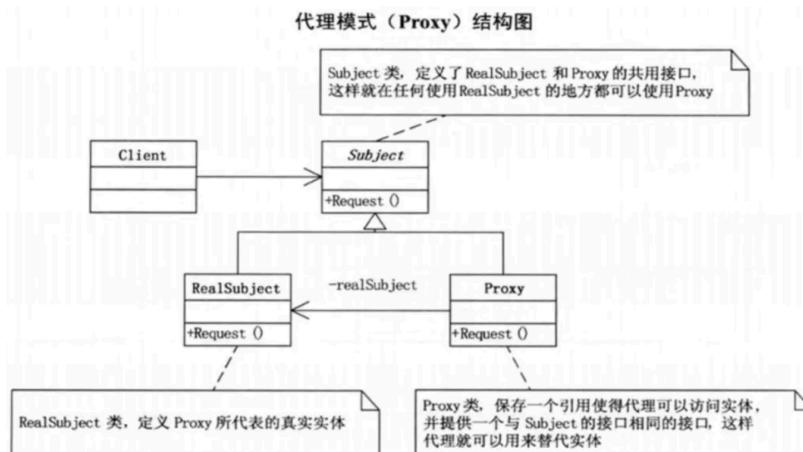


动机:在面向对象系统中,有些对象由于某种原因(比如对象创建的开销很大,或者某些操作需要安全控制,或者需要进程外的访问等),直接访问会给使用者,或者系统结构带来很多麻烦。如何在不失去透明操作对象的同时来管理/控制这些对象物有的复杂性? 增加一层间接层是软件开发中常见的解决方式。

定义:为其他对象提供一种代理以控制(隔离,使用接口)对这个对象的访问。 <设计模式> GoF

要点总结:增加一层间接层是软件系统中对许多复杂问题的一种常见解决方法。在面向对象的系统中,直接使用某些对象会带来很多问题。作为间接层的proxy对象便是解决这一问题的常用手段。具体proxy设计模式的实现方法,实现粒度都相差很大,有些可能对单个对象做细粒度的控制。如copy-on-write技术,有些可能对组件模块提供抽象代理层,在架构层次对对象做proxy。proxy并不一定要求保持接口完整一致性,只要能够实现间接控制有时候损及一些透明性是可以接受的。



角色定义:

Subject抽象主体角色, 抽象类或者接口, 是一个普通的业务类型定义

RealSubject具体主体角色, 也叫作被委托角色, 被代理角色。业务逻辑的具体执行者

Proxy代理主体角色, 委托类, 代理类。

```
1 #include<iostream>
2 using namespace std;
3 class Subject{
4 public:
5     virtual void Request() = 0; //公共接口
6     virtual ~Subject(){}
7 };
8 class RealSubject : public Subject {
9 public:
10     void Request() {
11         cout << "I am RealSubject" << endl;
12     }
13 };
14 class Proxy : public Subject{
15 private:
16     RealSubject* realSubject; //代理通过该指针访问实体
17 public:
18     void Request(){
19         if(NULL == realSubject){
20             realSubject = new RealSubject();
21         }
22         realSubject->Request();
23     }
24 };
25 int main(){
```

```

26     Proxy* p = new Proxy();
27     p->Request();
28     delete p;
29     return 0;
30 }

```

```

1  #include <iostream>
2  using namespace std;
3
4  class Cloth{ //服装生立商,相当于Subject
5  public:
6      virtual void MakeCloth(const string strClothType) = 0;
7      virtual ~Cloth(){}
8  };
9  class ManCloth : public Cloth{//=realSubject
10 public:
11     void MakeCloth(string strType){
12         cout << "Make " << strType << " Cloth" << endl;
13     }
14 };
15 class WoManCloth : public Cloth{//=realSubject
16 public:
17     void MakeCloth(string strType){
18         cout << "Make " << strType << " Cloth" << endl;
19     }
20 };
21 class ClothWholeSale : public Cloth{//服装批发商=Proxy
22 private:
23     ManCloth *pMan;
24     WoManCloth *pWoman;
25 public:
26     ~ClothWholeSale(){
27         delete pMan;
28         delete pWoman;
29     }
30     void MakeCloth(string strType){
31         if(strType == "Man"){
32             if(NULL == pMan){
33                 pMan = new ManCloth();
34             }
35             pMan->MakeCloth(strType);
36         }
37         if(strType == "WoMan"){
38             if(NULL == pWoman){
39                 pWoman = new WoManCloth();
40             }
41             pWoman->MakeCloth(strType);
42         }
43     }
44 };
45 int main(){
46     ClothWholeSale *pSale = new ClothWholeSale();
47     pSale->MakeCloth("Man");

```

```
48     pSale->MakeCloth("WoMan");
49     return 0;
50 }
```

```
192:DesignPattnsStudy weishichun$ g++ -o Proxy1.out Proxy_1.cpp
192:DesignPattnsStudy weishichun$ ./Proxy1.out
I am RealSubject
192:DesignPattnsStudy weishichun$ g++ -o Proxy2.out Proxy_2.cpp
192:DesignPattnsStudy weishichun$ ./Proxy2.out
Make Man Cloth
Make WoMan Cloth
192:DesignPattnsStudy weishichun$
```