

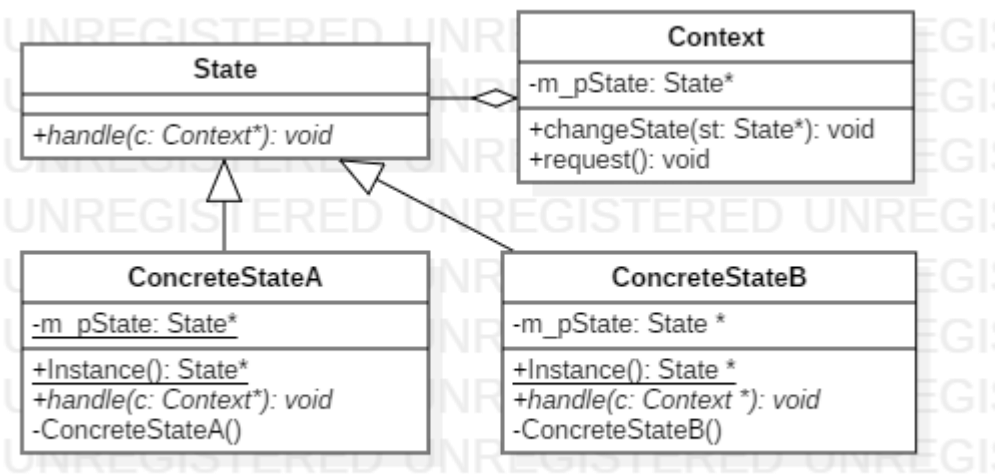
# 模式定义

**状态模式：** 允许一个对象在其内部状态改变时改变他的行为，对象开起来似乎修改了他的类。其别名为状态对象（Object for states），状态模式是一种对象行为型模式。

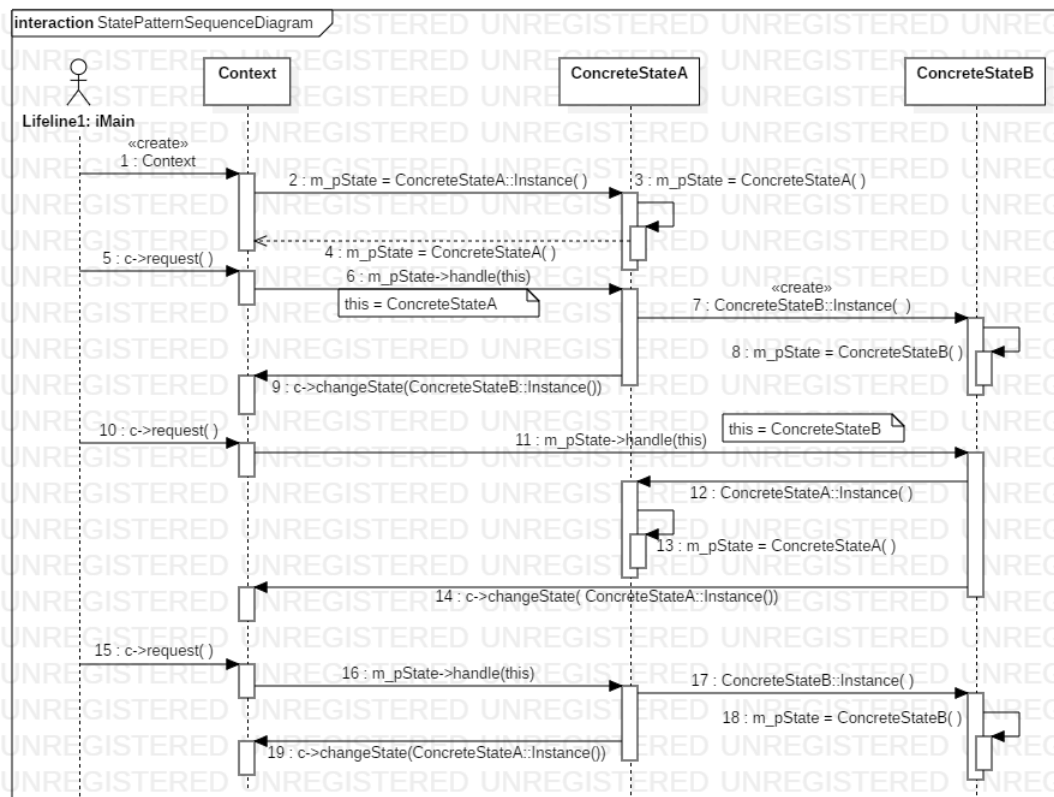
# 模式结构

状态模式包含如下角色：

- Context：环境类
- State：抽象状态类
- ConcreteState：具体状态类



# 时序图



# 关键代码

```
int main()
{
    char a = '0';
    if ('0' == a)
    {
        cout << "yes" << endl;
    }
    else
    {
        cout << "no" << endl;
    }

    Context* c = new Context();
    c->request();
    c->request();
    c->request();

    system("pause");
    return 0;
}
```

```

Context::Context()
{
    //default is a
    m_pState = ConcreteStateA::Instance();
}

void Context::request()
{
    m_pState->handle(this);
}

void ConcreteStateA::handle(Context * c)
{
    cout << "doing something in State A. \n done,change state to B" << endl;
    c->changeState(ConcreteStateB::Instance());
}

State * ConcreteStateB::Instance()
{
    if (NULL == m_pState)
    {
        m_pState = new ConcreteStateB();
    }
    return m_pState;
}

void Context::changeState(State * st)
{
    m_pState = st;
}

```

## 测试结果

```

int main()
{
    char a = '0';
    if ('0' == a)
    {
        cout << "yes" << endl;
    }
    else
    {
        cout << "no" << endl;
    }
    Context* c = new Context();
    c->request();
    c->request();
    c->request();

    system("pause");
    return 0;
}

```

```

D:\Design Pattern1\DesignPatterDer
yes
doing something in State A.
done,change state to B
doing something in State B.
done,change state to A
doing something in State A.
done,change state to B
请按任意键继续. . .

```