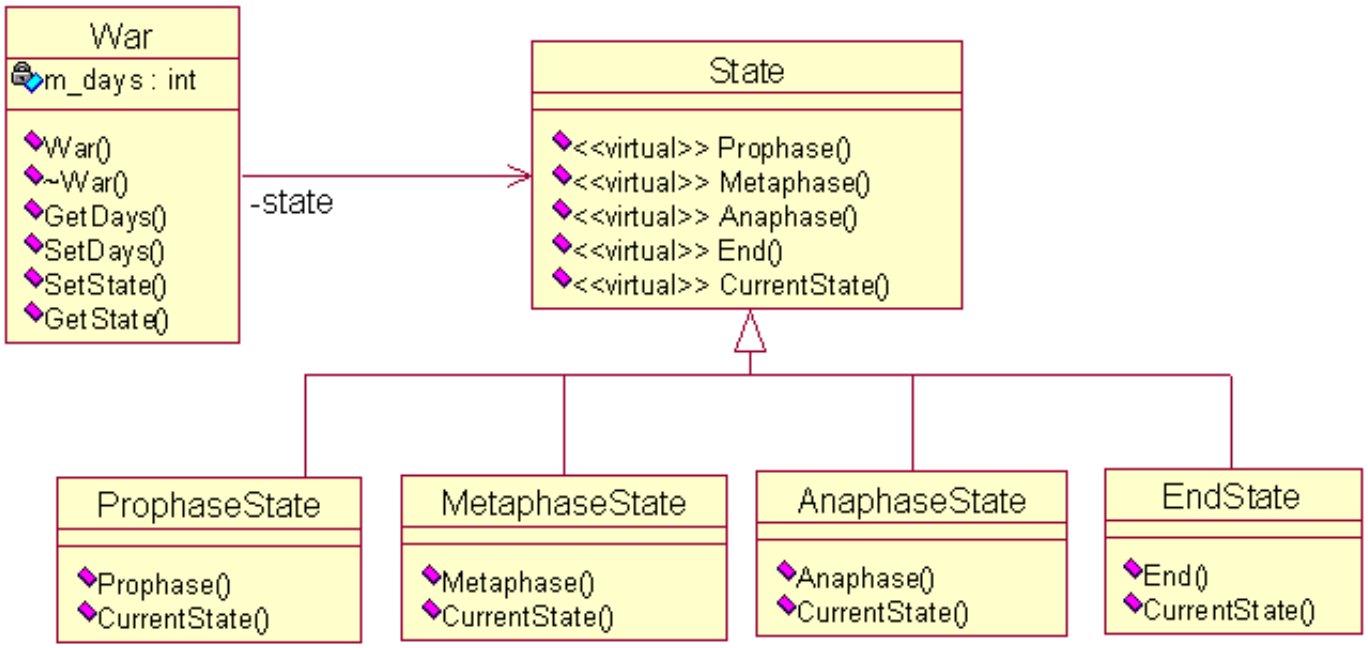
設計模式C++實現（16）——狀態模式

星期六, 2013 12月 14, 1:14 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。本文介紹狀態模式的實現。

狀態模式：允許一個對象在其內部狀態改變時改變它的行為。對象看起來似乎修改了它的類。它有兩種使用情況：（1）一個對象的行為取決於它的狀態，並且它必須在運行時刻根據狀態改變它的行為。（2）一個操作中含有龐大的多分支的條件語句，且這些分支依賴於該對象的狀態。本文的例子為第一種情況，以戰爭為例，假設一場戰爭需經歷四個階段：前期、中期、後期、結束。當戰爭處於不同的階段，戰爭的行為是不一樣的，也就說戰爭的行為取決於所處的階段，而且隨著時間的推進是動態變化的。下面給出相應的UML圖。



實現的代碼比較簡單，給出War類和State類，War類中含State對象（指針形式）。

```
1.  class War;
2.  class State
3.  {
4.  public:
5.      virtual void Prophase() {}
6.      virtual void Metaphase() {}
7.      virtual void Anaphase() {}
8.      virtual void End() {}
9.      virtual void CurrentState(War *war) {}
10. };
11. //戰爭
12. class War
13. {
14. private:
15.     State *m_state;  //目前狀態
```

```
    int m_days;      //戰爭持續時間
public:
    War(State *state): m_state(state), m_days(0) {}
    ~War() { delete m_state; }
    int GetDays() { return m_days; }
    void SetDays(int days) { m_days = days; }
    void SetState(State *state) { delete m_state; m_state = state; }
    void GetState() { m_state->CurrentState(this); }
};
```

給出具體的狀態類：

```
//戰爭結束
class EndState: public State
{
public:
    void End(War *war) //結束階段的具體行為
    {
        cout<<"戰爭結束"<<endl;
    }
    void CurrentState(War *war) { End(war); }
};
//後期
class AnaphaseState: public State
{
public:
    void Anaphase(War *war) //後期的具體行為
    {
        if(war->GetDays() < 30)
            cout<<"第"<<war->GetDays()<<"天：戰爭後期，雙方拚死一搏"<<endl;
        else
        {
            war->SetState(new EndState());
            war->GetState();
        }
    }
    void CurrentState(War *war) { Anaphase(war); }
};
//中期
class MetaphaseState: public State
{
public:
    void Metaphase(War *war) //中期的具體行為
    {
        if(war->GetDays() < 20)
            cout<<"第"<<war->GetDays()<<"天：戰爭中期，進入相持階段，雙發各有損耗"<<endl;
        else
        {
            war->SetState(new AnaphaseState());
            war->GetState();
        }
    }
    void CurrentState(War *war) { Metaphase(war); }
```

```
};
//前期
class ProphaseState: public State
{
public:
    void Prophase(War *war)  //前期的具體行為
    {
        if(war->GetDays() < 10)
            cout<<"第"<<war->GetDays()<<"天：戰爭初期，雙方你來我往，互相試探對方"<<endl;
        else
        {
            war->SetState(new MetaphaseState());
            war->GetState();
        }
    }
    void CurrentState(War *war) { Prophase(war); }
};
```

使用方式：

```
//測試案例
int main()
{
    War *war = new War(new ProphaseState());
    for(int i = 1; i < 40;i += 5)
    {
        war->SetDays(i);
        war->GetState();
    }
    delete war;
    return 0;
}
```