

# 红岩Android第三次作业-面向对象进阶

## Level0 复习知识

复习上节课所讲的内容，结合课件以及所给出的文章链接掌握对**final**、**static**关键字、接口回调、抽象类以及内部类等知识的使用。

## Level1 英雄反击

利用接口回调的知识完善第二次作业中的文字小游戏内容，实现**当每次英雄被攻击后，如果英雄仍未死亡，就执行英雄反击**，如果英雄死亡则直接结束战斗的功能。

## Level2 国王排名

不知道大家有没有看最近新出的一部动漫《国王排名》，动漫里的波吉王子虽然身形微弱、力量很小，但是它的身手却很灵敏。

请为波吉王子实现一套**闪避系统**，该系统分为四个等级：

- 第一级，如果当前的攻击值大于自身防御值的三倍时，王子选择直接逃跑；
- 第二级，如果当前的攻击值大于自身防御值的两倍时，王子选择挡下一击后再逃跑；
- 第三级，如果当前的攻击值大于自身防御值时，王子选择躲避此次攻击并对对方执行反击；
- 第四级，如果当前的攻击值小于等于自身防御值时，王子选择直接反击。

此题可能会用到的英文单词：dodge——闪避；enemy——敌人；attack——攻击；ATK——attack的缩写。

**注意：**这里要求四个等级的闪避系统被依次链式调用，即敌人的攻击方法只能被调用一次，在第一级闪避系统判断完后自动调用第二级闪避系统判断，后面的以此类推，因此不需要在第二级中判断攻击值是否小于防御值的三倍。

大体上需要设计三个类：

1. **抽象类：**Dodge.class，表示对闪避系统的抽象，其中**包含一个抽象方法dodgeEnemy()和一个Dodge本身的变量next**，如果当前防御系统不满足条件，则调用next变量执行其中的dodgeEnemy()方法。下面有一个简单的示例：

```
public abstract Dodge {  
    Dodge next; // 下一级闪避系统  
    abstract void dodgeEnemy(Enemy enemy); // 躲避敌人攻击的抽象方法，在其子类里具体实现  
}
```

最后你需要在主函数中，将四个等级的闪避系统对象依次赋值给上一级闪避系统的next变量。像下面这样：

```
dodgeLv1.setNext(dodgeLv2);  
dodgeLv2.setNext(dodgeLv3);
```

2. **具体实现类：**DodgeLv1.class, DodgeLv2.class, DodgeLv3.class, DodgeLv4.class，分别表示四个等级的闪避系统。

### 3. 敌人类：Enemy.class。

- 其中包含一个常量ATK，表示敌人的伤害值，并选择在**构造方法中对其赋值**；
- 包含两个方法：
  1. 攻击王子的方法，并接收王子的闪避系统参数：**void attack(Dodge dodge)**，在其中调用闪避系统的闪避方法。
  2. 被王子反击的方法：**void attacked()**，在其中直接打印“反击成功”的字符串即可（在闪避系统里回调该方法）。

在Dodge类中声明一个**静态常量**表示王子自身的防御值DODGE。选择逃跑时直接打印“王子逃跑”的字符串即可。

## Level3 国王阅兵

在动漫中的世界里，有多个王国，每个王国内都有一位国王，国王手下都有很多士兵。

请你为国王们设计一个阅兵系统，让国王们可以**一声令下让自己的所有士兵都能收到通知**，回答“国王好”。

此题可能会用到的英文单词：king——国王；soldier——士兵；review——阅兵；response——回答。

大体上也需要设计三个类：

1. **抽象类**：King.class，国王类，王国内所有的军队力量都归为国王一人所有，所以**该抽象类需要持有所有将军士兵对象的数组**，并在其中提供删减士兵的公共方法。除此之外，**抽象类中需要有一个阅兵的抽象方法review()**，让国王们自己实现该方法，即对自己的士兵们阅兵。

```
abstract class King {  
    List<Soldier> soldiers = new ArrayList<>();  
    abstract void review();  
}
```

2. **接口**：Soldier，不管是将军还是士兵都需要回应国王的阅兵，因此可以将**回应国王的这一行为**可以提取成Soldier接口中的一个方法void response()。
3. **具体实现类**：KingBoji.class，表示波吉国王类，在重写的阅兵方法中依次调用其父类的士兵数组对象中的response方法。Soldier1.class, Soldier2.class等等，实现Soldier接口并重写该士兵对阅兵的回应方式。

**进阶1**：使用匿名内部类代替Soldier1.class, Soldier.class等类被国王添加

**进阶2**：使用lambda表达式进一步优化匿名内部类的语法

## Level4 国王排名进阶

这道题可以先不用写，下节课上课前结合着参考答案看一下即可。

完成这个作业需要使用的知识点有**静态内部类**，**构造器模式**和**链表**。

改造Dodge抽象类，在其中加入一个构造器，在主函数中通过链式调用为dodge对象添加匿名内部类。

```
Dodge_Lv4 dodge = new Dodge_Lv4.Builder()  
    .add(new Dodge_Lv4() {  
        @Override  
        void dodgeEnemy(Enemy enemy) {  
            if (enemy.getATK() > DODGE * 3)  
                System.out.println("王子逃跑");  
            else
```

```
        getNext().dodgeEnemy(enemy);
    }
})
.add(new Dodge_Lv4() {
    @Override
    void dodgeEnemy(Enemy enemy) {
        if (enemy.getATK() > DODGE * 2)
            System.out.println("王子挡下一击后逃跑");
        else
            getNext().dodgeEnemy(enemy);
    }
}).build();
```