

类与对象(1)

类的定义

```
class className
{
    //类体：由成员函数和成员变量组成
};
//后面加分号
class为定义类的关键字，className为类的名字，{}中为类的主体
```

类中的元素称为类的成员；

类中的数据称为类的属性或者成员变量；

类中的函数称为类的方法或者成员函数；

定义方式

1. 声明和方法全部在类体中

```
class student
{
    void show()
        //显示信息
    {
        cout<<name<<"-"<<id<<endl;
    }
    char name[10]; //姓名
    int id; //学号
};
```

2. 声明放在.h文件中，定义放在.cpp文件中

```
.h文件——声明
class student
{
    void show(); //显示信息
    char name[10]; //姓名
    int id; //学号
};

.cpp文件——定义
#include "student.h"
void student::show()
{
    cout<<name<<"-"<<id<<endl;
}
```

访问限定符和封装

访问限定符

访问限定符：public（共有）、protected（保护）、private（私有）

- public修饰的成员在类外可以直接被访问；
- protected和private修饰的成员在类外不能直接被访问；
- 访问限定符作用域从该访问限定符开始的位置直到下一个访问限定符出现时位置；
- 访问限定符只在编译的时候有用，当数据映射到内存后，没有任何访问限定符上的区别；

1.C++中class和struct的区别？

- C++兼容C语言，所以C++中struct可以当成结构体去使用。另外C++中struct和class都可以用来定义类，区别是struct的成员默认访问方式是public，class的成员默认访问方式是private。

封装

面向对象三大特性：封装、继承、多态

封装：封装本质是一种管理，将数据和方法放到一起定义，把不想让别人看到的用protect/private封装起来，想给你看到的用public定义。

类的作用域

类定义了一个新的作用域，类的所有成员都在类的作用域中。在类体外定义成员，需要使用 "::" (作用域解析符)指明成员属于哪个类域。

```
class student
{
public:
    void show(); //显示信息
private:
    char name[10]; //姓名
    int id; //学号
};
//这里需要指定成员属于哪个类域
void student::show()
{
    cout<<name<<"-"<<id<<endl;
}
```

类的实例化

用类类型创建对象的过程，称为类的实例化。

类实例化出对象，就相当于定义出了类的成员变量。

```
class student
{
public:
    void show()
        //显示信息
    {
        cout<<name<<"-"<<id<<endl;
    }
private:
    char name[10]; //姓名
}
```

```
    int id;//学号
};
int main()
{
    student s1;
    //这就是类的实例化
    return 0;
}
```

类对象模型

计算类对象的大小

1. 类中既有成员变量，又有成员函数，但是在计算大小时，只考虑成员变量。(多个成员时，需要内存对齐)
2. 没有成员变量的类大小是1(开1个字节不是为了存储数据，而是为了占位表示类存在);

this指针

C++编译器给每个非静态的成员函数增加了一个隐藏的指针参数，让该指针指向当前对象(函数运行时调用该函数的对象)，在函数体中所有成员变量的操作，都是通过该指针去访问。这个指针就是this指针。

特性

1. this指针的类型：类类型* const
2. 只能在成员函数的内部使用
3. this指针本质上其实是一个成员函数的形参，是对象调用成员函数时，将对象地址作为实参传递给this形参，所以对象中不存储this指针
4. this指针是成员函数第一个隐含的指针形参，一般情况由编译器通过寄存器自动传递，不需要用户传递

this指针存在哪里？

- 因为this指针是隐含的形参，所以存在栈(栈帧)里面