

# 计算机科学基础II

## 实验十 类与对象的基本概念

曹鹏

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

Tel: 13851945861

# 实验目的

## 实验十

类是C++扩展数据类型，可以封装不同类型的数据成员和函数成员，类是面向对象程序设计的基础。本次实验内容包括面向对象的基本概念、构造函数与析构函数，从实际问题抽象出类等，通过实验要求：

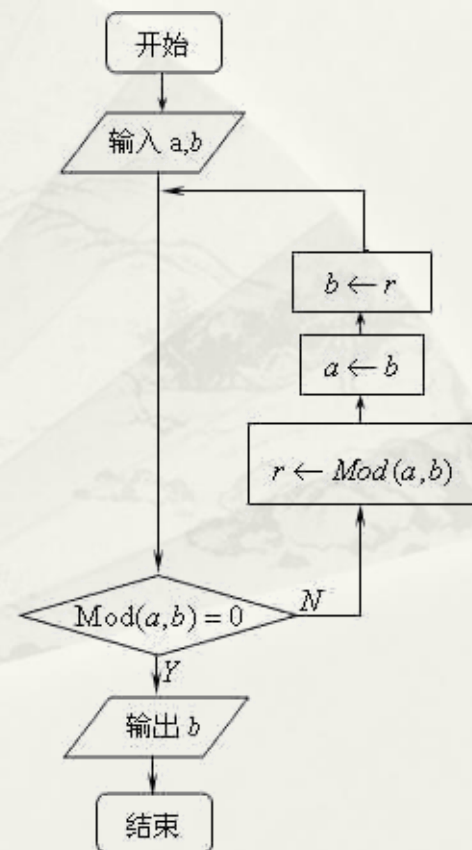
1. 掌握面向对象的基本概念，类的定义方法。
2. 掌握类成员的访问权限及访问类成员的方法。
3. 掌握内联函数和默认函数。
4. 掌握构造函数与析构函数的意义及使用方法。

编程：建立一个分数类。分数类的数据成员包括分子和分母，操作包括约分、通分、比较、加、减、乘、除、求倒数、显示和输入。分数类定义如下：

```
class fraction{  
    int above;      //分子  
    int below;      //分母  
    void reduction(); //约分 (欧几里得法)  
    fraction makeCommond(fraction); //通分  
    //f2=f1.makeCommond(f2);  
    //f1=1/2, f2=2/3,通分后f1=3/6, f2=4/6  
};
```

欧几里得法  
求最大公约数

Iter.	0	1	2
r		12	6
a	12	18	12
b	18	12	6
a%b	12	6	0



public:

```
fraction(int a=0,int b=1); //构造函数
fraction add(fraction);    //两分数相加
//c=a.add(b); 执行后a不变, b不变, c是和
fraction sub(fraction);    //本分数减去实参分数
fraction mul(fraction);    //两分数相乘
fraction div(fraction);    //本分数除以实参分数
fraction reciprocal();     //求倒数
//b=a.reciprocal();执行后a不变, b是a的倒数
bool equal(fraction);      //等于运算
bool greaterThan(fraction); //大于运算
bool lessThan(fraction);   //小于运算
void display();            //显示分数
void input();              //输入分数
```

```
};
```

```

int main(){
    fraction f1(-3,-5),f2(-3,5),f3(3,-7),f4,f5(8);
    cout<<"f1"; f1.display();
    cout<<"f2"; f2.display();
    cout<<"f3"; f3.display();
    cout<<"f4"; f4.display();
    cout<<"f5"; f5.display();
    if(f1.greaterThan(f2)) cout<<"f1>f2"<<endl;
    if(f2.lessThan(f3)) cout<<"f2<f3"<<endl;
    if(f1.equal(f1)) cout<<"f1==f1"<<endl;
    f4=f1.add(f3);
    cout<<"f4=f1+f3"; f4.display();
    f4=f1.sub(f2);
    cout<<"f4=f1-f2"; f4.display();
    f4=f1.mul(f3);
    cout<<"f4=f1*f3"; f4.display();
    f4=f2.div(f3);
    cout<<"f4=f2/f3"; f4.display();
    f4=f2.reciprocal();
    cout<<"f4=1/f2"; f4.display();
    f4.input();
    cout<<"f4"; f4.display();
    return 0;
}

