

设计

设计三大原则

1. 面向接口编程,而非实现
 - 接口：父类的虚函数,抽象基类,函数指针
 - 实现：具体的子类
2. 多用组合,而非继承
 - 组合：

```
1 class A {  
2     ...  
3     B b;    // class B object  
4     ...  
5 };
```

- 继承：

```
1 class A : public B {  
2     ...  
3 };
```

3. 封装变化
 - 将相同的东西封装起来

工厂模式

定义

简单工厂模式(Simple Factory Pattern)：又称为静态工厂方法(Static Factory Method)模式，它属于类创建型模式。在简单工厂模式中，可以根据参数的不同返回不同类的实例。简单工厂模式专门定义一个类来负责创建其他类的实例，被创建的实例通常都具有共同的父类。

结构

- **Factory：工厂角色**

工厂角色负责实现创建所有实例的内部逻辑

- **Product：抽象产品角色**

抽象产品角色是所创建的所有对象的父类，负责描述所有实例所共有的公共接口

- **ConcreteProduct：具体产品角色**

具体产品角色是创建目标，所有创建的对象都充当这个角色的某个具体类的实例。

class SimpleFactory

```
if (arg == "A")
{
    return new ConcreteProductA();
}
else if(arg == "B")
{
    return new ConcreteProductB();
}
```

