设计模式实验

一、实验目的

1. 结合实例，熟练绘制设计模式结构图。
2. 结合实例，熟练使用 Java 语言实现设计模式。
3. 通过本实验，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设计模式。

二、实验要求

1. 结合实例，绘制设计模式的结构图。
2. 使用 Java 语言实现设计模式实例，代码运行正确。

三、实验内容

1. 单例模式

某 Web 性能测试软件中包含一个虚拟用户生成器（Virtual User Generator）。为了避免生成的虚拟用户数量不一致，该测试软件在工作时只允许启动唯一一个虚拟用户生成器。采用单例模式设计该虚拟用户生成器，绘制类图并分别使用饿汉式单例、双重检测锁和 IoDH 三种方式编程模拟实现。

2. 原型模式

在某在线招聘网站中，用户可以创建一个简历模板。针对不同的工作岗位，可以复制该简历模板并进行适当修改后，生成一份新的简历。在复制简历时，用户可以选择是否复制简历中的照片：如果选择“是”，则照片将一同被复制，用户对新简历中的照片进行修改不会影响到简历模板中的照片，对模板进行修改也不会影响到新简历；如果选择“否”，则直接引用简历模板中的照片，修改简历模板中的照片将导致新简历中的照片一同修改，反之亦然。现采用原型模式设计该简历复制功能并提供浅克隆和深克隆两套实现方案，绘制对应的类图并编程模拟实现。

3. 简单工厂模式

简单工厂模式使用简单工厂模式设计一个可以创建不同几何形状（Shape）（例如圆形（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方法 draw()和擦除方法 erase()，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。

4. 建造者模式

在某赛车游戏中，赛车包括方程式赛车、场地越野赛车、运动汽车、卡车等类型，不同类型的赛车的车身、发动机、轮胎、变速箱等部件有所区别。玩家可以自行选择赛车类型，系统将根据玩家的选择创建出一辆完整的赛车。现采用建造者模式实现赛车的构建，绘制对应的类图并编程模拟实现。

5. 抽象工厂模式

某系统为了改进数据库操作的性能,用户可以自定义数据库连接对象 Connection 和语句对象 Statement ，针对不同类型的数据库提供不同的连接对象和语句对象，例如提供 Oracle 或 MySQL 专用连接类和语句类，而且用户可以通过配置文件等方式根据实际需要动态更换系统数据库。使用抽象工厂模式设计该系统，绘制对应的类图并编程模拟实现。

四、实验结果

需要提供设计模式实例的结构图（类图）和实现代码。

五、实验小结

请总结本次实验的体会，包括学会了什么、遇到哪些问题、如何解决这些问题以及存在哪些有待改进的地方。