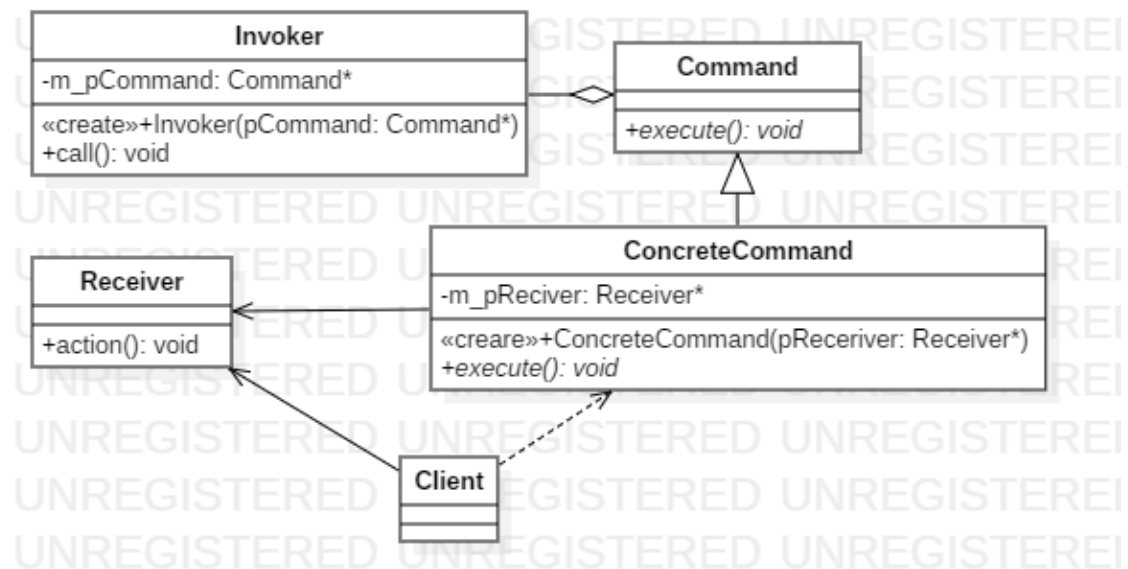


模式定义

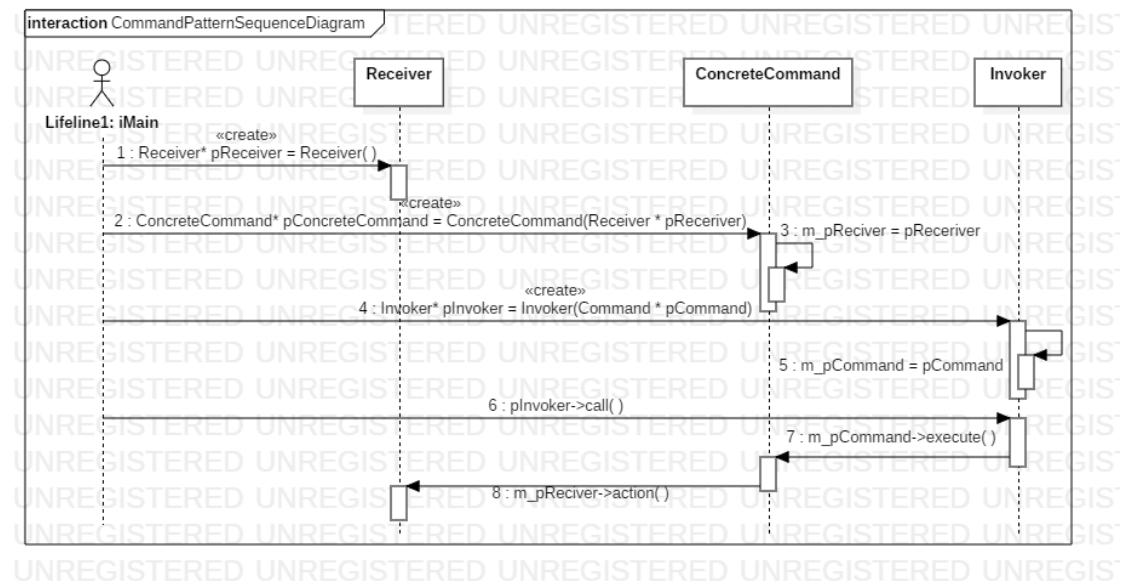
命令模式：将一个请求封装为一个对象，从而使我们可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。命令模式是一种对象行为型模式，其别名为动作模式或者事务模式。

模式结构

- Command: 抽象命令类
- ConcreteCommand: 具体命令类
- Invoker: 调用者
- Receiver: 接收者
- Client: 客户类



时序图



关键代码

```
int main()
{
    Receiver* pReceiver = new Receiver();
    ConcreteCommand* pConcreteCommand = new ConcreteCommand(pReceiver);
    Invoker* pInvoker = new Invoker(pConcreteCommand);

    pInvoker->call();

    delete pReceiver;
    delete pConcreteCommand;
    delete pInvoker;

    system("pause");
    return 0;
}

ConcreteCommand::ConcreteCommand(Receiver * pReceiver)
{
    m_pReciver = pReceiver;
}

Invoker::Invoker(Command * pCommand)
{
    m_pCommand = pCommand;
}

void Invoker::call()
{
    cout << "Invoker call." << endl;
    m_pCommand->execute();
}
```

```

void ConcreteCommand::execute()
{
    cout << "ConcreteCommand execute." << endl;
    m_pReceiver->action();
}

void Receiver::action()
{
    cout << "receiver action." << endl;
}

```

测试结果

```

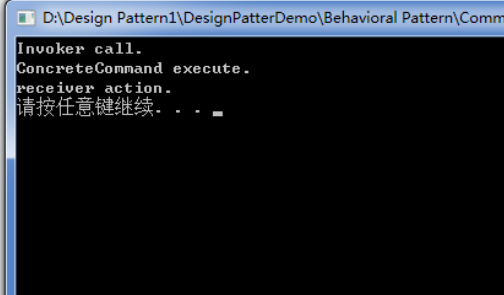
using namespace std;
int main()
{
    Receiver* pReceiver = new Receiver();
    ConcreteCommand* pConcreteCommand = new ConcreteCommand(pReceiver);
    Invoker* pInvoker = new Invoker(pConcreteCommand);

    pInvoker->call();

    delete pReceiver;
    delete pConcreteCommand;
    delete pInvoker;

    system("pause");
    return 0;
}

```



```

D:\Design Pattern1\DesignPatterDemo\Behavioral Pattern\Comm
Invoker call.
ConcreteCommand execute.
receiver action.
请按任意键继续. . .

```