实验一 创建型设计模式实验

一、实验目的

1．综合实例，熟练绘制常见的创建型设计模式结构图。

2. 结合实例，熟练掌握不同创建型设计模式的特点并能够根据不同需求使用不同的创建型设计模式

3. 通过实验，熟练掌握不同创建型设计模式代码的编写

二、实验要求

1．独立完成实验

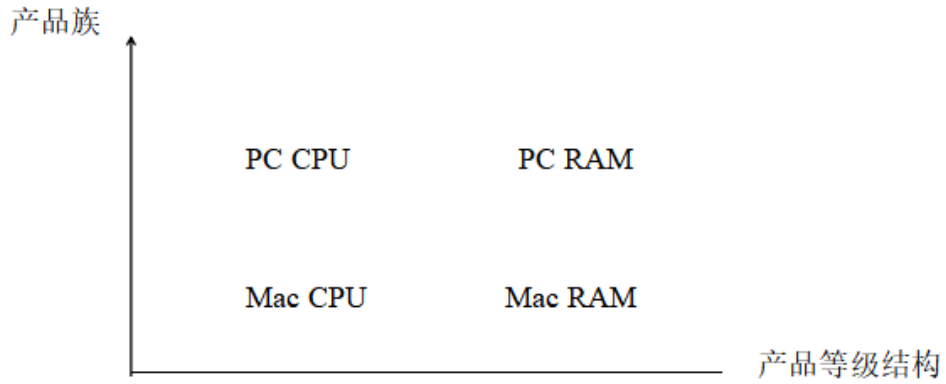
2．书写实验报告书

三、实验内容

1. 抽象工厂模式的运用

（1）案例背景：

计算机包含内存（RAM），CPU 等硬件设备，根据如图所示的“产品等级结构-产品族示意图”，使用抽象工厂模式实现计算机设备创建过程并绘制类图（课本 105 页第二题）



（2）实现步骤：

- 根据题意，使用抽象工厂模式并画出类图，类图中应包含一个抽象工厂类 AbstractFactory，PcFactory 和 MacFactory 两个具体工厂，CPU，RAM 两个抽象产品类，PcCPU，MacCPU，PcRAM，MacRAM 四个具体产品类。
- 根据类图，实现上述类的具体代码以及用户类 Client 和辅助类 XMLUtil 以实现通过 XML 文件来制造不同的产品
- 更改 XML 中的属性，观察用户类是否能使用不同的产品

（3）案例总结：

抽象工厂模式主要适用于：

- 一个系统不应当依赖于产品类实例如何被创建、组合和表达的细节
- 系统中有多于一个的产品族，但每次只使用其中某一产品族
- 属于同一个产品族的产品将在一起使用，这一约束必须在系统的设计中体现出来
- 产品等级结构稳定，在设计完成之后不会向系统中增加新的产品等级结构或者删除已有的产品等级结构

2.建造者模式的运用

（1）案例背景：

计算机组装工厂可以将 CPU，内存，硬盘，主机，显示器等硬件设备组装在一起构成一台完整的计算机，且构成的计算机可以是笔记本电脑，也可以是台式机，还可以是不提供显示器的服务器主机。对于用户来言，无需关心计算机的组成设备和组装过程，工厂返回给用户的是完整的计算机对象。所以我们可以使用建造者模式来实现计算机的组成过程，请绘制出类图并编程实现

（2）实现步骤：

- 根据题意，使用建造者模式并画出类图。类图中应包含抽象建造者类 ComputerBuilder，复合产品类 Computer，具体建造者类 Notebook，Desktop 和 Server，其中台式机和服务器主机使用相同的 CPU，内存，硬盘和主机，但是服务器不包含显示器，而笔记本使用自己独自的一套硬件设备。此外还需要指挥者类 ComputerAssembleDirector，此类中应有将硬件设备组合在一起的建造方法 assemble()并返回用户需要的具体计算机

- 根据类图，实现上述类的具体代码以及用户类 Client 和辅助类 XMLUtil 以实现通过 XML 文件来制造不同的计算机
- 更改 XML 中的属性，观察用户类是否能够获取到不同的计算机以及这些计算机的组装是否符合要求

（3）案例总结：

- 建造者模式主要运用于：
- 需要生成的产品对象有复杂的内部结构，这些产品对象通常包含多个成员变量
- 需要生成的产品对象的属性相互依赖，需要指定其生成顺序
- 对象的创建过程独立于创建该对象的类。在建造者模式中通过引入了指挥者类，将创建过程封装在指挥者类中，而不在建造者类和客户类中
- 隔离复杂对象的创建和使用，并使得相同的创建过程可以创建不同的产品

3. 单例模式的运用

（1）案例背景：

在实际的运用中，我们有时一个类不止需要产生一个对象，可能需要两个或者三个。在课上我们讲过，使用单例模式的思想可以实现多例模式，从而确保系统中某个类的对象只能存在有限个，请设计并实现代码，从而实现多例模式

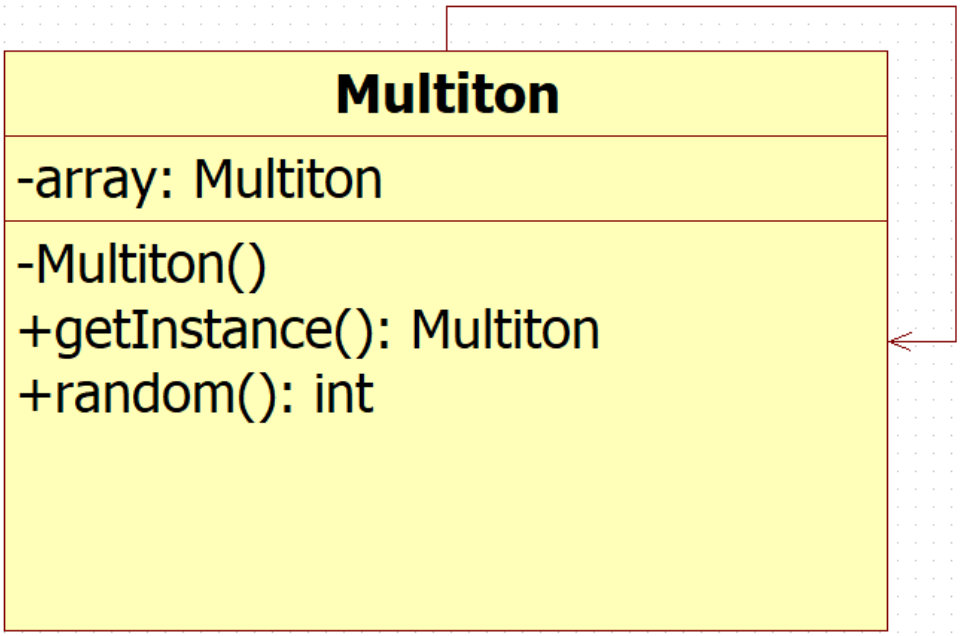
（2）实现步骤：

- 由于本题的实现较为复杂，所以我们直接给出参考类图（见下一页）
- 根据类图，实现多例模式的代码

（3）案例总结：

单例模式主要适用于：

- 系统只需要一个实例对象，或者因为资源消耗太大而只允许创建一个对象
- 客户调用类的单个实例只允许使用一个公共访问点，除了该公共访问点，不能通过其他途径访问该实例
- 根据不同的需求，也可以按照单例模式的思想来实现多例模式


Multiton
-array: Multiton
-Multiton()
+getInstance(): Multiton
+random(): int