

# 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

# 能翻译整数的机器人

---

## □ 机器人的特征

- 姓名

- 型号

- .....

## □ 机器人的功能

- 翻译整数函数：

  - 例如128翻译成one hundred and twenty-eight

- 构造函数

- 设置函数

- 析构函数

- .....

---

# 机器人人类的定义

```
class robot
{
private:
    char name[20];    //机器人姓名
    char type[20];    //机器人型号
public:
    robot()//构造函数
    {
        strcpy(name,"XXXXXXX");
        strcpy(type,"XXXXXXX");
    }
    void set(char n[],char t[])    //设置修改数据
    {
        strcpy(name,n);
        strcpy(type,t);
    }
    void out(int a);//英文中每三位数读法相同，所以定义out函数翻译小于1000的整数
    void tran_int(int n);//将1至1999999999的整数翻译成英文句子
    ~robot(){}; //析构函数
};
```

# 特殊数字词汇定义

//定义两个全局字符指针数组，存取所需的单词

//**num1**中为**1**到**19**，空出了**0**，所以可以直接用**num1 [n]**调用，得到**n**对应单词

```
static char *num1[]=
```

```
{
```

```
    "","one ","two ","three ","four ","five ","six ","seven ","eight ",
```

```
    "nine ","ten ","eleven ","twelve ", "thirteen ","fourteen ",
```

```
    "fifteen ","sixteen ","seventeen ","eighteen ","nineteen "
```

```
};
```

//**num10**中为**20-90**，空出了**0**和**1**，所以可以直接用**num10[n/10]**调用，得到**n**对应单词

```
static char *num10[]=
```

```
{
```

```
    "","","twenty ","thirty ","forty ","fifty ","sixty ","seventy ",
```

```
    "eighty ","ninety "
```

```
};
```

# 小于1000整数翻译函数

```
void robot::out(int a)
{   int b=a%100;
    //若百位不为零，输出百位数加hundred，若此时十位个位均为0，不加and
    if(a/100!=0)
    {       cout<<num1[a/100]<<"hundred ";
            if(b!=0)
                cout<<"and ";
        }
    //当后两位在20以内时，直接调用num1[n]，输出
    if(b<20)
        cout<<num1[b];
    //当后两位大于20时
    else
    {   //先调用num10，输出十位数
        cout<<num10[b/10];
        //个位不为0时应输出"-"个位数
        if(b%10!=0)
            cout<<"\b-"<<num1[b%10];
    }
}
```

# 整数翻译函数

```
void robot::tran_int(int n)
{
    if(n>1999999999)
        cout<<"dev C++平台无法处理大于1999999999位的数！"<<endl;
    else
    {
        //三位三位取出，存入abcd中
        int a=n/1000000000,b=(n%1000000000)/1000000,c=(n%1000000)/1000,d=n%1000;
        //不等于0时，输出并加上million或thousand
        if(a!=0)
        {
            out(a);
            cout<<"billion ";
        }

        if(b!=0)
        {
            out(b);
            cout<<"million ";
        }
        if(c!=0)
        {
            out(c);
            cout<<"thousand ";
        }
        if(d!=0)
        {
            //据英文语法规则，最后两位前一定有and
            if(d<100&&(a!=0||b!=0||c!=0))
                cout<<"and ";
            out(d);
        }
    }
    cout<<endl;
}
```

# 机器人翻译测试

```
int main()  
{  
    int n;  
    cout<<"请输入n: ";  
    cin>>n;  
    cout<<n<<endl;  
    robot brown;  
    brown.tran_int(n);  
  
    return 0;  
}
```

---

**感谢收看！**