

# 第7章 数据的抽象与封装

|              |               |            |             |
|--------------|---------------|------------|-------------|
| 1、实体、对象与类的概念 | 2、类的定义        | 3、对象声明与引用  | 4、私有、公有与保护  |
| 5、日期类的设计     | 6、两种程序设计思想    | 7、汽车类的设计   | 8、几何图形圆类的设计 |
| 9、构造函数的定义    | 10、重载构造函数     | 11、析构函数的定义 | 12、整数翻译函数   |
| 13、实际意义的析构函数 | 14、Person类的设计 | 15、对象与指针   | 16、this指针   |

# 类的定义格式

```
class <类名>
{
    private:                //私有成员
        <数据成员说明语句>;
        <函数成员说明语句>;
    public:                 //公有成员
        <数据成员说明语句>;
        <函数成员说明语句>;
    protected:           //保护成员
        <数据成员说明语句>;
        <函数成员说明语句>;
};
```

# 实体钟表如何抽象描述

---

👉 **特征：即数据成员？**

👉 时、分、秒

👉 型号、生产厂、出厂日期、价格等

👉 **功能：即函数成员？**

👉 设置时间

👉 钟表运转或运行

👉 报时响铃

👉 显示时间

👉 .....  
.....

---

# 定义钟表类

```
class Clock //定义钟表类
{
private: //数据成员一般为私有成员
    int Hour;           //小时属性
    int Minute;       //分钟属性
    int Second;       //秒属性
    float Price;      //价格属性
public: //函数成员一般为公有成员
    void Set(int h,int m,int s,float p); //设置时间操作
    void Run();           //钟表运转操作
    void Report_Time();  //报时响铃操作
    void Show_Time( );   //显示时间操作
};
```

# 成员函数两种定义方法

## 👉 类定义的语法格式

```
class <类名>
{
    <类体>
}
```

### 1. 在类体内直接定义函数成员，该成员函数又称内联函数

👉 在类体内定义内联函数等同一一般函数定义格式

### 2. 在类体外定义函数成员

👉 在类体外定义成员函数的一般形式如下：

```
<类型> <类名> :: <函数名> (<参数表>)
{
    <函数体>
}
```

# 内联函数定义方法

```
class Clock //定义钟表类
{
private: //数据成员一般为私有成员
    int Hour; //小时属性
    int Minute; //分钟属性
    int Second; //秒属性
    float Price; //价格属性
public: //函数成员一般为公有成员
    void Set(int h,int m,int s,float p); //设置时间操作
    void Run(); //钟表运转操作
    void Report_Time(); //报时响铃操作
    void Show_Time( ) { cout<<Hour<<':'<<Minute<<':'<<Second; };
};
```

# 类外定义成员函数方法

```
void Clock::Set(int h,int m,int s,float p)    //设置修改4个数据成员的值的函数
{
    Hour = h;
    Minute = m;
    Second = s;
    Price = p;
}
void Clock::Run()    //模拟运行钟表运行函数
{
    int i=0;
    for(i=0;i<10;i++)    //只模拟运行10秒钟
    {
        Second++;
        if(Second==60)
        {
            Second=0;
            Minute++;
            if(Minute==60)
            {
                Minute=0;
                Hour++;
                if(Hour==24) Hour=0;
            }
        }
        cout<<"\r";    //不换行，返回当前行的首位
        Sleep(1000);    //程序暂停运行秒
        Show_Time();
    }
}
```

# 类外定义成员函数方法

---

```
void Clock::Report_Time()  
{  
    Show_Time();  
    if ( Minute==0&&Second==0 )  
    {  
        for(int i=0;i<Hour;i++)  
        {  
            cout<<"\007";  
            Sleep(1000);  
        }  
    }  
}
```



---

上面介绍了钟表类的完整定义  
下一个微片段介绍对象声明与引用

---

感谢收看！