

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

类声明和成员函数定义的分离

类声明和成员函数定义的分离

- “类的公用接口与私有实现的分离”形成了信息隐蔽。
- “将接口与实现分离”是软件工程的一个最基本的原则。
- 如何实施？
- 参照：函数的声明和实现的分离
- 进一步：程序的多文件组织

```
class Student
{
public:
    void set_data(int n, char *p);
    void display( );
private:
    int num;
    char name[20];
};
void Student::set_data(int n, char *p)
{... ..}
void Student::display( )
{... ..}
```

```
#include<iostream>
using namespace std;
int max(int,int);
int main( )
{
    int m,a,b;
    cin>>a>>b;
    m=max(a,b);
    cout<<m<<endl;
    return 0;
}
int max(int x,int y)
{
    return (x>y)?x:y;
}
```

一个程序，多个源/头文件

主模块 main.cpp

```
#include <iostream>
#include "student.h"
void main( )
{
    ...
}
```

成员函数定义文件 student.cpp

```
#include <iostream>
#include "student.h"
void Student: :display( )
{
    ...
}
```

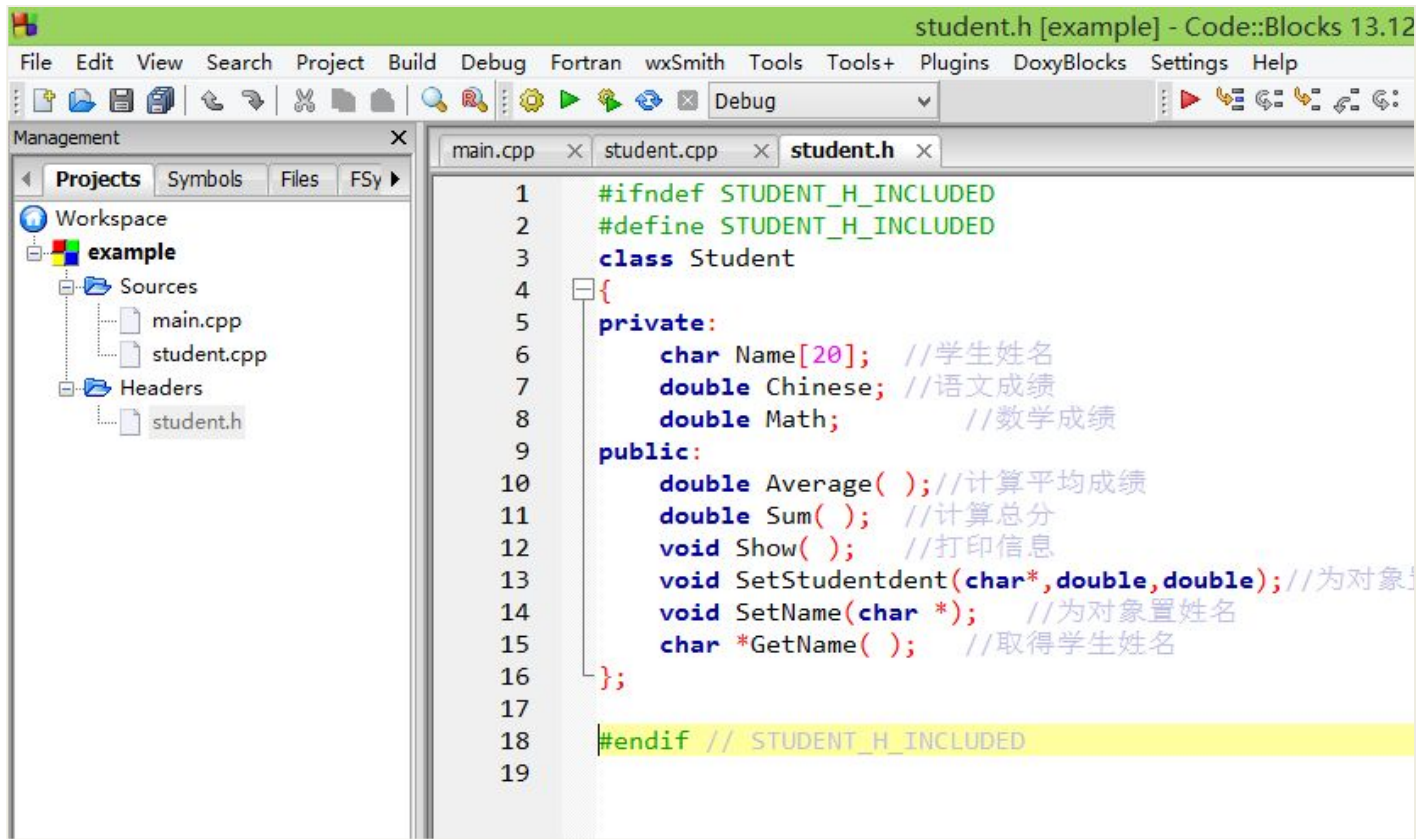
main.obj

或 .o 文件

student.obj

main.exe

示范：在Code::Blocks中的实施



