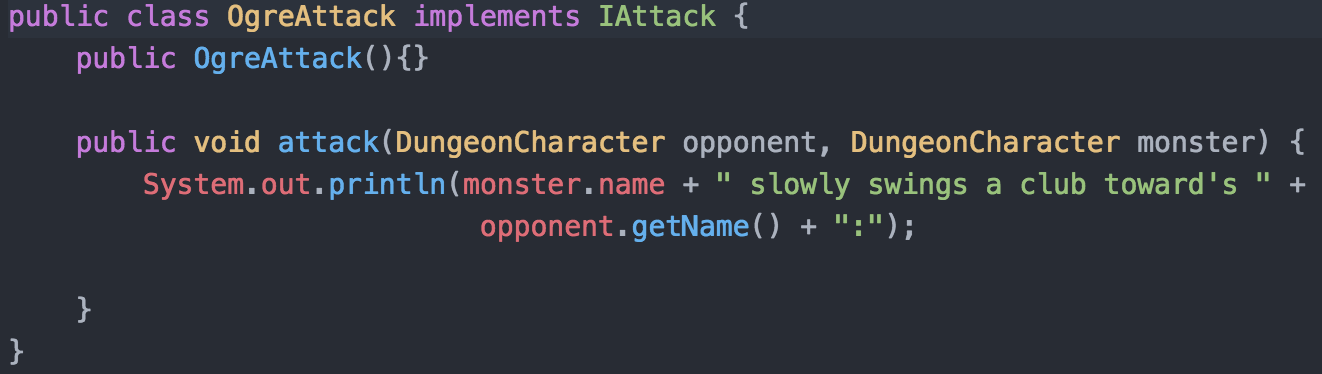
Refactors for Heroes and Monsters assignment:

1. Got rid of concrete classes, Hero and Monster are now the concrete classes. This makes things easier to extend in the future when adding more Heroes/Monsters, as well as adding ability to let other heroes/monsters perform different abilities if we want to in the future. Below is a snippet of an Ability Class that each Hero has. Monsters have specific Attack classes that only call their respective attack method.



