



# 实验4 结构型模式1

## 1 实验目的

学会用UML设计类图，熟练掌握适配器模式。

## 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 适配器模式

原型模式通过继承或组合的方式实现代码复用，并用于实现新接口，具体知识参考讲义。

## 4 实验内容

### 实验1 动图字幕编辑器

#### 问题描述

学习设计实现一个简单的动图（GIF）字幕编辑器，能对动图的每一帧画面添加字幕，支持添加字幕的动图导出。



## 提示与解析:

- 1、经过分析，动图每一帧包含图片和延时信息，这里可以自定义一个动画帧 `Frame`：

```
public class Frame {  
    private Image image; // 这里为JavaFX中的Image类  
    private int delay; // 毫秒  
}
```

- 2、动图处理的核心功能是解码和编码，面向JavaFX的图形界面程序设计两个接口：

```
public interface IGIFReader {  
    public Frame[] read(String gifPath);  
}  
  
public interface IGIFWriter {  
    public void write(Frame[] frames, String targetPath);  
}
```

- 3、采用适配器模式设计 `GIFReader` 和 `GIFWriter` 两个工具类。
- 4、界面的外层布局采用 `SplitPane`，分为图片信息展示区和字幕编辑区
- 5、图片信息展示区采用 `VBox` 垂直布局，其中文件浏览和播放控制又分别采用 `HBox` 水平布局。

- 6、浏览按钮单击事件弹出文件选择框（FileChooser），选择GIF图片。通过解码器获取目标文件的每一帧图片信息，采用数组等容器存储。
- 7、图片展示区采用画布（Canvas）渲染。
- 8、当滑块状态改变时，获取当前滑块位置对应的帧，显示当前帧对应的图片。
- 9、单击播放按钮后播放完整的动图，主要采用Timeline实现。
- 10、字幕编辑要素包含复杂信息，例如起始帧、结束帧和字幕等，为此设计一个字幕编辑面板TextPane：

```
public class TextPane extends VBox{  
    private Pane father;  
    private TextField content;  
    private TextField start;  
    private TextField end;  
}
```

- 11、单击“添加字幕”按钮添加字幕编辑单元
- 12、单击“导出动态图片”按钮导出带字幕的动图

### 具体要求：

- 1、调试样例程序，使其正常运行；
- 2、分析和理解程序的实现过程；
- 3、哪些类采用适配器模式设计，具体是如何实现的？结合程序功能解析设计意图和作用。（给出类图、实现代码截图以及必要的说明）
- 4、程序功能上还存在哪些问题？是否能谈谈完善的思路。（例如字幕位置拖拽、添加特效、帧位置的输入保护等）

## 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；