类声明和成员函数定义的分离

❏ 将类的声明放在指定的头文件中

❏ 做法：用户如果想用该类，只要把有关的头文件包含进来即可，不必在程序中重复书写类的声明（重复声明还会带来其他问题）。

❏ 好处：减少工作量，节省篇幅，提高编程的效率。

❏ 程序中可以用该类来定义对象

❏ 在头文件中包含了类的声明，在程序中就可以用该类来定义对象，就可以调用这些对象的公用成员。

❏ 成员函数的定义另外放在一个文件中，与声明类的头文件分开

❏ 类成员函数的定义一般不放在头文件中，而放在另一个文件中

❏ 内置函数的定义要写在头文件中

❏ 这种方式实现了信息隐蔽

用好头文件

❏ 正确使用头文件带来的好处

- ❏ 支持代码的局部单独编译，尤其对类及成员函数声明
- ❏ 保证所有文件中使用的实体，源自同一声明
- ❏ 当声明需要修改时，只需要更新头文件

❏ 使用头文件的原则

- ❏ 头文件用于声明，而不是定义
- ❏ 头文件一般含有支持分别编译所需的类定义及变量和函数的声明
- ❏ 一些const对象要定义在头文件中

❏ 一个例外

- ❏ 内置(inline)成员函数需要定义在头文件中

一些小细节

```
#ifndef STUDENT_H_INCLUDED
#define STUDENT_H_INCLUDED
class Student
{
    .....
}
#endif // STUDENT_H_INCLUDED
```

student.h

```
#include "student.h"
.....
```

school.h

```
#include "student.h"
#include "school.h"
.....
```

school.cpp

类库

❏ 单独编译带来的好处

- ❏ 不必每次都重复对所有代码进行编译，而只需编译一次即可。
把编译后所形成的目标文件保存起来，以后在需要时把它调出来直接与程序的目标文件相连接即可。
- ❏ 在实际工作中，并不是将一个类声明做成一个头文件，而是将若干个常用的功能相近的类声明集中在一起，形成**类库**。

