**实验2 创建型设计模式实验**

**一、实验目的**

1．熟练使用面向对象设计原则对系统进行重构；

2．熟练使用PowerDesigner和任意一种面向对象编程语言实现几种常见的创建型设计模式，包括简单工厂模式、工厂方法模式、抽象工厂模式和单例模式，理解每一种设计模式的模式动机，掌握模式结构，学习如何使用代码实现这些模式。

**二、实验内容**

1．在某图形库API中提供了多种矢量图模板，用户可以基于这些矢量图创建不同的显示图形，图形库设计人员设计的初始类图如下所示：



在该图形库中，每个图形类（如Circle、Triangle等）的init()方法用于初始化所创建的图形， setColor()方法用于给图形设置边框颜色，fill()方法用于给图形设置填充颜色，setSize()方法用于设置图形的大小，display()方法用于显示图形。

客户类(Client)在使用该图形库时发现存在如下问题：

① 由于在创建窗口时每次只需要使用图形库中的一种图形，因此在更换图形时需要修改客户类源代码；

② 在图形库中增加并使用新的图形时需要修改客户类源代码；

③ 客户类在每次使用图形对象之前需要先创建图形对象，有些图形的创建过程较为复杂，导致客户类代码冗长且难以维护。

现需要根据面向对象设计原则对该系统进行重构，要求如下：

① 隔离图形的创建和使用，将图形的创建过程封装在专门的类中，客户类在使用图形时无须直接创建图形对象，甚至不需要关心具体图形类类名；

② 客户类能够方便地更换图形或使用新增图形，无须针对具体图形类编程，符合开闭原则。

绘制重构之后的类图并说明在重构过程中所运用的面向对象设计原则。

2．使用**简单工厂模式**设计一个可以创建不同几何形状(Shape)，如圆形(Circle)、矩形(Rectangle)和三角形(Triangle)等的绘图工具类，每个几何图形均具有绘制Draw()和擦除Erase()两个方法，要求在绘制不支持的几何图形时，抛出一个UnsupportedShapeException异常，**绘制类图并编程模拟实现。**

3. 现需要设计一个程序来读取多种不同类型的图片格式，针对每一种图片格式都设计一个图片读取器(ImageReader)，如GIF图片读取器(GifReader)用于读取GIF格式的图片、JPG图片读取器(JpgReader)用于读取JPG格式的图片。图片读取器对象通过图片读取器工厂ImageReaderFactory来创建，ImageReaderFactory是一个抽象类，用于定义创建图片读取器的工厂方法，其子类GifReaderFactory和JpgReaderFactory用于创建具体的图片读取器对象。试使用**工厂方法模式**设计该程序，**绘制类图并编程模拟实现**。需充分考虑系统的灵活性和可扩展性。

4. 某软件公司欲开发一套界面皮肤库，可以对桌面软件进行界面美化。不同的皮肤将提供视觉效果不同的按钮、文本框、组合框等界面元素，其结构如下图所示：



该皮肤库需要具备良好的灵活性和可扩展性，用户可以自由选择不同的皮肤，开发人员可以在不修改既有代码的基础上增加新的皮肤。试使用**抽象工厂模式**设计该皮肤库，绘制类图**并编程模拟实现**。

