

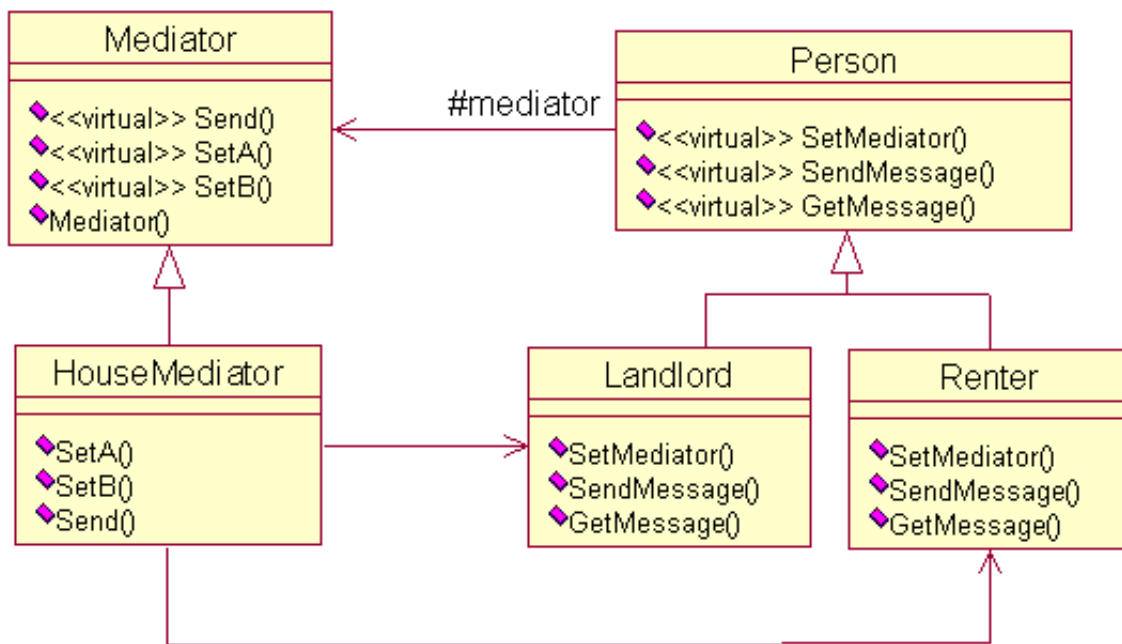
設計模式C++實現（13）——中介者模式

星期六, 2013 12月 14, 1:10 上午

軟件領域中的設計模式為開發人員提供了一種使用專家設計經驗的有效途徑。設計模式中運用了面向對象編程語言的重要特性：封裝、繼承、多態，真正領悟設計模式的精髓是可能一個漫長的過程，需要大量實踐經驗的積累。最近看設計模式的書，對於每個模式，用C++寫了個小例子，加深一下理解。主要參考《大話設計模式》和《設計模式：可復用面向對象軟件的基礎》兩本書。本文介紹中介者模式的實現。

中介者模式：用一個中介對象來封裝一系列的對象交互。中介者使各對象不需要顯式地相互引用，從而使其耦合鬆散，而且可以獨立地改變它們之間的交互。中介者模式的例子很多，大到聯合國安理會，小到房屋中介，都扮演了中間者的角色，協調各方利益。

本文就以租房為例子，如果沒有房屋中介，那麼房客要自己找房東，而房東也要自己找房客，非常不方便。有了房屋中介機構就方便了，房東可以把要出租的房屋信息放到中介機構，而房客可以去中介機構諮詢。在軟件中，就是多個對象之間需要通信，如果沒有中介，對象就需要知道其他對象，最壞情況下，可能需要知道所有其他對象，而有了中介對象就方便多了，對象只需與中介對象通信，而不用知道其他的對象。這就是中介者模式，下面以租房為例，給出中介者模式的UML圖。



實現不難，下面給出C++的實現：

```

1. class Mediator;
2. //抽象人
3. class Person
4. {
5. protected:
6.     Mediator *m_mediator; //中介
7. public:
8.     virtual void SetMediator(Mediator *mediator){} //設置中介
9.     virtual void SendMessage(string message) {} //向中介發送信息
10.    virtual void GetMessage(string message) {} //從中介獲取信息
11. };
  
```

```

12. //抽象中介機構
13. class Mediator
14. {
15. public:
16.     virtual void Send(string message, Person *person) {}
17.     virtual void SetA(Person *A) {} //設置其中一方
18.     virtual void SetB(Person *B) {}
19. };
20. //租房者
21. class Renter: public Person
22. {
23. public:
24.     void SetMediator(Mediator *mediator) { m_mediator = mediator; }
25.     void SendMessage(string message) { m_mediator->Send(message, this); }
26.     void GetMessage(string message) { cout<<"租房者收到信息"<<message; }
27. };
28. //房東
29. class Landlord: public Person
30. {
31. public:
32.     void SetMediator(Mediator *mediator) { m_mediator = mediator; }
33.     void SendMessage(string message) { m_mediator->Send(message, this); }
34.     void GetMessage(string message) { cout<<"房東收到信息："<<message; }
35. };
36. //房屋中介
37. class HouseMediator : public Mediator
38. {
39. private:
40.     Person *m_A; //租房者
41.     Person *m_B; //房東
42. public:
43.     HouseMediator(): m_A(0), m_B(0) {}
44.     void SetA(Person *A) { m_A = A; }
45.     void SetB(Person *B) { m_B = B; }
46.     void Send(string message, Person *person)
47.     {
48.         if(person == m_A) //租房者給房東發信息
49.             m_B->GetMessage(message); //房東收到信息
50.         else
51.             m_A->GetMessage(message);
52.     }
53. };

```

客戶使用方式如下：

```

1. //測試案例
2. int main()
3. {
4.     Mediator *mediator = new HouseMediator();
5.     Person *person1 = new Renter(); //租房者
6.     Person *person2 = new Landlord(); //房東
7.     mediator->SetA(person1);
8.     mediator->SetB(person2);
9.     person1->SetMediator(mediator);
10.    person2->SetMediator(mediator);
11.    person1->SendMessage("我想在南京路附近租套房子，價格800元一個月\n");
12.    person2->SendMessage("出租房子：南京路100號，70平米，1000元一個月\n");

```

```
13. delete person1; delete person2; delete mediator;  
14. return 0;  
15. }
```