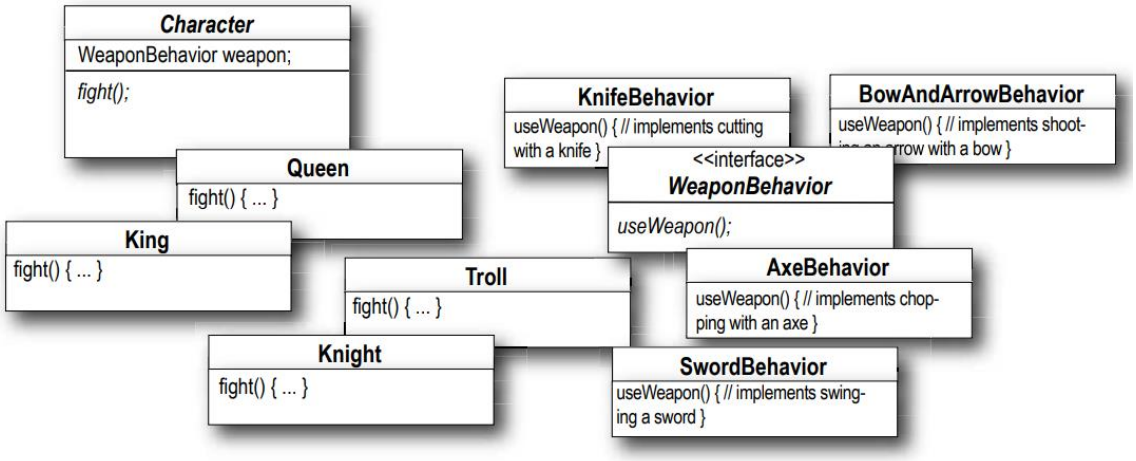
Design Puzzle

Below you'll find a mess of classes and interfaces for an action-adventure game. You'll find classes for game characters along with classes for weapon behaviors the characters can use in the game. Each character can make use of one weapon at a time, but can change weapons at any time during the game. Your job is to sort it all out.

Your task:

1. Arrange the classes.
2. Identify one abstract class, one interface and eight classes.
3. Draw arrows between classes for the following:
    a. Arrow for inheritance ("extends").
    b. Arrow for interface ("implements").
    c. Arrow for "HAS-A".
4. Put the method setWeapon() into the right class.




```
setWeapon(WeaponBehavior w) {
    this.weapon = w;
}
```

The purpose of many of the design patterns is to make it easy to change some property of the system. What design pattern would you use to make it easy to change:

A. The algorithm that an object uses.

B. The class of the object that a method returns.

C. The kind and number of objects that react to changes in the object you are designing.

D. Adding operations to classes without changing the class.

E. How methods in a class behave.