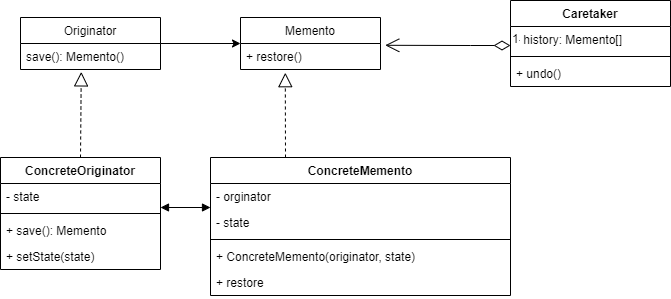
# 装配

## 备忘录模式

### 1.1 UML图



### 1.2 说明

备忘录模式的使用：在装配环节，记录装配前的状态和装配后的状态，若装配后有故障，则按照备忘录返回到上一步，直到状态正常。

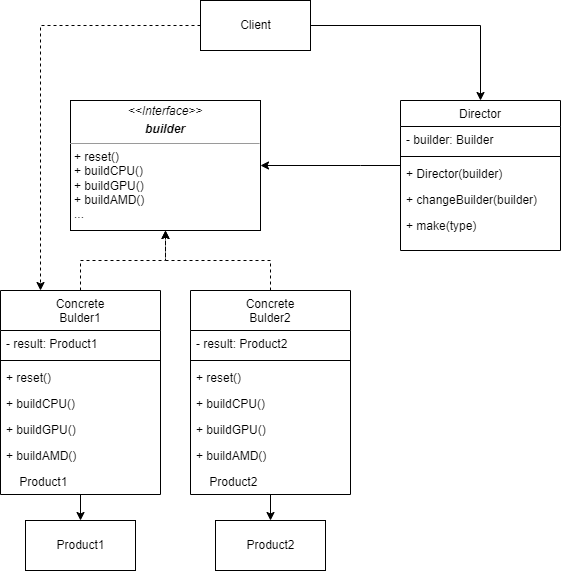
允许存在多种不同类型的原发器和备忘录。 每种原发器都和其相应的备忘录类进行交互。 原发器和备忘录都不会将其状态暴露给其他类。

负责人对于备忘录的成员变量和方法的访问权限非常有限： 它们只能在栈中保存备忘录， 而不能修改其状态。

每个备忘录将与创建了自身的原发器连接。 原发器会将自己及状态传递给备忘录的构造函数。 由于这些类之间的紧密联系， 只要原发器定义了合适的设置器 （setter）， 备忘录就能恢复其状态。

## 建造者模式

### 2.1 UML图



### 2..2说明

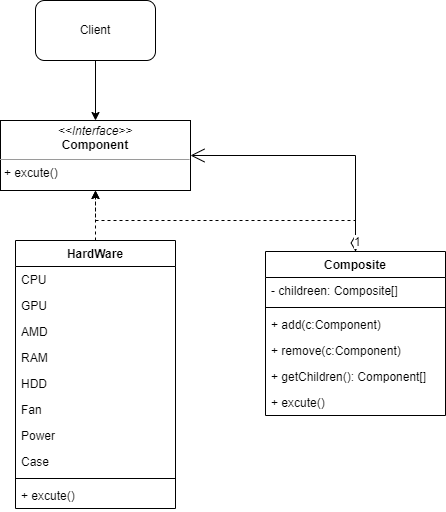
将计算机的装配过程划分为一组步骤， 比如 install-CPU和 install-GPU等每次创建对象时，需要通过生成器对象执行一系列步骤。

重点在于无需调用所有步骤， 而只需调用创建特定对象配置所需的那些步骤即可。

当需要创建不同样式的计算机时， 其中的一些构造步骤可能需要不同的实现。 例如， 有些电脑的CPU配置可能更高，。在这种情况下， 创建多个不同的生成器， 用不同方式实现一组相同的创建步骤。 然后就可以在创建过程中使用这些生成器 （例如按顺序调用多个构造步骤） 来生成不同类型的对象。

## 组合模式

### UML图

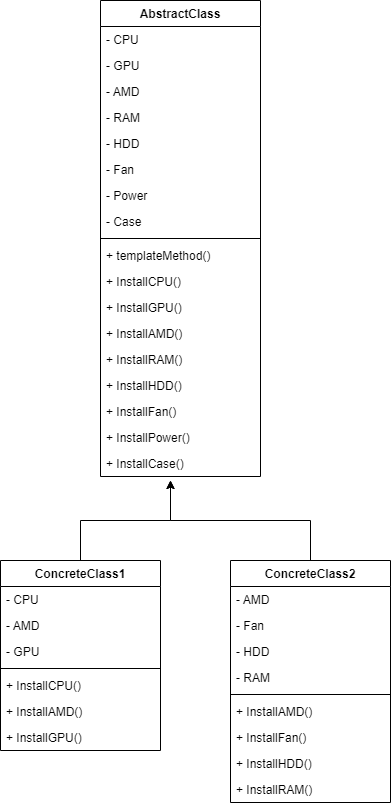


### 说明

装配电脑时，每个组件是对象，电脑是一个整体对象。每台电脑包括若干硬件和软件， 硬件由CPU、显卡等构成， 软件由操作系统、应用软件等组成。指令由用户下达， 通过每个层级传递， 直到每个硬件都知道执行应该完成的指令。

## 模板方法模式

### UML图



### 说明

模板方法用于装配电脑。 标准电脑装配方案中可提供几个扩展点， 允许潜在使用者调整电脑的部分细节。

每个装配步骤 （例如安装CPU、GPU、显卡等） 都能进行微调， 这使得组装成的电脑的性能会略有不同。

具体实现为将所有步骤声明为抽象类型，强制要求子类自行实现这些方法。在电脑的装配过程中，子类中已有所有必要的实现，因此只需调整这些方法的签名，使之与超类的方法匹配即可。