

# 程序设计与算法语言II

## 第九章 流类库和输入/输出

曹鹏

Email: [caopeng@seu.edu.cn](mailto:caopeng@seu.edu.cn)

Tel: 13851945861

# 本章提纲

---

- ◆ C++的基本流类体系 (9.1节)
- ◆ 输入/输出 (9.3.3节)
- ◆ 文本文件的操作 (9.4.1,9.4.2节)
- ◆ 二进制文件的操作 (9.4.3节)
- ◆ 文件与对象 (9.6节)

# 提取/插入运算符

如何从键盘输入变量?

◆提取运算符>>

◆标准输入流对象cin

```
cin>>i;  
cin>>ch;
```

如何将变量输出到显示器?

◆插入运算符<<

◆标准输出流对象cout

```
cout<<i;  
cout<<ch;
```

◆如何从键盘输入一个类对象?

◆如何向显示器输出一个类对象?

➔重载提取运算符(>>)和插入运算符(<<)

# 提取/插入运算符的重载

- ◆ 系统已对内置数据类型实现了重载，但不支持非基本数据类型，例如程序员定义的类
- ◆ 输入类对象：只能逐个输入类对象的各个成员变量
- ◆ 输出类对象：只能逐个输出成员变量

//T实例化为int类型

```
template <typename T>
class A{
    T t;
public:
    void set(){ //输入
        cin >> t;
    }
    void print(){ //输出
        cout << t;
    }
}
```

//T实例化为float类型

```
template <typename T>
class A{
    T t;
public:
    void set(){
        cin >> t;
    }
    void print(){
        cout << t;
    }
}
```

//T实例化为Complex类

```
template <typename T>
class A{
    T t;
public:
    void set(){
        cin >> t.real;
        cin >> t.image;
    }
    void print(){
        cout << t.real;
        cout << t.image;
    }
}
```

仅实例化为基本数据类型  
可以实现统一的类模板

一旦需要支持实例化为自定义类  
无法实现统一的类模板

# 提取/插入重载运算符的声明

解决措施：在自定义类重载提取/插入运算符重载函数

重载方式：友元函数

① friend ② istream& ③ operator>> ④ (istream&, ⑤ className&);  
friend ostream& operator<< (ostream& className&);

- ① 重载方式：是cin/cout调用，而cin/cout是文本流类对象，不能修改文本流类，只能定义为className类的友元函数
- ② 返回值：对应形如cin>>a>>b；在执行cin>>a后的返回值应该仍然是文本流对象，才能进一步执行cin>>b，因此返回值类型同参数1
- ③ 函数名：重载提取运算符(>>)和插入运算符(<<)
- ④ 参数1：对应运算符左边的cin/cout，处理的类对象本身，因此引用调用
- ⑤ 参数2：对应运算符右边的程序员自定义类对象，处理的也是类对象本身，因此也是引用调用

# 提取/插入重载运算符的定义

```
friend istream& operator>>( istream&, className&)  
{  
    //逐个输入className类对象的每个成员变量  
}
```

```
friend ostream& operator<<(ostream&, className&)  
{  
    //逐个输出className类对象的每个成员变量  
}
```

# 复数类提取/插入重载运算符的定义

```
class Complex
{
    double Real, Image;
public:
    Complex(double r=0.0, double i=0.0):Real(r),Image(i){}
    friend ostream& operator<<(ostream &s, const
Complex &z);
    friend istream& operator>>(istream &s, Complex &a);
};
```

友元函数定义在需要支持<<和>>的  
自定义类Complex中



# 复数类提取/插入重载运算符的定义

```
ostream& operator<<(ostream &s, const Complex &z)
```

```
//输出格式re, im
```

```
{  
    s<<z.Real<<','<<z.Image;  
    return s;  
}
```

◆ 输出格式为(re, im)  
如何实现?

```
istream& operator>>(istream &s, Complex &a)
```

```
//输入格式re, im
```

```
{  
    char c=0;  
    s>>a.Real>>c; //实部  
    if(c== ',') s>>a.Image; //虚部  
    return s;  
}
```

◆ 输入格式为

- ◆ (re, im)
- ◆ (re)
- ◆ (re

时如何处理?



# 复数类提取/插入重载运算符的应用

输入：1.0,2.0

输出：1.0,2.0

```
//不通过复数类提取操作符>>  
double real, image;  
char c;  
cin>>real>>c>>image;
```

```
//定义了复数类提取操作符>>  
complex c1;  
cin>>c1;
```

```
//不通过复数类插入操作符<<  
double real=1, image=2;  
cout<<real<<' ' <<image;
```

```
//定义了复数类插入操作符<<  
complex c1(1.0,2.0);  
cout<<c1;
```

# 本章小结(1)

- ◆基本流类体系 (9.1节)
  - ◆输入/输出设备→文件
    - ◆从输入设备取得数据：提取(>>)
    - ◆向输出设备添加数据：插入(<<)
  - ◆标准输入设备：键盘；标准输出设备：显示器
- ◆文件操作 (9.3.3节, 9.6节)
  - ◆文本/二进制文件的区别
    - ◆文本文件，最小信息单位：字符
    - ◆二进制文件，最小信息单位：字节
  - ◆操作过程
    - ◆定义文件流对象：构造函数
    - ◆打开磁盘文件：open函数
      - ◆返回值：void
      - ◆参数：第2个参数是打开方式，区分文本/二进制方式
    - ◆读写文件
    - ◆关闭文件流对象：close函数
      - ◆返回值：void

## 本章小结(2)

### ◆读写文本文件 (9.4.1, 9.4.2节)

#### ◆重载提取/插入操作符函数

#### ◆定义ClassName类的友元函数

```
friend istream& operator>>( istream&, className&);  
friend ostream& operator<<(ostream&, className&);
```

### ◆读写二进制文件 (9.4.3节)

#### ◆调用read/write函数

#### ◆调用istream/ostream类的成员函数

#### ◆可在ClassName类的成员函数中调用

```
istream& istream::read(char *, int);  
ostream& ostream::write(const char *,int);
```

---



# End