AlexOne

三、策略模式

```cpp
#include <iostream>
using namespace std;
// 重点: 转型、多态
// Define a family of algorithms,encapsulate each one,and make them interchangeable.

class Weapon{                               // 将刀，枪都抽象成一个武器类
public:
    virtual void use()=0;
};

class Knife:public Weapon{                  // 覆写虚函数use
public:
    void use() { cout<<"Use knife"<<endl; }
};

class Gun:public Weapon{                    // 覆写虚函数use
public:
    void use() { cout<<"Use gun"<<endl; }
};

class CCsprite{
public:
    CCsprite(Weapon *w){ _w=w; }            // 改变接口指针的指向
    void changWeapon(Weapon *w){ _w=w; }    // 达到切换武器的效果
    void faighting(){ _w->use(); }
private:                                    // 保护和私有都可以
    Weapon * _w;                            // 将StrategyInterface对象指针私有化
};

int main(){
    Knife k;                                //将武器作为参数传进去
    CCsprite character(&k);
    character.faighting();

    Gun g;                                  //切换枪战斗
    character.changWeapon(&g);
    character.faighting();

    character.changWeapon(&k);              //换回刀战斗
    character.faighting();

    return 0;
}
```