

友元 (Friend)

郭 炜 刘家瑛





友元

- 友元函数
- 友元类





友元函数

- 一个类的友元函数可以访问该类的私有成员

```
class CCar; //提前声明 CCar类, 以便后面CDriver类使用
```

```
class CDriver {
```

```
    public:
```

```
        void ModifyCar( CCar * pCar) ; //改装汽车
```

```
};
```

```
class CCar {
```

```
    private:
```

```
        int price;
```

```
    friend int MostExpensiveCar( CCar cars[], int total); //声明友元
```

```
    friend void CDriver::ModifyCar(CCar * pCar); //声明友元
```

```
};
```



```
void CDriver::ModifyCar( CCar * pCar)
{
    pCar->price += 1000; //汽车改装后价值增加
}
int MostExpensiveCar( CCar cars[], int total) //求最贵汽车的价格
{
    int tmpMax = -1;
    for( int i = 0; i < total; ++i )
        if( cars[i].price > tmpMax)
            tmpMax = cars[i].price;
    return tmpMax;
}
int main()
{
    return 0;
}
```



- 将一个类的成员函数(包括构造, 析构函数)
→ 另一个类的友元

```
class B {  
    public:  
        void function();  
};
```

```
class A {  
    friend void B::function();  
};
```



友元类

- ▲ A是B的友元类 → A的成员函数可以访问B的私有成员

```
class CCar {  
    private:  
        int price;  
        friend class CDriver; //声明CDriver为友元类  
};  
class CDriver {  
    public:  
        CCar myCar;  
        void ModifyCar() { //改装汽车  
            myCar.price += 1000; // CDriver是CCar的友元类→可以访问其私有成员  
        }  
};  
int main() { return 0; }
```

Note:

友元类之间的关系

不能传递, 不能继承