Java SOLID Adventure

November 29, 2018

Introduction

Welcome to the Java SOLID Adventure, a fantasy world where you are the Hero. Everything in this world works on 5 SOLID principles, those who violate those rules will be severely punished by the Supreme Teacher.

A small reminder of the **Rules**:

- **Single Responsibility:**
  One class should only have one responsibility and it should be able to execute it well.
- **Open for Extension/Closed for Modification:**
  Modules should allow for extensions to add new features without affecting existing code to make sure the changes are confined to a smaller part of code and new bugs are not introduced to existing code.
- **Liskov Substitution:**
  A child class must be able to completely substitute and act-in for it's base(parent) class.
- **Interface Segregation:**
  A client should never be forced to implement a function that it does not require. Instead of having bloated interfaces, segregate them based on roles.
- **Dependency Inversion:**
  Entities must depend on abstractions instead of concretions. Instead of using direct references from a high-level module to a low-level module, use abstractions.

Main Quest

Your main task in this world is to describe and create two armies, one of allies and another for enemies, using the rules of this world of course. Each type of soldiers have their own moves and actions, when creating, take into account their type and actions in Character Description section. There are several creatures that are mandatory for you to implement, they are listed in the section below, but feel free to add any additional action you think is suitable for that kind of character. For method implementation, just add some console output in the methods in order to get some text to see.

Characters Description

In this section you can find general description of the characters and some examples of classes and methods. Look at the possibilities of each character and build an army suitable for you.

Nb: No need for complex logic, will be enough to print method name in console.

List of **mandatory** allied creatures and what they can do:

Allies

- AllySoldier
  - Attack
  - Walk
  - Protect
- AllyDragon
  - Attack
  - Fly
- AllyMage
  - Attack
  - Cast Spell
  - Fly
- AllyCommander
  - Attack
  - Promote Subordinate
  - Send Orders

Enemies

- EnemySoldier
  - Attack
  - Walk
- EnemyDragon
  - Attack
  - Fly
- EnemyWitch
  - Attack
  - Curse

  - Fly

- EnemyCommander
  - Attack
  - Promote Subordinate
  - Send Orders

The creatures described above are the main actors of your armies, add new ones if needed. For ex: add a **Race** property for every creature. (Hint: use Interfaces)

An example of an AllyCommander listed below sows how we can create such an entity extending it's race (person) and implementing the Ally interface, because it's our Commander. (**Nb:** those examples also do violate the SOLID principles)

```java
public class AllyCommander extends Person implements Ally {

  void sendOrders(AllySoldier allySoldier) {
    System.out.println( ”Order to all army, Attack!”)}

  @Override
  public void help() {}

  @Override
  public void protect() {}

  @Override
  public void attack() {}

  @Override
  public void walk() {}

  @Override
  public void fly() {}
}
```

Listing 1: AllyCommander Example

Another example is the Interface for Allies which describes the actions:

```java

public interface Ally {

  void help();

  void protect();

  void attack();

  void walk();

  void fly();

}
```

Listing 2: Ally Interface Example