Chain of Responsibility 패턴

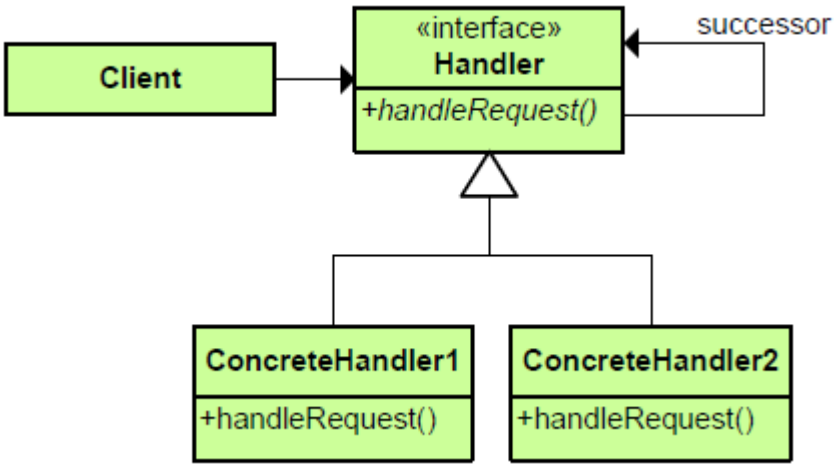
타입

Behavioral Pattern

문제

해결

클래스 다이어그램



예제

chainofresponsibility.cpp

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;

class Base
{
    Base *next; // 1. "next" pointer in the base class
  public:
    Base()
    {
        next = 0;
    }
```

```
    void setNext(Base *n)
    {
        next = n;
    }
    void add(Base *n)
    {
        if (next)
          next->add(n);
        else
          next = n;
    }
    // 2. The "chain" method in the base class always delegates to the
next obj
    virtual void handle(int i)
    {
        next->handle(i);
    }
};

class Handler1: public Base
{
  public:
     /*virtual*/void handle(int i)
    {
        if (rand() % 3)
        {
            // 3. Don't handle requests 3 times out of 4
            cout << "H1 passsed " << i << "  ";
            Base::handle(i); // 3. Delegate to the base class
        }
        else
          cout << "H1 handled " << i << "  ";
    }
};

class Handler2: public Base
{
  public:
     /*virtual*/void handle(int i)
    {
        if (rand() % 3)
        {
            cout << "H2 passsed " << i << "  ";
            Base::handle(i);
        }
        else
          cout << "H2 handled " << i << "  ";
    }
};
```

```
class Handler3: public Base
{
  public:
     /*virtual*/void handle(int i)
    {
        if (rand() % 3)
        {
            cout << "H3 passsed " << i << "  ";
            Base::handle(i);
        }
        else
          cout << "H3 handled " << i << "  ";
    }
};

int main()
{
  srand(time(0));
  Handler1 root;
  Handler2 two;
  Handler3 thr;
  root.add(&two);
  root.add(&thr);
  thr.setNext(&root);
  for (int i = 1; i < 10; i++)
  {
    root.handle(i);
    cout << '\n';
  }
}
```

참고

http://en.wikibooks.org/wiki/C%2B%2B_Programming/Code/Design_Patterns#Chain_of_Responsibility