

C++面向对象程序设计

第三讲：封装

抽象：获取实体的属性和行为

对具体实体（对象）进行概括，抽取一类对象的公共属性和行为。

- 注意本质，围绕重点，抓住共性 ——》属性和行为
- 属性：数据抽象，某类对象的属性或状态（对象相互区别的依据）
 - 标识属性
 - 确定属性的值域
- 行为：方法抽象，某类对象的行为特征或具有的功能
 - 标识行为
 - 行为的处理对象（有哪些？来源是什么？）
 - 行为的处理结果（有什么？谁接受/收处理结果？）

抽象的实例：时间

- 时间的构成要素（应用范围）
 - 时、分、秒、百分秒、毫秒、微秒、纳秒
- 时间的使用
 - 啥时候了？几点了？
 - 6点（上午还是下午？）
 - 30分钟后出发、2小时后去哪儿等等
 - 到底是啥时候呢？
 -

抽象的实例：时间属性

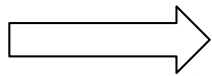
- 属性
 - 小时
 - 值域：0-12 / 0-23 ?
 - 分钟
 - 值域：0-59
 - 秒
 - 值域：0-59

抽象的实例：时间功能

- 设置当前时间
 - 设定时间对象的值(时间)，谁如何给定？
- 显示
 - 将时间对象的值打印在屏幕上
- 时/分/秒
 - 获取时间对象的分量值（时、分、秒）
- 计算
 - 加上时/分/秒后的时间值，是改变对象本身的值，还是生成新的对象
-
 - 还能想到什么？

封装时间对象——类

整合属性与行为并进行程序语言表示—〉对象泛化为类
对象泛称—〉类名；属性—〉成员变量；行为—〉成员函数

- 时间
 - 属性
 - 时 (0-23)
 - 分 (0-59)
 - 秒 (0-59)
 - 行为
 - 设定时间 (时、分、秒)
 - 显示时间 ()
 - 增加分量 (时/分/秒)
- 
- Time
 - 成员变量
 - char hour; 0-23
 - char minute; 0-59
 - char second; 0-59
 - 成员函数
 - void SetTime(char, char ,char)
 - void ShowTime ()
 - void Add(char, char)

类定义（封装）（续）

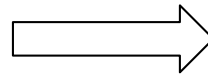
- **Time**

- **成员变量**

- **char hour; 0-23**
 - **char minute; 0-59**
 - **char second; 0-59**

- **成员函数**

- **void SetTime(char, char ,char)**
 - **void ShowTime ()**
 - **void Add(char, char)**



- **CTime**

- **成员变量**

- **int hour; 0-23**
 - **int minute; 0-59**
 - **int second; 0-59**

- **成员函数**

- **void SetTime(int, int, int)**
 - **void ShowTime ()**
 - **void add(int, char)**
 - **void addHour(int)**
 - **void addSec(int)**
 - **void addMin(int)**

封装：定义时间类(头文件)

```
#ifndef TIME_H_
#define TIME_H_
namespace nsname {
class CTime {
public:
    int hour;
    int minute;
    int second;
    void SetTime(int h, int m, int s);
    void ShowTime();
    void addHour(int);
    void addMin(int);
    void addSec(int);
};
} //namespace nsname
#endif /* TIME_H_ */
```

文件名： mytime

类的实现——时间类

```
#include "mytime"
#include <iostream>
#include <iomanip>
using namespace std;
namespace nsname {

void CTime::SetTime(int h, int m, int s)
{
    hour = h;
    minute = m;
    second = s;
}
void CTime::ShowTime()
{
    cout << hour << ":" << minute << second;
}
```

文件名: mytime.cpp

```
void CTime::addHour(int h)
{
    hour += h;
}
void CTime::addMin(int m)
{
    minute += m;
}
void CTime::addSec(int s)
{
    second += s;
}
}
```

类的使用——时间对象

```
#include "mytime"
```

```
using namespace nsname;
```

```
int main() {
```

```
    CTime c1;
```

```
    c1.SetTime(14,20,30);
```

```
    c1.ShowTime();
```

```
    c1.SetTime(8,2,30);
```

```
    c1.ShowTime();
```

```
    c1.SetTime(8,2,3);
```

```
    c1.ShowTime();
```

```
    c1.addHour(5);
```

```
    c1.ShowTime();
```

```
    return 0;
```

```
}
```

类定义（封装）

类的属性和行为的访问限制

- 限制数据成员和函数成员的访问权限
 - 公有 public
 - 完全公开的属性和行为
 - 私有 private
 - 个体专属的属性和行为
 - 保护 protected
 - 家族私有的属性和行为

类的定义

- 类是一种用户自定义的数据类型
- 声明形式
 - Class 类名称
 - {
 - public:
 - 公有成员
 - private:
 - 私有成员
 - protected:
 - 保护成员
 - } ;

注意给定访问限制的方式：

从限制方式开始的后续成员，
直到变更访问限制

默认的成员访问限制是“私有成员”

类的定义（续）

- 成员访问限制符号对其后续成员均有效，直到遇到下一个成员访问限制符号；
- 紧跟在类名称的后面的成员，没有访问限制符号注明的话，均为私有成员；
- 从程序的角度而言
 - 随时随地可以访问的是类的公有成员；
 - 类的私有成员则只能在类的成员函数中被访问；
 - 类的保护成员可以被类及其派生类的成员函数中被访问。

课堂作业：

- 定义一个日期类
 - 能想到什么，就做什么！！