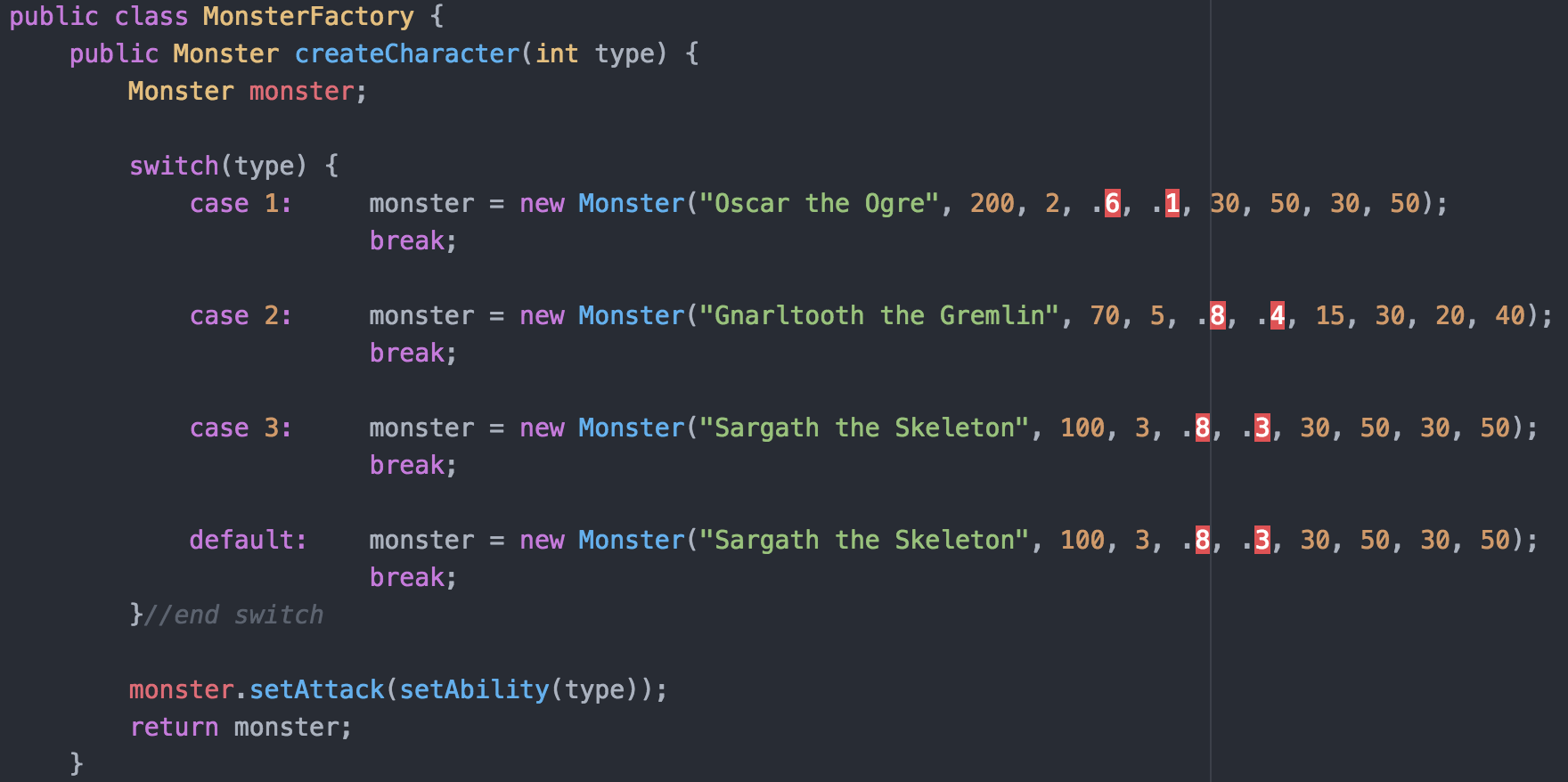
Example of the Monster Attack Ability



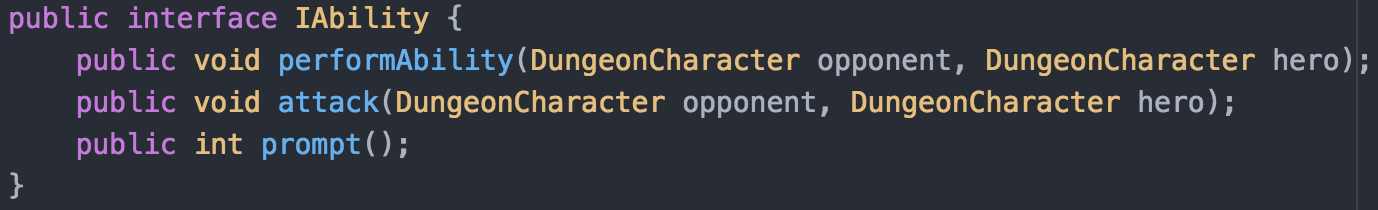
1. Added HeroFactory.java to create Heroes in an easier more organized way. Every time a monster is created they will have the ability style that suits them assigned to them.



1. Added MonsterFactory.java to create Monsters in an easier more organized way. Every time a monster is created they will have the attack style that suits them assigned to them.



1. Applied strategy pattern so each hero can have their own different attack/special abilities. This was achieved by adding an interface(IAbility.java) that each Hero will have as an attribute, the interface allows us to switch actions for the heroes. (See first picture for full class example.)



1. Applied strategy pattern so each Monster can have their own different attacks. This was achieved by adding an interface (IAttack.java) that each Monster will have as an attribute, this allows us to switch attacks for the Monsters. (See second picture for full class example.)

