

类成员的可访问范围

郭 炜 刘家瑛

北京大学



类成员的可访问范围

- 关键字 -- 类成员可被访问的范围
 - **private:** 指定私有成员, 只能在成员函数内被访问
 - **public :** 指定公有成员, 可以在任何地方被访问
 - **protected:** 指定保护成员
- 三种关键字出现的次数和先后次序都没有限制



对象成员的访问权限

定义一个类

```
class className {
```

private:

私有属性和函数

public:

公有属性和函数

protected:

保护属性和函数

```
};
```

说明类成员的
可见性



对象成员的访问权限

- 缺省为私有成员

```
class Man {  
    int nAge;           //私有成员  
    char szName[20]; // 私有成员  
  
public:  
    void SetName(char * Name){  
        strcpy(szName, Name);  
    }  
};
```



对象成员的访问权限

- ▀ 类的**成员函数内部**, 可以访问:
 - 当前对象的全部属性, 函数
 - 同类其它对象的全部属性, 函数
- ▀ 类的**成员函数以外**的地方,
 - 只能够访问该类对象的**公有成员**



```
class CEmployee {  
    private:  
        char szName[30]; //名字  
    public :  
        int salary;      //工资  
        void setName(char * name);  
        void getName(char * name);  
        void averageSalary(CEmployee e1,CEmployee e2);  
};  
void CEmployee::setName( char * name) {  
    strcpy( szName, name); //ok  
}  
void CEmployee::getName( char * name) {  
    strcpy( name, szName); //ok  
}
```



```
void CEmployee::averageSalary(CEmployee e1,CEmployee e2){
    salary = (e1.salary + e2.salary )/2;
}
int main(){
    CEmployee e;
    strcpy(e.szName,"Tom1234567889"); //编译错, 不能访问私有成员
    e.setName( "Tom"); // ok
    e.salary = 5000;    //ok
    return 0;
}
```

- 设置私有成员的目的
 - 强制对成员变量的访问一定要通过成员函数进行
- 设置私有成员的机制 -- 隐藏



szName → char szName[5]

- 如果szName不是私有, 需要修改全部:

```
strcpy(man1.szName, "Tom1234567889");
```

- 如果将szName变为私有,

所有对 szName的访问都是通过成员函数来进行,

```
man1.setName( "Tom12345678909887");
```