```

f1为:  $3/5$   
 f2为:  $-3/5$   
 f3为:  $-3/7$   
 f4为:  $0/1$   
 f5为:  $8/1$   
 f1>f2  
 f2<f3  
 f1==f1  
 f4=f1+f3为:  $6/35$   
 f4=f1-f2为:  $6/5$   
 f4=f1\*f3为:  $-9/35$   
 f4=f2/f3为:  $7/5$   
 f4=1/f2为:  $-5/3$   
 请顺序输入分子和分母 (整数) :  
 3 6  
 f4为:  $1/2$

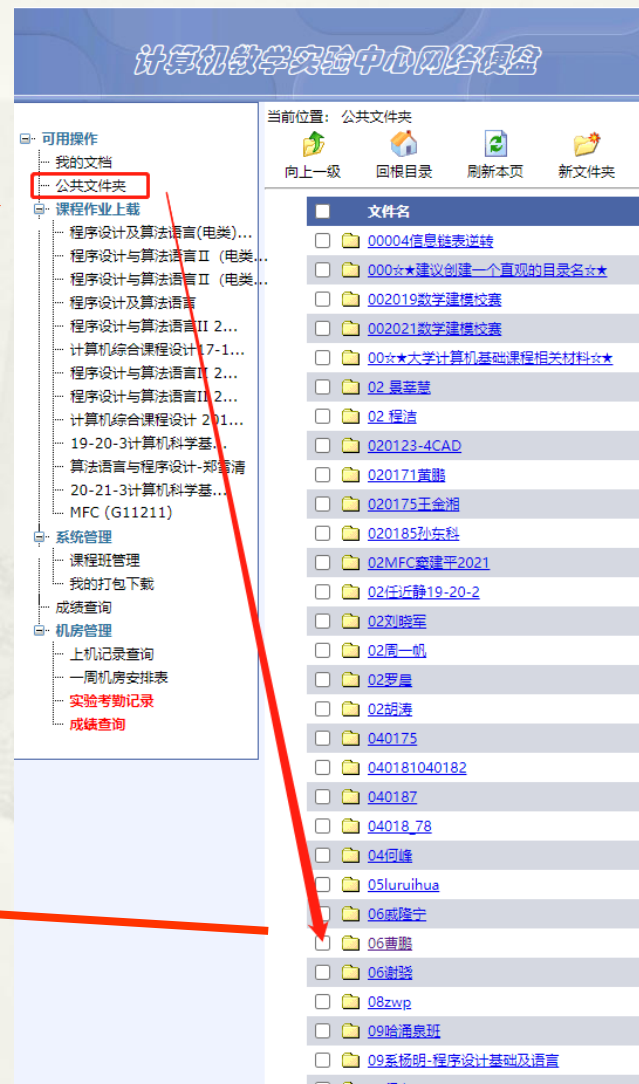
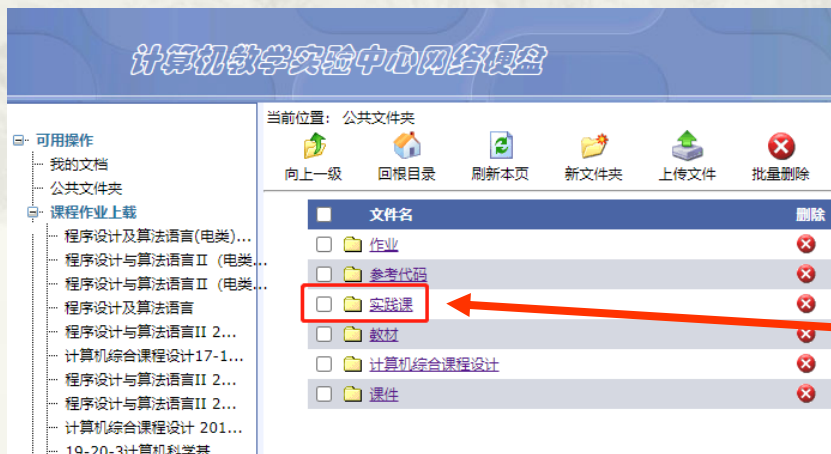
编程时间

