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

回顾函数重载

- 什么是函数重载？
  - 若干个不相同的函数允许共用一个函数名称
- 下面3个排序函数共用一个函数名称
  - void sort(int a[],int n);//采用冒泡排序方法
  - void sort(float a[],int n);//采用快速排序方法
  - void sort(double a[],int n);//采用选择排序方法
- 调用运行时根据实际参数确定哪一个函数运行

重载构造函数

- 一个类中允许定义多个构造函数
- 将日期类再增加一个构造函数

```
Date()
{
    Year=1900; Month=1; Day=1;
}
```

- 在声明对象时自动选择执行某个构造函数

数据成员初始化的四种方法

1. **在构造函数的函数体中进行初始化**

```
Date(Date &d){ year=d.year;month=d.month;day=d.day;}
```

2. **在构造函数的头部初始化**

类名::构造函数名(参数表):变量1(初值1),…变量n(初值n){…}

例如: Date::Date():year(1900),month(1),day(1){}

3. **混合初始化：前两种方式结合**

例如： Date::Date(int y,int m,int d):year(y),month(m){ day=d; }

4. **使用默认参数初始化**

```
Date(char yy[], int mm = 1, int dd = 1)
{   year = atoi(yy);
    month = mm;
    day = dd;
}
```

日期类中4个重载构造函数

```
#include<iostream>
#include<string.h>
#include<stdlib.h>
using namespace std;
class Date
{           int year,month,day;
public:
            Date():year(1900),month(1),day(1) { }        //无参构造函数
            Date(int yy,int mm = 1,int dd = 1);          //默认参数构造函数
            Date(Date &d):year(d.year),month(d.month),day(d.day){};//日期对象参数构造函数
            Date(char *ps);                  //字符串日期构造函数
            void print_ymd();
};
```

默认参数构造函数定义

```
Date::Date(int yy,int mm,int dd):year(1900),month(1),day(1)
{
        if (yy>=1900&&yy<=9999)
                year = yy;
        else
        {
                return;
        }
        if (mm>=1&&mm<=12)
                month = mm;
        else
        {
                        year=1900;
                        return;
        }
        if (dd>=1&&dd<=31)
                day = dd;
        else
        {
                        year=1900;
                        month=1;
                        return;
        }
};
```

日期类中重载构造函数

```
Date::Date(char *ps):year(1900),month(1),day(1)
{               char py[5],pm[3],pd[3];
                strncpy(py,ps,4);
                ps=ps+5;
                strncpy(pm,ps,2);
                ps=ps+3;
                strncpy(pd,ps,2);
                int yy=atoi(py),mm=atoi(pm),dd=atoi(pd);
                if (yy>=1900&&yy<=9999)   year = yy;  else   return;
                if (mm>=1&&mm<=12)
                                month = mm;
                else
                {               year=1900;
                                return;
                }
                if (dd>=1&&dd<=31)
                                day = dd;
                else
                {               year=1900;
                                month=1;
                                return;
                }
}
```

日期类重载构造函数测试

```
int main()
{           Date date1;                 //使用无参构造函数
            cout << "date1:";
            date1.print_ymd();
            Date date2(2006);           //使用哪个构造函数?
            cout << "date2:";
            date2.print_ymd();
            Date date3(2006,4);         //使用哪个构造函数?
            cout << "date3:";
            date3.print_ymd();
            Date date4(2006,4,8);       //使用哪个构造函数?
            cout << "date4:";
            date4.print_ymd();
            Date date5(2006,14,8);      //使用哪个构造函数?
            cout << "date5:";
            date5.print_ymd();
            Date date6(date4);          //使用哪个构造函数?
            cout << "date6:";
            date6.print_ymd();
            Date date7("2008-08-08");   //使用哪个构造函数?
            cout << "date7:";
            date7.print_ymd();
            return 0;
}
```

感谢收看！