❏ 两种C++类库

- ❏ C++编译系统提供的**标准类库**；
- ❏ 用户根据自己的需要做成的**用户自定义类库**，提供给自己和自己授权的人使用。



类库的组成

❏ 类库包括两个组成部分

(1) 类声明头文件——公用接口；

(2) 已经过编译的成员函数的定义，它是目标文件——私有实现。

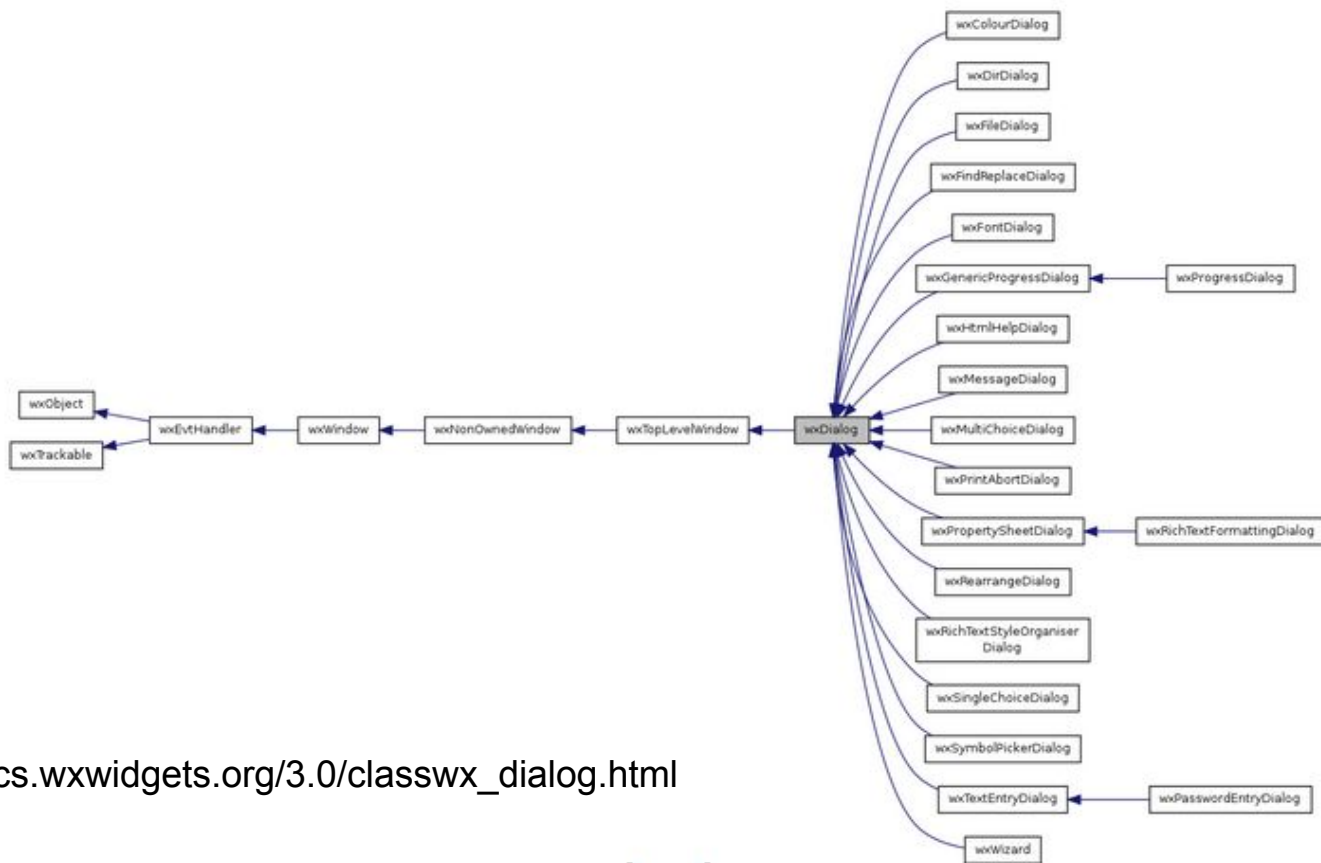
❏ 在用户程序中包含类声明头文件（用户使用类的公用接口），用户无需知道类的实现方法。

❏ 通过阅读头文件中成员函数的函数原型声明，也就清楚了类的使用方法。

C++类库



例：wxwidgets类库中的一部分



http://docs.wxwidgets.org/3.0/classwx_dialog.html

类库的工程意义

- 由于接口与实现分离，为软件开发商向用户提供类库创造了很好的条件。
 - 📁 开发商把用户所需的各种类的声明按类放在不同的头文件中，同时对包含成员函数定义的源文件进行编译，得到成员函数定义的目标代码。
 - 📁 开发商向用户（应用开发者）提供这些头文件和类的实现的目标代码(不提供函数定义的源代码)。
 - 📁 用户在使用类库中的类时，只需将有关头文件包含到自己的程序中，并且连接成员函数定义的目标代码即可。
- 由于类库的出现，用户可以像使用零件一样方便地使用在实践中积累的通用的或专用的类，大大减少了程序设计的工作量，有效地提高了工作效率。



wxWidgets
Cross-Platform GUI Library



THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>