~60min

讲解时间

~20min

cc.seu.edu.cn

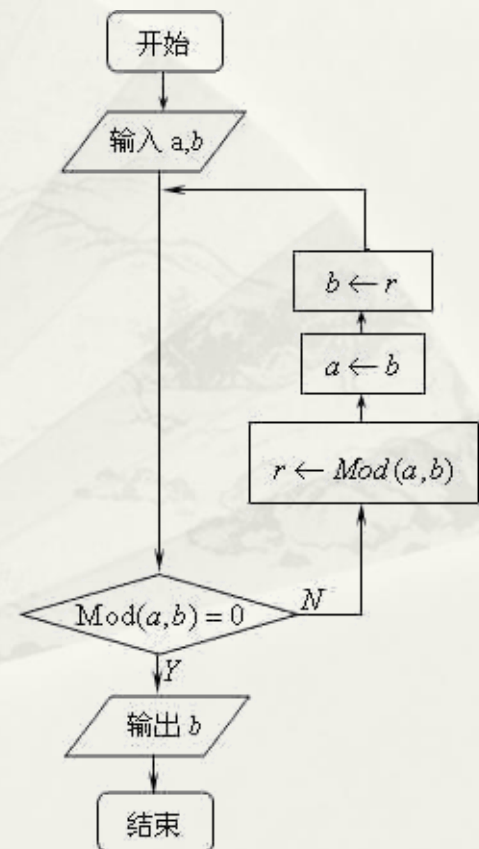




```

void fraction::reduction(){ //约分先求最大公约数
    int a,b,temp;
    if(below<0){
        above=-above;
        below=-below;
    }
    a=abs(above);
    b=abs(below);
    while(a%b){ //欧几里德法求最大公约数
        temp=a;
        a=b;
        b=temp%b; //temp%b即为r
    }
    above/=b;
    below/=b;
}

```



fraction fraction::makeCommond(fraction b){	类对象自身		参数b	
	分子	分母	分子	分母
int temp;	1	2	2	3
reduction(); b.reduction();				
above*=b.below; b.above*=below;	3	2	4	3
temp=below*b.below;				
below=b.below=temp;	3	6	4	6
return b;				
}				



```
fraction::fraction(int a,int b){ //构造函数
    above=a; below=b;
}
fraction fraction::add(fraction b){
    fraction temp;
    b=makeCommond(b); //通分
    temp.above=above+b.above;
    temp.below=below;
    temp.reduction(); //约分
    return temp;
}
fraction fraction::sub(fraction b){
    fraction temp;
    b=makeCommond(b); //通分
    temp.above=above-b.above;
    temp.below=below;
    temp.reduction(); //约分
    return temp;
}
```

```
fraction fraction::mul(fraction b){
    fraction temp;
    temp.above=above*b.above;
    temp.below=below*b.below;
    temp.reduction();    //约分
    return temp;
}
fraction fraction::div(fraction b){
    fraction temp;
    if(b.above==0){
        cout<<"零不能作除数! "<<endl;
        exit(1);
    }
    temp.above=above*b.below;
    temp.below=below*b.above;
    temp.reduction();    //约分
    return temp;
}
```

```
fraction fraction::reciprocal(){
    fraction temp;
    temp.above=below;
    temp.below=above;
    temp.reduction();    //约分
    return temp;
}
void fraction::display(){
    reduction();        //约分
    cout<<"为: "<<above<<"/"<<below<<endl;
}
```

```
bool fraction::equal(fraction b){  
    b=makeCommond(b);    //通分  
    return(above==b.above);  
}  
bool fraction::greaterThan(fraction b){  
    b=makeCommond(b);    //通分  
    return(above>b.above);  
}  
bool fraction::lessThan(fraction b){  
    b=makeCommond(b);    //通分  
    return(above<b.above);  
}
```

```
void fraction::input(){
    while(1){
        cout<<"请顺序输入分子和分母（整数）："<<endl;
        cin>>above>>below;
        if(below==0) cout<<"分母不可为零！"<<endl;
        else{
            reduction();
            return;
        }
    }
}
```

---



# End