

西安邮电大学

(计算机学院)

课内实验报告

实验名称：设计模式 1

专业名称：软件工程

班 级：软件 2003

学生姓名：葛玉菲

学 号：04203103

指导教师：高巍然

实验日期：2022-12-5

一. 实验目的及实验环境

1. 实验目的

创建型模式学习及应用；

2. 实验环境

(1) 硬件

- CPU：Intel® Core™ i5-1035G1 CPU @ 1.00GHz × 8
- 内存：15.3 GiB
- 硬盘空间：512.1 GB

(2) 软件

- 操作系统名称及版本：Ubuntu 20.04.5 LTS
- 工具：IDEA

二. 实验内容

1. 要求结合实际案例，每人完成一个创建型模式的类图设计

2. 结合前面类图设计，实现该创建模式的主要代码并通过编译运行，提交代码说明和运行结果截图；

简单工厂模式

实例：简单电视机工厂

某电视机厂专为各知名电视机品牌代工生产各类电视机，当需要海尔牌电视机时只需要在调用该工厂的工厂方法时传入参数“Haier”，需要海信电视机时只需要传入参数“Hisense”，工厂可以根据传入的不同参数返回不同品牌的电视机。

现使用简单工厂模式来模拟该电视机工厂的生产过程。

类图：

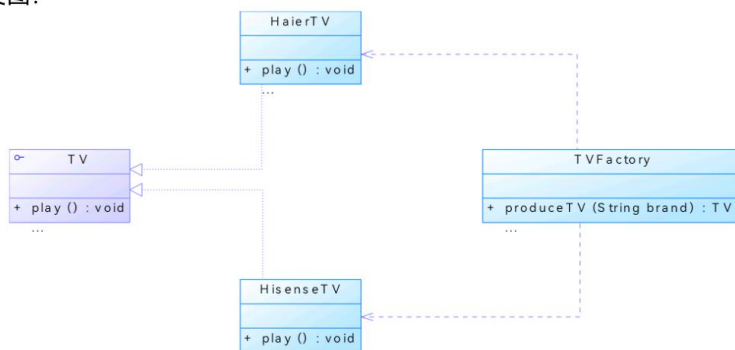
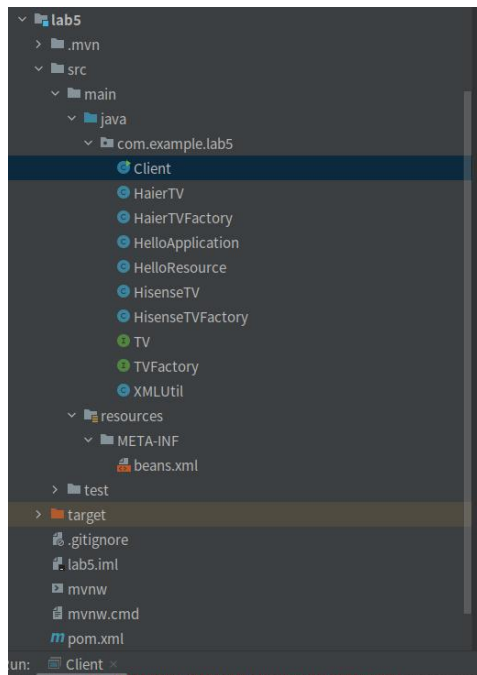


图4-4-1

三. 主要实验步骤截图



四 . 源代码

Client 类 :

```
public class Client
{
    public static void main(String args[])
    {
        try
        {
            TV tv;
            TVFactory factory;
            factory=(TVFactory)XMLUtil.getBean();
            tv=factory.produceTV();
            tv.play();
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```

config.xml 文件

```
<?xml version="1.0"?>
<config>
    <className>HisenseTVFactory</className>
</config>
```

HaierTV 类 :

```
public class HaierTV implements TV
{
    public void play()
    {
        System.out.println("海尔电视机播放中.....");
    }
}
```

HaierTVFactory 类：

```
public class HaierTVFactory implements TVFactory
{
    public TV produceTV()
    {
        System.out.println("海尔电视机工厂生产海尔电视机。");
        return new HaierTV();
    }
}
```

HisenseTV 类：

```
public class HisenseTV implements TV
{
    public void play()
    {
        System.out.println("海信电视机播放中.....");
    }
}
```

HisenseTVFactory :

```
public class HisenseTVFactory implements TVFactory
{
    public TV produceTV()
    {
        System.out.println("海信电视机工厂生产海信电视机。");
        return new HisenseTV();
    }
}
```

TV 类：

```
public interface TV
{
    public void play();
}
```

TVFactory 类：

```
public interface TVFactory
```

```

{
    public TV produceTV();
}

```

XMLUtil 类：

```

import javax.xml.parsers.*;
import org.w3c.dom.*;
import org.xml.sax.SAXException;
import java.io.*;
public class XMLUtil
{
    //该方法用于从 XML 配置文件中提取具体类名，并返回一个实例对象
    public static Object getBean()
    {
        try
        {
            //创建文档对象
            DocumentBuilderFactory dFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder builder =
dFactory.newDocumentBuilder();
            Document doc;
            doc = builder.parse(new File("config.xml"));

            //获取包含类名的文本节点
            NodeList nl =
doc.getElementsByTagName("className");
            Node classNode=nl.item(0).getFirstChild();
            String cName=classNode.getNodeValue();

            //通过类名生成实例对象并将其返回
            Class c=Class.forName(cName);
            Object obj=c.newInstance();
            return obj;
        }
        catch(Exception e)
        {
            e.printStackTrace();
            return null;
        }
    }
}

```