**1、Builder模式**

**主要作用：**Builder模式的主要使用就是在参数较多并且可能有多个默认值时使用，是一种创建型设计模式，在参数较多的情况下有两点问题（1）参数较多较为混乱，前后顺序有可能出现问题，这种问题往往不易发现；（2）部分参数有默认值那么就需要重载多个构造函数，容易混乱并且不同的参数往往需要传入多个null。

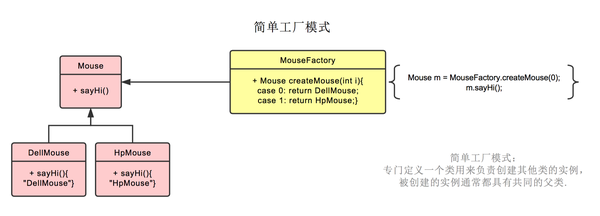
**构建方式：**使用静态内部类来构建，内部具有一个参数类或者参数值来进行参数的设置，注意设置函数（setXXX）必须返回this，用于链式调用，最终提供一个create函数来创建对象。

**2、工厂方法模式**

**主要作用：**定义一个用于创建对象的接口(也可以使抽象类)，让**子类决定**实例化哪一个类。工厂模式的主要区别在于工厂模式的三个类型。

**（1）简单工厂模式**

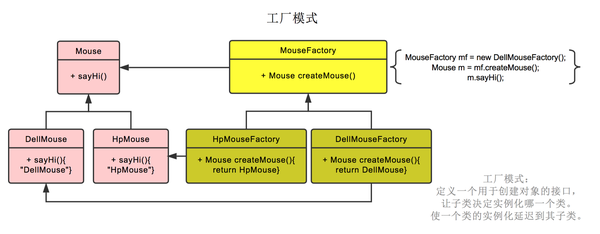
简单工厂模式不是23种里的一种，简而言之，就是有一个专门生产某个产品的类。比如下图中的鼠标工厂，专业生产鼠标，给参数0，生产戴尔鼠标，给参数1，生产惠普鼠标。



简单工厂模式的主要作用就是屏蔽产品类的实现细节，产品实现的细节如何变化不是客户代码所关心的部分，并且客户代码的更改也只需要变化给定的参数即可。不过这不是推荐设计模式中的一种因为最重要的一点是它不符合开闭原则，如果要增加一个产品，我们需要对工厂类的实现进行大量的修改。

**（2）工厂方法模式**

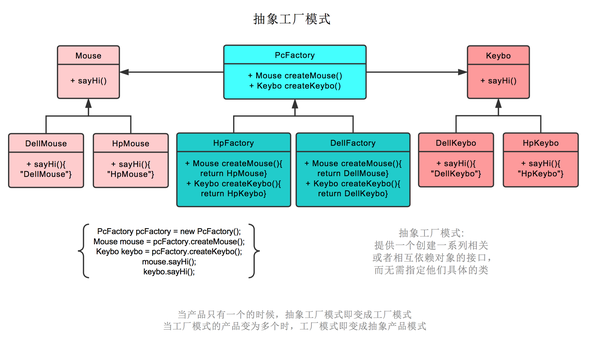
工厂模式也就是鼠标工厂是个父类，有生产鼠标这个接口，这个抽象工厂使我们所定义的工厂的基类，戴尔鼠标工厂，惠普鼠标工厂继承它，可以分别生产戴尔鼠标，惠普鼠标。 生产哪种鼠标不再由参数决定，而是创建鼠标工厂时，由戴尔鼠标工厂创建。后续直接调用鼠标工厂.生产鼠标()即可。



工厂模式的好处是如果要增加一个新的鼠标，只需要实现具体的鼠标工厂，而不需要修改已有的工厂类，符合开闭原则。也减少了具体实现类之间的耦合，我们只需要使用我们希望使用的工厂类即可，不需要使用我们用不着的。

**（3）抽象工厂方法模式**

抽象工厂模式也就是不仅生产鼠标，同时生产键盘。也就是PC厂商是个父类，有生产鼠标，生产键盘两个接口。戴尔工厂，惠普工厂继承它，可以分别生产戴尔鼠标+戴尔键盘，和惠普鼠标+惠普键盘。创建工厂时，由戴尔工厂创建。后续工厂.生产鼠标()则生产戴尔鼠标，工厂.生产键盘()则生产戴尔键盘。



抽象工厂模式主要针对的是为创建一组相关的对象的接口，比如鼠标键盘它们在某种意义上来说对于具体实现工厂类都是属于一个品牌的，应该把它们放在同一个抽象工厂接口中。这也是它与工厂模式的主要区别，这里的产品是具有相关性的，比如Android中的Button和TextView与IOS中的Button和TextView。

**3、享元模式**

**主要作用：**享元模式是对象池的一种实现，主要使用是在系统中存在大量相似对象时，对对象进行内部的复用，避免对象被频繁创建和销毁从而消耗额外的系统资源的一种设计模式。

**构建方式：**使用一个Map来进行缓存，它的键就是享元对象的内部状态，值就是对象本身。当需要获取一个享元对象时，优先使用缓存，否则创建一个并放入容器中。Android中并没有采用这种模式，因为考虑到状态的变化太过复杂不一定缓存中有完全一致的，所以采用将废弃的对象保存在一个链表中，当需要时直接取出进行相应的设置，起到了对象复用的效果。

**4、代理模式**

**主要作用：**代理模式也叫委托模式，是一种结构性设计模式，它的作用非常广泛，主要的作用就是对被代理对象进行访问控制，对于被代理对象的访问需要通过代理类来进行，从而进行需求上的改变。

**构建方式：**

（1）远程代理：Binder的代理模式就是使用此方式，为某个对象在不同内存地址空间提供局部代理，屏蔽了跨进程通信的具体细节，这样Client端不需要考虑Server的存在；

（2）保护代理：使用代理控制对原始对象的访问，常常用于对原始对象有不同访问权限的地方。

（3）智能引用：使用代理来访问某个对象，并在代理中对对象进行引用计数来维护对象的声明周期。

**5、装饰模式**

**主要作用：**动态地给对象添加一些额外的职责，可以透明地动态地扩展类的功能。装饰模式的特点就是对于某个功能（某个函数）进行功能扩展并且不影响它本身的实现。比如IO里面的IO类可以不影响构造函数中传入的对象的实现，透明地扩展其功能。这个逻辑的具体过程就是在装饰类中实现的，而任何一个扩展类都要继承自装饰类。

**构建方式：**定义一个抽象组件类，抽象装饰者类实现装饰的逻辑，通过耦合方式调用构造方法中传入的组件类的方法，扩展功能类都继承自抽象装饰类并且在扩展的方法中调用父类的方法。

6、