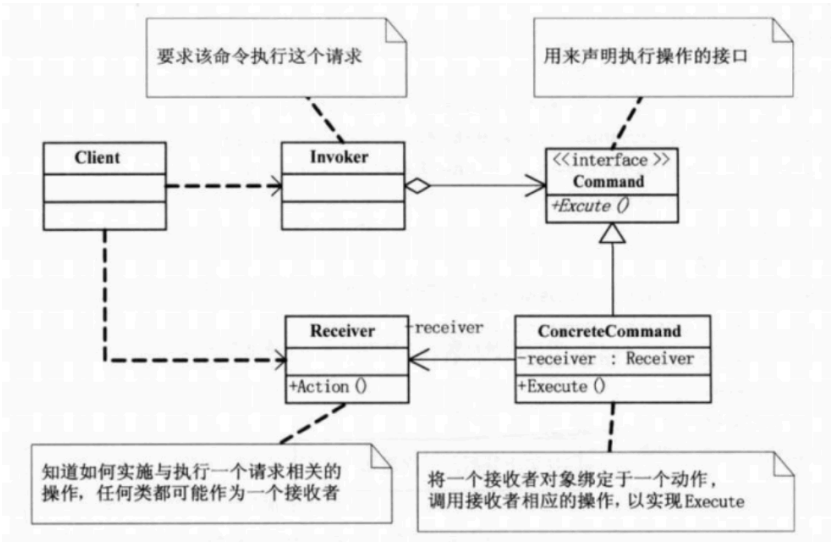
定义:将一个请求(行为)封装为一个对象,从而使你可用不同的请求对客户进行参数化, 对请求排队或记录请求日志, 以及支持可撤销操作. ---<设计模式>GoF



Command：声明执行操作的接口；

ConcreteCommand：将一个接收者对象绑定于一个动作，之后，调用接收者相应的操作，以实现Execute来完成相应的命令；

Client：创建一个具体命令对象，但是并没有设定它的接收者；

Invoker：要求该命令执行这个请求；

Receiver：知道如何实施与执行一个请求相关的操作，任何类都可能作为一个接收者。

```
#include <iostream>
#include <vector>
using namespace std;
class Reciver{
public:
    virtual ~Reciver(){}
    virtual void Open()=0;
    virtual void Close()=0;
};
class TvRecier:public Reciver{
public:
    void Open(){cout << "开电视" << endl;}
    void Close(){cout << "关电视" << endl;}
};
class Command{
protected:
    Reciver* pReciver;
public:
    Command(Reciver* r):pReciver(r){}
    virtual ~Command(){}
    virtual void Excute()=0;
};
class OpenCommand : public Command{
public:
    OpenCommand(Reciver* r):Command(r){}
    void Excute(){
```

```
        pReciver->Open();
    }
};
class CloseCommand : public Command{
public:
    CloseCommand(Reciver* r):Command(r){}
    void Excute(){
        pReciver->Close();
    }
};
class RemoteControl{
public:
    void Excute(Command * pCmd){
        vecpCmd.push_back(pCmd);
        pCmd->Excute();
    }
private:
    vector<Command*> vecpCmd;
};
int main(){
    Reciver *pR = new TvRecier();
    Command *pOpen = new OpenCommand(pR);
    Command *pClose = new CloseCommand(pR);
    RemoteControl *pRemoteCtrl = new RemoteControl();
    pRemoteCtrl->Excute(pOpen);
    pRemoteCtrl->Excute(pClose);
    delete pRemoteCtrl;
    delete pClose;
    delete pOpen;
    delete pR;
    return 0;
}
```

```
weishichundembp:DesignPattnsStudy weishichun$ g++ -o C.out Command_1.cpp
weishichundembp:DesignPattnsStudy weishichun$ ./C.out
开电视
关电视
weishichundembp:DesignPattnsStudy weishichun$
```