**三、实验要求**

1. 选择合适的面向对象设计原则对系统进行重构，正确无误地绘制重构之后的类图；

2. 结合实例，正确无误地绘制简单工厂模式、工厂方法模式、抽象工厂模式的模式结构图；

3. 使用任意一种面向对象编程语言实现简单工厂模式、工厂方法模式、抽象工厂模式，代码运行正确无误。

**四、实验步骤**

1. 选择合适的面向对象设计原则对系统进行重构，使用PowerDesigner绘制重构之后的类图；

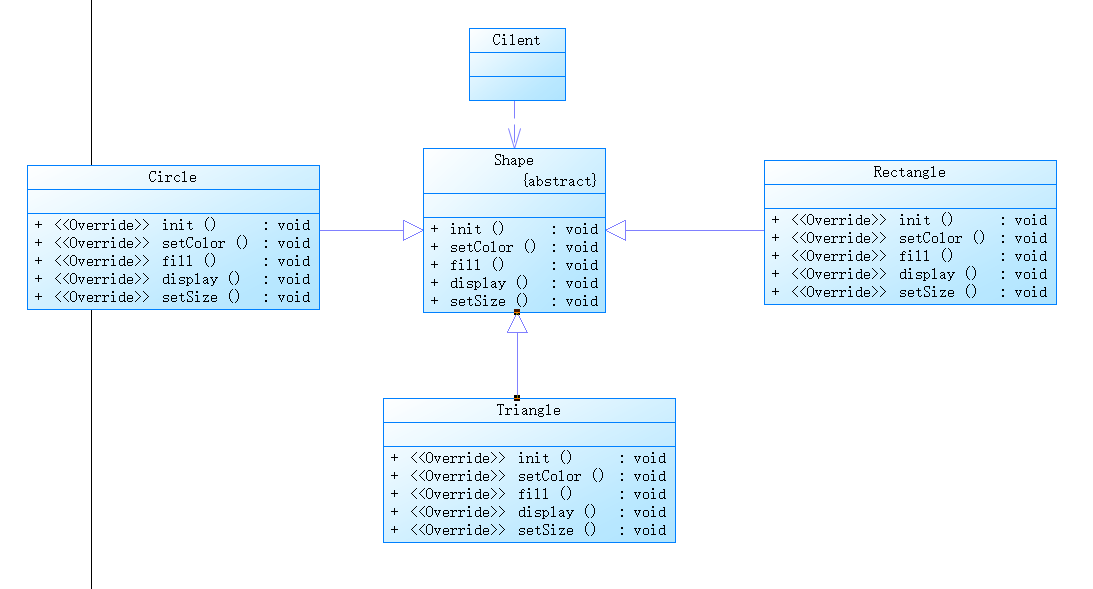
2. 结合实例，使用PowerDesigner绘制简单工厂模式实例结构图并用面向对象编程语言实现该模式实例；

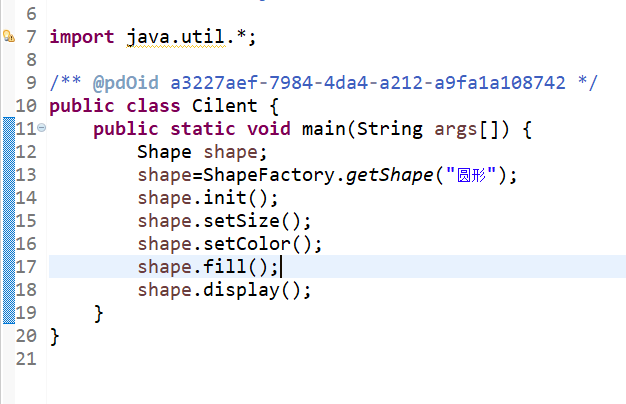
3. 结合实例，使用PowerDesigner绘制工厂方法模式实例结构图并用面向对象编程语言实现该模式实例；

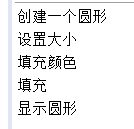
4. 结合实例，使用PowerDesigner绘制抽象工厂模式实例结构图并用面向对象编程语言实现该模式实例；

