

面向对象方法与C++程序设计

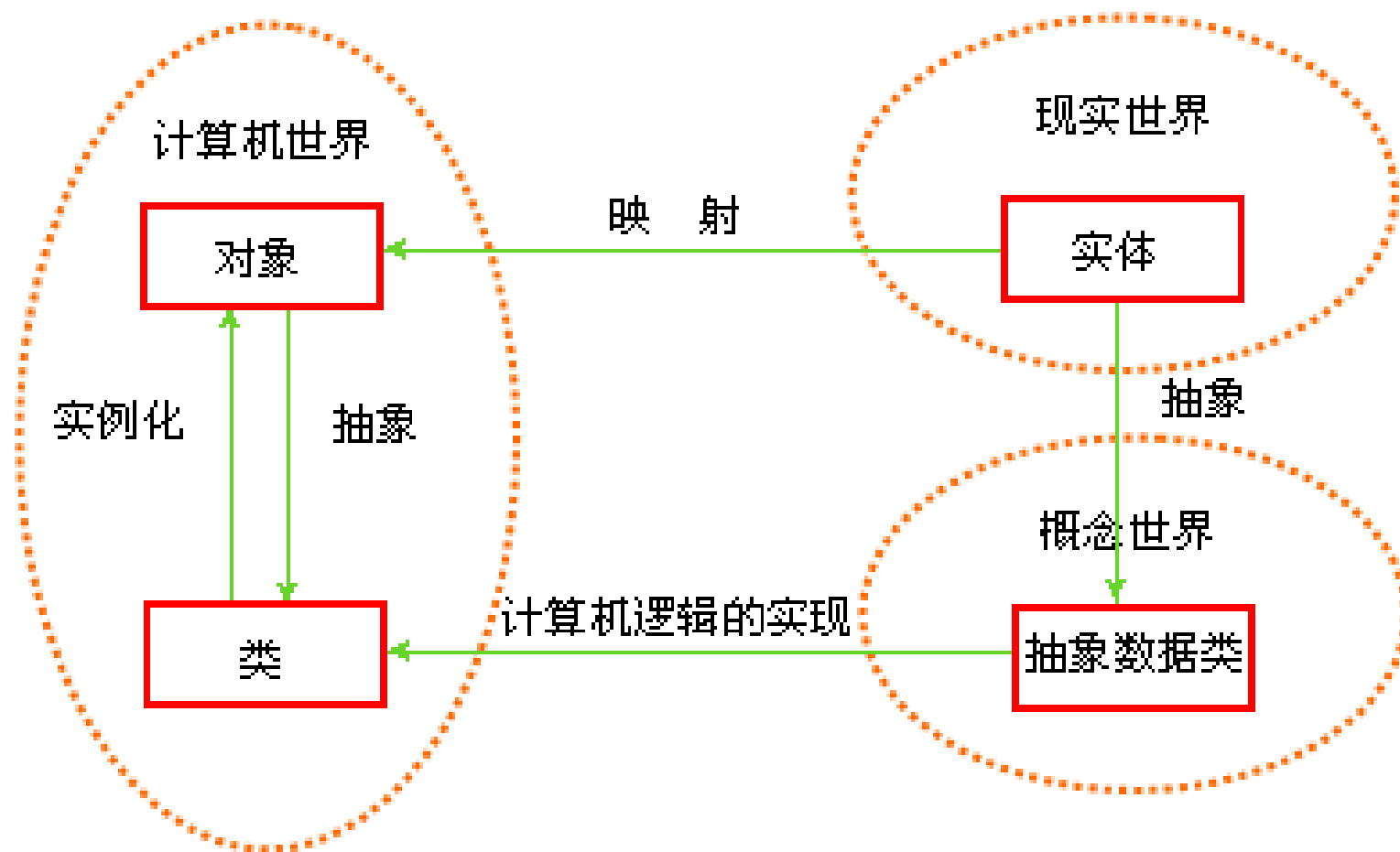
第2章

类与对象

大连理工大学
主讲人-赵小薇



理解类



理解类



类是具有相同或相似性质的对象的抽象。抽象是从众多的事物中抽取出共同的、本质性的特征，而舍弃其非本质的特征的过程。

类还起到封装的作用，封装有两个含义：

一是把对象的全部属性和行为结合在一起，形成一个不可分割的独立单位；

二是尽可能隐蔽对象的内部属性和实现细节，对外形成一道屏障，与外部的联系只能通过公共接口实现。



类的定义



class 类名

{

private:

