

上机作业十

姓名	学号	日期
袁宇昊	201611130126	2018.11.22

实验目的

- 操作符重载的应用。

实验总结：

请在以下总结实验中发现的问题和解决办法或心得体会。请勿黏贴过多源码。

- 问题：**cout<<a+b;报错 其中a和b是RMB类型元素。

解决：将重载函数'<<'的变量声明改为const，因为一般的引用无法绑定到右值。

```
friend ostream& operator << (ostream& os,RMB& tm); //原定义 报错
friend ostream& operator << (ostream& os,const RMB& tm); //加上const不报错

//原来会报错的语句：
RMB a,b;
cout<<a+b;
```

- 问题：**使用了总价值的概念来简化人民币的计算：

```
int RMB::tot() //表示人民币的总价值
{
    int f=1;
    if(flag=='-') //符号位
        f=-1;
    return f*(Yuan*100+Jiao*10+Fen*1);
}
```

和它配套的有一个trans()函数，把总价值返回给元角分和符号位。

```

void RMB::trans(int total)
{
    flag='+';
    if(total<0)//符号位
        total*=-1, flag='-';
    Fen=total%10;    total/=10;
    Jiao=total%10;   total/=10;
    Yuan=total;
    return ;
}

```

3. 问题：构造函数的写法：共写了三个构造函数

```

RMB(char f1,int y,int j,int f=0);//一般化的构造函数
RMB(char f1,int tot=0);//带符号位的构造函数
RMB(int tot=0);//只有总价值的构造函数

```

其实只用到了第三个构造函数：

```

RMB::RMB(int total)
{
    trans(total);//引用了trans()函数，因为这个构造的算法和trans()完全一样。
}

```

4. 问题：自加写法：

解决：既要返回值，又要更新原来的值。

```

RMB RMB::operator ++()//前缀
{
    trans(tot()+1);
    return RMB(tot());
}
RMB RMB::operator ++(int)//后缀
{
    trans(tot()+1);
    return RMB(tot()-1);
}

```

5. 问题：符号位的处理

解决：符号用的是char类型保存的 '-'表示负值， '+'表示正值。涉及符号位的处理有读入、求价值和、转换。

符号位的读入（重载>>符号）：

```

istream& operator >> (istream& is,RMB& tm)
{
    is>>tm.flag>>tm.Yuan>>tm.Jiao>>tm.Fen;
    tm.trans(tm.tot());//标准化
}

```

求价值和：

```
int RMB::tot()//表示人民币的总价值
{
    int f=1;
    if(flag=='-')//符号位
        f=-1;
    return f*(Yuan*100+Jiao*10+Fen*1);
}
```

转换：

```
void RMB::trans(int total)
{
    flag='+';
    if(total<0)//符号位
        total*=-1,flag='-';
    Fen=total%10;    total/=10;
    Jiao=total%10;   total/=10;
    Yuan=total;
    return ;
}
```