

上机作业四

姓 名	学 号	日 期
袁宇昊	201611130126	2018.10.09

实验目的

- 无参数构造函数的定义和使用。
- 拷贝构造函数的定义和使用。
- 使用构造函数的初始化列表。
- 析构函数的定义和使用。

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。

1. 问题：调用以下除法函数报错：

```
double Number::division(void)
{
    return n1/n2;
}
```

解决：因为n2可能为0，而0作除数会报错，修改代码如下：

```
double Number::division(void)
{
    if(fabs(n2)<=0.0001)
        return INF;
    return n1/n2;
}
```

其中INF表示无穷大。

2. 问题：以下声明函数报错

```
CMoney();
CMoney(int newY=0,int newJ=0,int newF=0);
```

解决: 第一个函数是无参量构造函数, 第二个函数是带缺省值得构造函数, 对于CMoney a这一语句来说, 就会出现重载函数的二义性问题, 所以报错, 修改代码如下

```
CMoney();  
CMoney(int newY, int newJ, int newF);
```

因为无参数构造函数CMoney()中就把三个变量都初始化成了0, 所以没有必要再使用带缺省值的函数, 直接改用带参数的函数。

3. 问题: 析构函数里面要写啥

解决: 在第二个作业里, 没有指针变量, 所以不需要delete掉指针, 然后就没有什么需要特别处理的地方了。所以析构函数里除了一句cout, 没有其他的语句。

```
CMoney::~~CMoney()  
{  
    cout<<"析构函数被调用"<<endl;  
}
```

4. 问题: 人民币的进位

解决: 类似于十进制的进位法:

```
CMoney::CMoney(int newY, int newJ, int newF)  
{  
    Fen=newF%10;  
    newJ+=newF/10;  
    Jiao=newJ%10;  
    newY+=newJ/10;  
    Yuan=newY;  
    cout<<"有参数构造函数被调用"<<endl;  
}
```

5. 问题: 第三个作业中涉及到了浮点数的比较。而浮点数在比较时不够精确, 特别是在判断两浮点数是否相等的时候有很大的误差。 **解决:** 使用sig函数 让浮点数有一个非常小的模糊值 (eps), 这样可以大大减少误差。

```
const double eps=1e-6;  
int sig(double a)  
{  
    if(a>eps)  
        return 1;  
    else if(a<=-eps)  
        return -1;  
    else  
        return 0; //非常接近0的浮点数 就把它当作0  
}
```

6. 心得: 判断三角形的类型要注意三角形不同类型的包涵关系: 等腰三角形包含了等边三角形, 直角三角形包含了等腰直角三角形等。要注意判断类型时调用函数的顺序。

```

int Triangle::type(void)
{
    if(Isisorig())//等腰直角三角形
        return 4;
    if(Isrig())//直角三角形
        return 3;
    if(Isequ())//等边三角形
        return 2;
    if(Isiso())//等腰三角形
        return 1;
    return 0;
}

```

7. **心得:** 计算三角形面积: 海伦公式:

$$\sqrt{p(p-a)(p-b)(p-c)}$$

其中

$$p = (a + b + c) / 2$$

8. **问题:** 下列代码报错:

```

cout<<"第"<<i+1<<"个结点是";
a[i].printNode();
cout<<"第"<<i+1<<"个结点的左结点是";
a[i].nextNodeLeft()->printNode();
cout<<"第"<<i+1<<"个结点的右结点是";
a[i].nextNodeRight()->printNode();

```

解决: 因为有的结点的左结点（右结点）为空，不能调用->printNode()。修改如下:

```

cout<<"第"<<i+1<<"个结点是";
a[i].printNode();
if(a[i].nextNodeLeft() != NULL)
{
    cout<<"第"<<i+1<<"个结点的左结点是";
    a[i].nextNodeLeft()->printNode();
}
else
    cout<<"第"<<i+1<<"个结点的左结点是空结点"<<endl;
if(a[i].nextNodeRight() != NULL)
{
    cout<<"第"<<i+1<<"个结点的右结点是";
    a[i].nextNodeRight()->printNode();
}

```

预先判断一下左右结点是否为空。