私有的成员数据和成员函数声明;

public:

公有的成员数据和成员函数声明;

protected:

保护的成员数据和成员函数声明;

};



类的定义



```
class Clock
{
private :
    int hour, minute, second;           //关于时间的数据;
public :
    void setClock(int h, int m, int s); //调整时间值;
    void showClock ( );                 //显示时间值;
};
```



类的定义



```
class TimeBomb
```

```
{
```

```
private :
```

```
    Clock timer;
```

```
    int explosive;
```

```
    .....
```

```
public :
```

```
    void setTime();
```

```
    .....
```

```
};
```

//关于定时器的数据;
//炸药量

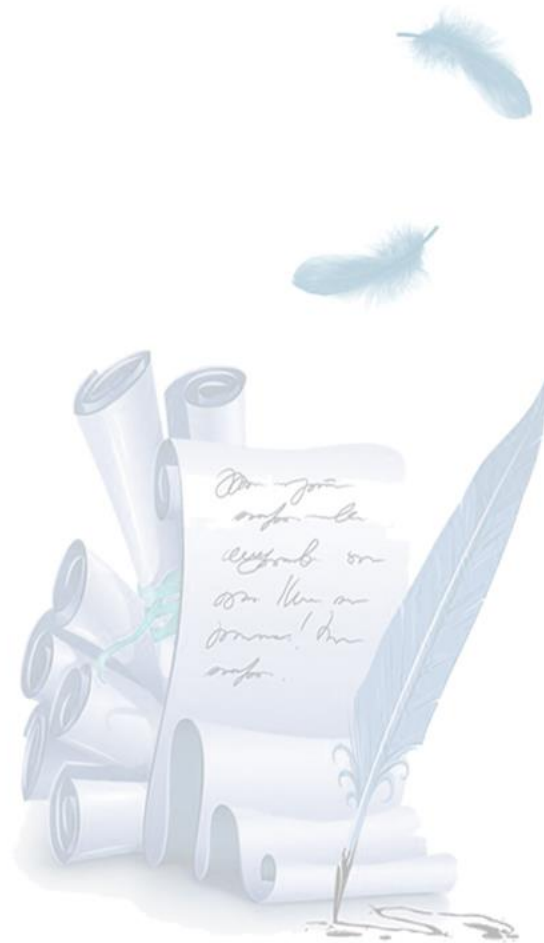
//设置爆炸时间;



类的定义



1. 类class是定义面向对象程序模块的数据类型，是“模板”
2. 类的定义是声明一个数据结构，而不是定义一个函数，定义的最后有分号结束。
3. 说明类成员访问权限的关键字private、protected和public可以按任意顺序、任意出现多次，一个成员只能有一种访问权限。
4. 不能在类内给数据成员赋初值，只有给对象数据成员赋初值。
5. 成员函数可以重载，Clock中可以定义多个set函数。



类的实现



```
class Clock
{
private :
    int hour, minute, second;           //关于时间的数据
public :
    void setClock (int h, int m, int s){
        hour= ( h >=0 && h <=23)? h : 0 ;
        minute= ( m >=0 && m <=59)? m : 0 ;
        second= ( s >=0 && s <=23)? s : 0 ;    }
    void showClock ( ){
        cout<<hour<<minute<<second;    }
};
```

类内
实现



类的实现



```
class Clock
{
private :
    int hour, minute, second;           //关于时间的数据;
public :
    void setClock (int h, int m, int s);
    void showClock ( );
};
void Clock::setClock (int h, int m, int s){
    hour= ( h >=0 && h <=23)? h : 0 ;
    minute= ( m >=0 && m <=59)? m : 0 ;
    second= ( s >=0 && s <=23)? s : 0 ; }
void Clock::showClock ( ){
    cout<<hour<<minute<<second; }
```

类外
实现

类名+
作用域
运算符



内联函数



1. 类内实现为内联函数
2. 类外实现加inline为内联函数，如：

```
inline void Clock::showClock ( ) {  
    cout<<hour<<minute<<second  
}
```



内联函数不调用，
而是编译时代码
嵌入主调函数

内联函数体中包含
复杂结构，声明为
内联函数，编译系
统也不会按照内联
函数处理

