



# 实验5 结构型模式2

## 1 实验目的

学会用UML设计类图，熟练掌握装饰者模式。

## 2 实验环境

开发环境：JDK 8.0（或更高版本，高版本要下载独立JavaFX）

开发工具：Eclipse

设计工具：StarUML（或PlantUML）

## 3 基础知识

### 3.1 装饰者模式

装饰者能动态地给对象添加职责（功能），相比通过类扩展增加功能的方式更灵活，具体见讲义。

## 4 实验内容

### 实验1 动图字幕编辑器

#### 问题描述

学习用装饰者模式设计一个装扮系统，实现对游戏角色动态换装和增加装饰，这里主要考虑三种装扮：皮肤、保护罩和等级，可以动态选择不同的装扮组合，并能对装饰的位置进行调整。



## 提示与解析：

1、将游戏角色的外观生成过程看作多个图形要素的融合过程。例如基础层定义一个图形框架，规定了范围和位置，在基础层之上依次添加皮肤、保护罩、等级等装饰要素。这里涉及到一个需要动态扩展的功能：更新和生成外观。将其定义为接口：

```
public interface ILook {  
    public Entity entityToDraw();  
}
```

2、定义抽象装饰者类，实现 `entityToDraw()` 的公共代码，即图形的融合方式。这里通过 `getImageForMerge()` 获取要参与融合的装饰图片，由具体装饰类实现。

```

public abstract class LookDecorator implements ILook{
    private ILook look;
    private int offx;
    private int offy;
    private int width;
    private int height;

    @Override
    public Entity entityToDraw() {
        Entity e = look.entityToDraw();
        Image img = getImageForMerge();
        Rectangle2D r = e.getFrame();
        Canvas canvas = new Canvas(r.getWidth(), r.getHeight());
        GraphicsContext gc = canvas.getGraphicsContext2D();
        gc.drawImage(e.getLook(), 0, 0);
        gc.drawImage(img, getOffx(), getOffy(), getWidth(), getHeight());
        SnapshotParameters parameters = new SnapshotParameters();
        parameters.setFill(Color.TRANSPARENT);
        e.setLook(canvas.snapshot(parameters, null));
        e.setFrame(new Rectangle2D(0, 0, r.getWidth(), r.getHeight()));
        return e;
    }
    public abstract Image getImageForMerge();
}

```

3、定义具体装饰类，以皮肤类为例。主要实现 `getImageForMerge()` 函数，确定参与融合的图片。

```

public class Skin extends LookDecorator {
    private DynamicImage looks;

    @Override
    public Entity entityToDraw() {
        Entity e = super.entityToDraw();
        return e;
    }

    @Override
    public Image getImageForMerge() {
        looks.nextFrame();
        return looks.getImage();
    }
}

```

4、界面功能见源代码注释。

### **具体要求：**

- 1、调试样例程序，使其正常运行；
- 2、绘制装饰者模式相关的类图，说明实现过程和使用效果。

## **5 实验要求**

### **5.2 实验评价**

- 1、完成实验内容（60%）
- 2、对实验思路进行阐述（20%）
- 3、对实验过程进行总结（20%）

### **5.1 实验报告**

- 1、根据要求完成实验内容、思路阐述和总结。
- 2、截取清晰的核心代码、设计图和效果到报告中；