**五、实验结果**

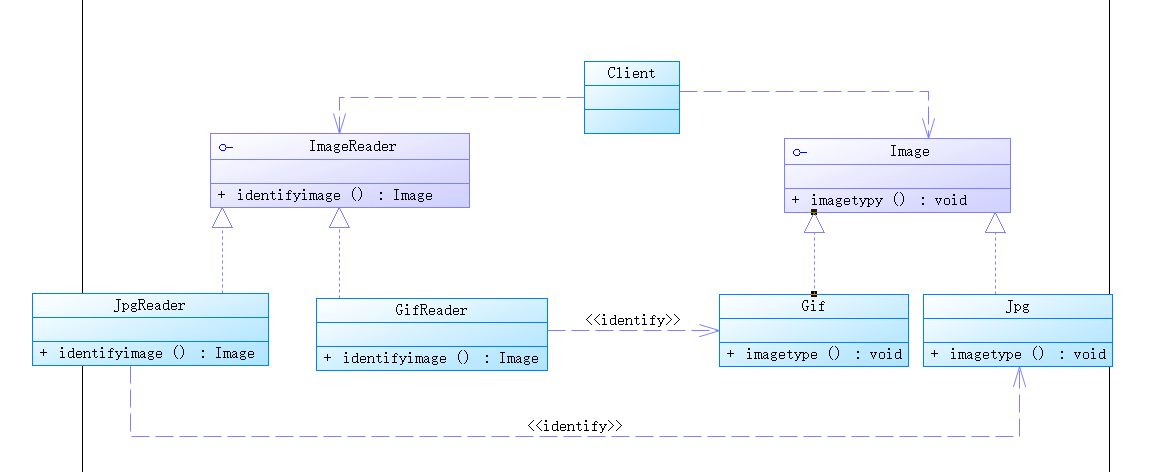
**1.**

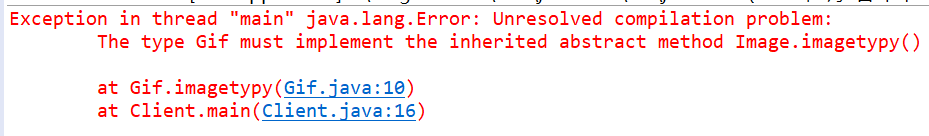




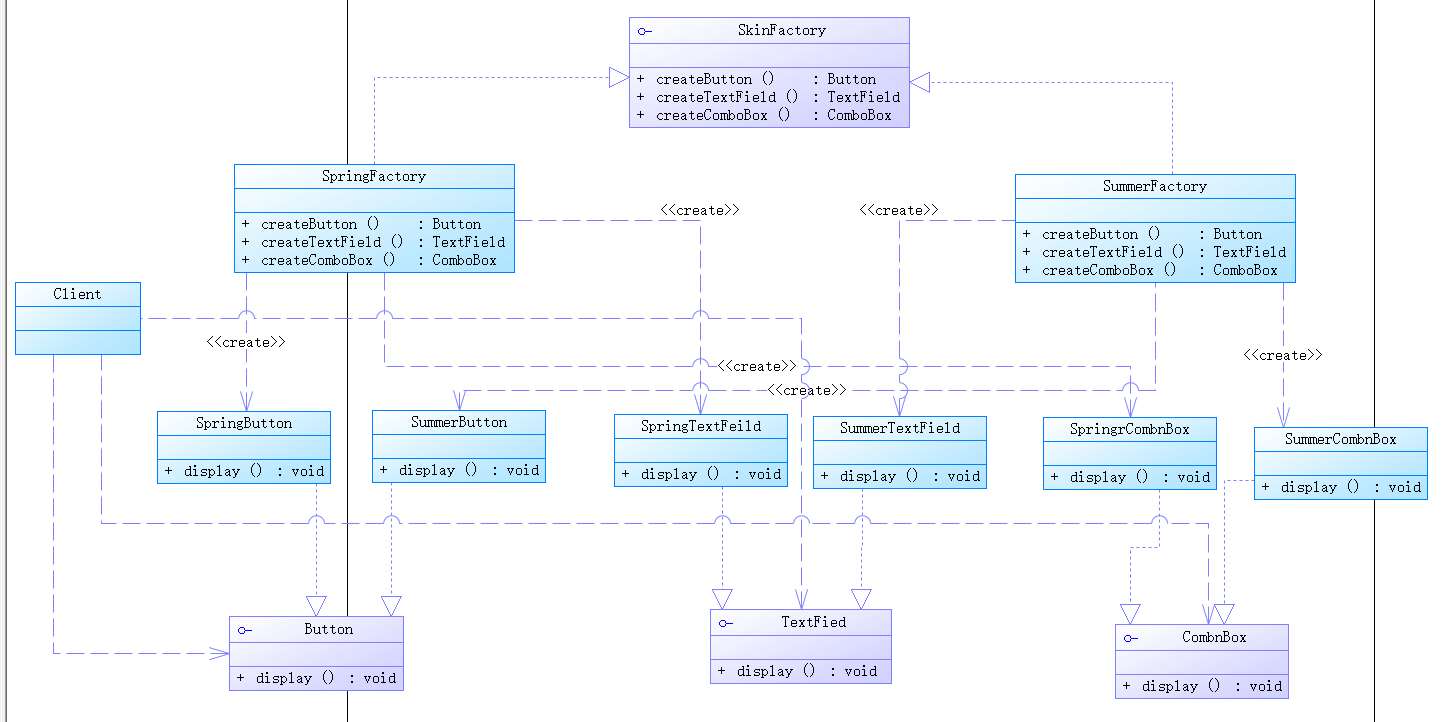


**2．**





**3.**



**六、实验小结**

本次实验把所有关于工厂的设计模式集中绘制。依次用了简单工厂，工厂方法，抽象工厂。实验过程中，简单工厂方法和抽象工厂模式没有出现的问题，但是工厂方法出现异常。主要是具体工厂实现抽象工厂时候，eclipse在具体工厂的implement时，总是报错具体工厂必须为抽象类才能实现，导致程序无法运行。