



《软件体系结构与设计》

第三章

作业 5

班 级： 111171
学 号： 20171000970
学 生 姓 名： 董安宁
指 导 教 师： 尚建嘎

中国地质大学地理与信息工程学院软件工程系

2019 年 10 月

基于 Java 的多态性实现一个模拟的播放器插件程序。

基于插件的播放器的原理是在宿主程序声明一个接口类, 然后基于接口类生成不同的项目, 打包成为 jar 包, 然后再在播放器运行的时候动态加载这些 jar 包。由于这些 jar 包都是基于接口重写的, 因此他们的主要功能函数都是相同的 (在本例中就是播放音乐啦), 因此在加载他们的时候无需考虑, 就知道该使用什么函数。这样一来就可以实现插件的功能了。

实验步骤:

① 定义接口类:

首先定义 IPlayerPlugin 接口, 由于其功能只是播放音乐, 因此函数设计的就是根据路径来播放音乐, 如下图所示, 只有一个 play 函数等待实现:

```
package com.plugin;

//定义插件类
public interface IPlayerPlugin {
    //播放音乐
    public void play(String path);
}
```

图 1 定义插件接口

② 分别实现接口类:

首先把接口所在的类进行打包, 形成 jar 文件, 然后新建接口实现的类, 导入该 jar 文件, 这样就可以导入该包的接口了。如下图所示, 左边的工程结构中可以看到导入了我自己写的 Music 包, 然后就可以如右图所示, 导入 IPlayerPlugin 插件, 这样就可以在下面重写 play 功能, 如下图所示。



```
package com.wav;

import java.io.FileInputStream;
import com.plugin.IPlayerPlugin;
import sun.audio.AudioPlayer;
import sun.audio.AudioStream;

public class WAVPlayerPlugin2 implements IPlayerPlugin {

    @Override
    public void play(String path) {
        // TODO Auto-generated method stub
        try {
            FileInputStream fileau=new FileInputStream(path);
            AudioStream as=new AudioStream(fileau);
            AudioPlayer.player.start(as);
        }
        catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

图 2 在 WAVPlayerPlugin 中实现 play 功能

- ③ 然后再将该插件打包成 jar 包，这样一来就可以在宿主程序中运行了。宿主程序中应该像反射一样，动态调用 jar 包，使用里面的函数，就可以调用该 jar 包里面的功能了，这样就实现了插件的功能。

```
public IPlayerPlugin getInstance() throws Exception {
    //得到jar包的文件路径
    URL[] urls = new URL[1];
    urls[0] = new URL("file:"+jar);
    //动态加载jar包
    ClassLoader loader = new URLClassLoader(urls);
    // 插件实例化对象，插件都是实现PluginService接口
    Class<?> clazz = loader.loadClass(className);
    //返回插件类型
    IPlayerPlugin instance = (IPlayerPlugin)clazz.newInstance();
    return instance;
}
```

图 3 动态加载插件

- ④ 与上面的类似，借着实现别的插件，如播放 MP3 的插件。由于我没有找到播放 ogg 的代码，因此并没有实现该功能，最终界面如下图所示。

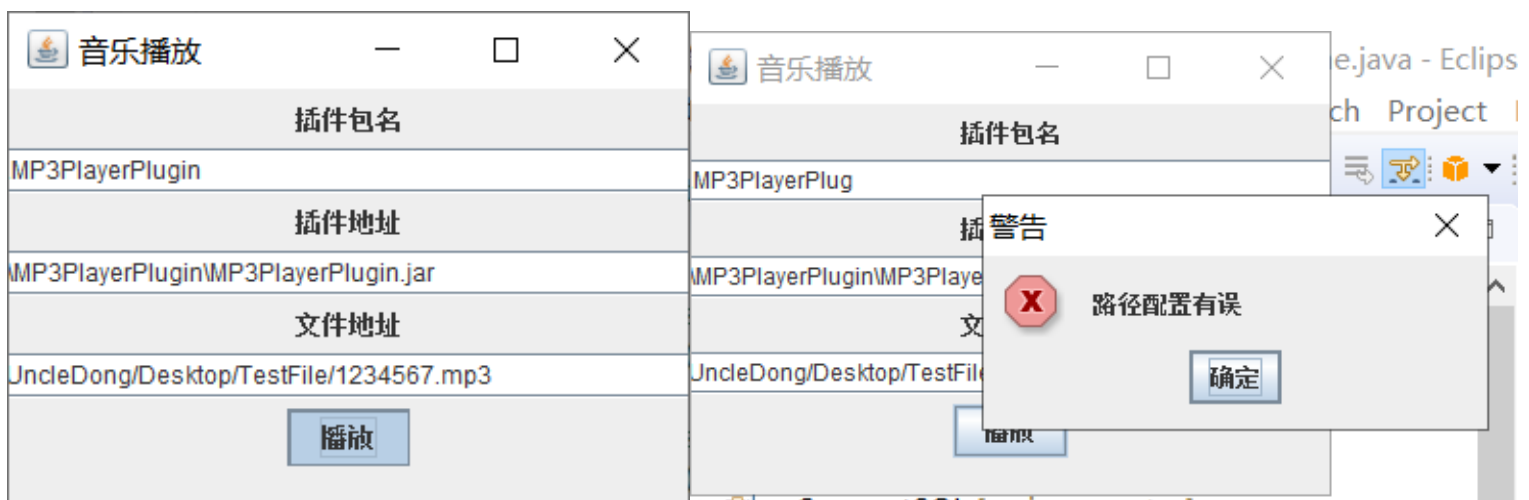


图 4 正确播放音频和路径配置有误

- ⑤ 代码已经上传到了 GitHub 仓库，附带两个插件的 jar 包和测试的音频文件
 GitHub 代码链接：<https://github.com/UncoDong/SAlnCug>

Java 接口开发 <https://blog.csdn.net/u010675669/article/details/86677540>