实验二 结构型设计模式实验

一、实验目的

1．综合实例，熟练绘制常见的结构型设计模式结构图。

2. 结合实例，使用 Java 实现常见的结构型设计模式。

3. 通过实验，理解不同结构型设计模式的使用动机，掌握不同结构型设计模式的特点和运用场合，学习如何使用代码实现这些设计模式。

二、实验要求

1．独立完成实验

2．书写实验报告书

三、实验内容

1. 适配器模式的运用

（1）案例背景：

在课堂上我们学习了单向适配器的使用和实现，现在我们需要实现一个双向适配器，编写代码，使用 Java 语言实现双向适配器，使猫可以学狗叫，狗可以学猫抓老鼠，请绘制相应类图并实现。（课本 167 页第三题）

（2）实现步骤：

- 根据题意，画出双向适配器的类图，类图中应该包含一个适配器类 Adapter；两个抽象类 Cat 类和 Dog 类，Cat 类中有发出叫声的方法 cry（）和捉老鼠的方法 catchMouse（），Dog 类中有发出狗叫声的方法 wang（）和动作方法 action（）；两个具体适配者类 ConcreteCat 类

和 ConcreteDog 类，两个抽象类互为抽象目标和抽象适配者，如果客户端针对 Cat 类编程，则 Cat 类充当抽象目标，Dog 类充当抽象适配者，ConcreteCat 类充当具体适配者，反之同理。

- 根据类图，实现上述类的具体代码以及用户类 Client，由于本题中只有一个适配器类 Adapter，所以不需要通过 XML 文件来改变用户类的操作
- 编译并运行代码，观察是否能让猫发出狗叫声和让狗实现抓老鼠的动作。

（3）案例总结：

在以下情况下可以使用适配器模式：

- 系统需要使用一些现有的类，而这些类的接口丌符合系统的需要， 甚至没有这些类的源代码
- 创建一个可以重复使用的类，用于和一些彼此之间没有太大关联 的类，包括一些可能在将来引进的类一起工作

2.组合模式的运用

（1）案例背景：

在操作系统中，一个文件夹中可能存放着图像文件，视频文件，文本文件，也可能存放其他的文件夹，而对不同类型的文件进行的浏览操作也不一样，使用透明组合模式，绘制类图并编程实现文件的浏览（课本 197 页第二题）。

（2）实现步骤：

- 根据题意，画出组合模式的类图，类图中应包含抽象文件类 AbstractFile，具体的图像文件类 ImageFile，视频文件类 VideoFile，文本文件类 TextFile 以及文件夹类 Folder，对每个文件都有 display（）方法，而对文件夹可以进行 add（）方法和 remove（）方法。
- 根据类图，实现上述类的具体代码以及用户类 Client，在用户类中需要将不同类型的文件放入文件夹中。
- 编译并运行程序，使程序能够输出对文件的浏览过程。

（4）案例总结：

在以下情况可以使用组合模式：

- 在具有整体和部分的层次结构中，希望通过一种方式忽略整体与部分的差异，使客户端可以一致的对待他们
- 在使用面向对象语言开发的系统中需要处理一个树形结构
- 在一个系统中能够分离出叶子对象和容器对象，而且他们的类型不固定，需要增加一些新的类型

3. 外观模式的运用

（1）案例背景：

在计算机主机（Mainframe）中，只需要按下主机的开机按钮（on（）），即可调用其他硬件设备和软件的启动方法，如内存（Memory）的自检（check（））、CPU 的运行（run（））、硬盘的（HardDisk）的读取（read（））、操作系统（OS）的载入（load（））等，如果某一过程发生错误则计算机启动失败。使用外观模式模拟该过程，绘制类图并编程实现。（课本 230 页第二题）

（2）实现步骤：

- 根据题意，画出外观模式的类图，使主机类 Mainframe 充当外观角色，内存类 Memory，CPU 类 CPU，硬盘类 HardDisk 和操作系统类 OS 充当子系统角色
- 根据类图，编写并实现代码
- 编译并运行代码，使代码能够输出模拟出来的电脑开机过程

（3）案例总结：

在以下情况下可以使用外观模式：

- 当要为一个复杂子系统提供一个简单接口的时候可以使用外观模式，该接口可以满足大部分用户需求，用户也可以越过外观类直接访问子系统
- 客户程序和多个子系统之间存在很大的依赖性。引入外观类将子系统与客户以及其他子系统解耦，可以提高子系统的独立性和可移植性。
- 在层次化结构中，可以使用外观模式定义系统中每一层的入口，层与层之间不直接产生联系，而是通过外观类建立联系，降低层之间的耦合度。