魔兽世界2文档

xxxxxxxxx xxx

# 说明

在对应的文件夹中有类图classdiagram.drawio，使用<https://app.diagrams.net/> 可查看。

# 需求变化分析

魔兽世界2主要是增加了“武器”这一需求，其次是dragon的士气和lion的忠诚度。

# 需求实现

由于需求是逐步变更的，这里主要说明针对变化的部分的做法，关于魔兽世界1的部分不再赘述。

## 士气、忠诚度

先从简单的下手，士气和忠诚度都是个性化的内容，所以只需要在Dragon类中添加对应morale属性代表士气，在Lion类中添加loyalty属性，并提供公共的getter和setter方法来获取和设置即可。而这两个属性就交给武士建造者类来设置，在LionBuilder类和DragonBuilder类的建造武士的buildWarrior方法中初始化这些值即可。

## 武器

接着，处理武器的问题，首先建造一个武器的基类Weapon，让该基类继承GameObject以获得名字的属性，在Weapon类中使用一个属性weaponNo记录武器的编号(0、1、2)，然后派生出三个子类代表三种武器，Arrow，Sword和Bomb。在三种武器的构造函数中，设置对应的武器的编号，比如Sword就设置成0，Bomb就设置成1，与题目相符。并且在构造函数中还要设置每种武器的名字，比如Arrow就设置为“arrow”，Bomb就设置为“bomb”，方便后续Logger打印使用。

### 武士与武器

由于武士需要持有武器，所以在Warrior类中添加一个List<Weapon>，表示该武士所拥有的武器。同时提供addWeapon方法来将一个武器添加到该武士的武器列表中。

### 武器的原型复制

因为武器大同小异，不同于武士有编号上的差异，所以使用原型模式来将武器赋给武士。首先让Weapon类实现Cloneable接口，然后重写clone方法，以实现武器的复制。接着构造一个WeaponFactory工厂类，用于生产武器，在其中有一个静态的List<Weapon>，装着武器的原型，然后提供addWeapon和getWeapon两个方法用于添加武器和获取武器，其中添加武器就直接添加到List中即可，而获取武器则是根据用户需要获取的武器的编号，使用List中对应的武器原型的clone方法复制一把武器，然后返回复制品。

### 武器的装配

武器由武器建造者装配给武士，在可以拥有武器的武士的Builder类中，使用WeaponFactory的getWeapon方法获取对应编号的武器，然后调用武士的addWeapon方法将武器装配给对应的武士即可。

## 收尾工作

至此，基本需求都完成了，做最后的收尾工作。在GameMain的Main方法中添加上三行代码，往WeaponFactory类中添加上三种武器原型，用于后续的基于原型模式的武器复制。

## 输出武士诞生信息

之后由于输出的需求也变更了，所以在Warrior类中写一个getBornMessage方法，返回与魔兽世界1类似的基本信息，而对于需要特别打印的武士，如dragon需要打印士气，lion需要打印忠诚度，就重写这个方法，并在第一行首先调用基类的getBornMessage以满足里氏替换原则，然后接着在返回的字符串上进行扩展，需要打印士气的就接着连接上打印士气字符串，需要打印忠诚度就连接上打印忠诚度的字符串，有的武士可以拥有武器的就一并在这里连接上打印武士的武器的信息，最后将字符串返回。

Logger类中有对应的方法，传入一个Warrior类的对象，调用Warrior类的getBornMessage方法获得武士的出生信息（多态行为），并打印。

在司令部构建武士完成后，调用Logger类的方法打印对应的信息即可。

# 额外修改

除了满足题目的需求以外，根据题目的武士建造情况，可以考虑将武士建造的过程改造成责任链模式执行。

## 寻找武士建造者、建造武士

在WarriorBuilder类中添加一个引用自己的引用next，并在Main方法中按照创建的顺序将各个WarriorBuilder连接起来，形成环形责任链。在每一个WarriorBuilder中添加一个方法findBuilder()，这个方法负责在链中找到下一个武士生产者。并且在司令部类中添加一个WarriorBuilder类的引用，指向当前的武士生产者。

司令部首先调用findBuilder方法，根据司令部剩余的生命元数量找到下一个武士生产者，接着调用这个找到的武士生产者的buildWarrior方法生产对应的武士，并将司令部类中的武士建造者的引用指向下一个武士建造者。

## NoWarriorBuilder哨兵

除此以外，还需要添加一个哨兵NoWarriorBuilder，也接在责任链上，如果发现当前的司令部的生命元已经无法生产任何一种武士时就由NoWarriorBuilder接管否则直接让它的next接管，其buildWarrior方法返回的是null，当司令部检测到建造的武士是null时就打印停止制造武士的信息，并从消息系统中反注册自己。