

面向对象程序设计基础作业五 设计文档

1. 模型部分

a. 功能简述

本程序内包含一个复数类 `cpxnum`，可以实现复数的储存、计算工作（详见 b）。在运行时，本程序将对包含的复数类进行测试（详见 2），并输出结果。下图为测试过程中两个样例输出。

```
TESTING FOR z = 1.5+2i
z + 1.5 = 3+2i
z - 1.5 = 2i
z * 4.5 = 6.75+9i
z / 4.5 = 0.333333+0.444444i
1.5 + z = 3+2i
1.5 - z = -2i
4.5 * z = 6.75+9i
4.5 / z = 2.7-3.6i
After ans = z++, ans = 1.5+2i, z = 2.5+2i
After ans = z--, ans = 2.5+2i, z = 1.5+2i
After ans = ++z, ans = 2.5+2i, z = 2.5+2i
After ans = --z, ans = 1.5+2i, z = 1.5+2i
PRESS ENTER TO CONTINUE
```

```
TESTING FOR w = 1.5+2i and z = 1.5
w + z = 3+2i
w - z = 2i
w * z = 2.25+3i
w / z = 1.5+2i
PRESS ENTER TO CONTINUE
```

b. 数据结构

```
23  ➡ cpxnum operator+ (cpxnum&) const;
24  ➡ cpxnum operator+ (double) const;
25  ➡ cpxnum operator- (cpxnum&) const;
26  ➡ cpxnum operator- (double) const;
27  ➡ cpxnum operator* (cpxnum&) const;
28  ➡ cpxnum operator* (double) const;
29  ➡ cpxnum operator/ (cpxnum&) const;
30  ➡ cpxnum operator/ (double) const;
31
32  ➡ cpxnum operator++ (int);
33  ➡ cpxnum& operator++ ();
34  ➡ cpxnum operator-- (int);
35  ➡ cpxnum& operator-- ();
36
37  ➡ bool operator== (cpxnum&) const;
38  ➡ bool operator== (double) const;
39
40  ➡ cpxnum& operator= (const cpxnum&);
41  ➡ cpxnum& operator= (double);
```

```
static double real(cpxnum&);
static double imag(cpxnum&);
static double abs(cpxnum&);
static cpxnum conj(cpxnum&);
```

本程序包含复数类 `cpxnum`，有两个 `double` 型私有参数 `real_part` 和 `imaginary_part`，分别存储复数的实数部分和虚数部分。`cpxnum` 内有四个公有接口，分别用于调用本复数的实部、虚部、模长和共轭。另外，`cpxnum` 类重载了如上左图所示的操作符，在 c. 算法部分将一一说明。

c. 算法

i. 自增、自减操作符

1. 前++、前--

此操作符可以使被操作数的实部增加/减少 1，虚部不变。用在表达式中时，先改变被操作数的值，再获取改变后被操作数的值。

2. 后++、后--

此操作符可以使被操作数的实部增加/减少 1，虚部不变。用在表达式中时，先获取被操作数的值，再改变被操作数的值。

ii. 四则运算符+ - × /

这些操作符的功能和它们数学上的功能相一致。本程序中的 `cpxnum` 类允许复数和实数混合运算，且顺序可交换。

特别注明：本程序没有进行“除以零”判断。鉴于计算过程的“除以零”异常会被正常抛出，在实际使用时和实数运算中处理除以零错误的方式是一样的。

iii. 赋值操作符

本程序为 `cpxnum` 类重载了赋值操作符 `=`，支持将复数或实数赋值给 `cpxnum` 类实例。当将实数赋值给 `cpxnum` 类时，`cpxnum` 类会将虚部设置为零。

iv. 逻辑运算符

本程序为 `cpxnum` 类重载了逻辑运算符 `==`，支持与另一个复数或实数相比较，返回二者是否相等。

其他的逻辑运算符没有重载，原因是复数不可比较大小。

v. 插入流运算符 (std::ostream) <<

为了更便捷地输出复数，本程序为 std::ostream 类重载了针对复数的插入流运算符。被调用时，会以 $a+bi$ 的形式输出复数结果；如果 a 或 b 等于零，对应的部分不会显示。如果 a 与 b 都等于零，会显示 0. 如果 a 或 b 等于正负 1，不会输出此多余的 1.

2. 验证部分

本程序的验证部分使用如下测试样例：

复数编号	值		代表的等价类
	实部	虚部	
z_1	1.5	2	普通复数
z_2	1.5	0	实数
z_3	0	0	零
z_4	0	1.5	纯虚数
z_5	-1	-1	实部与虚部为负数的普通复数

a. 仅涉及一个复数的运算符测试

对于上表每一个复数，分别做如下测试：输出它的值；复数与实数四则运算；实数与复数四则运算；自增（两种）；自减（两种）。这些测试可以验证重载的四则运算操作符（ \cdot 为 $+$ $-$ \times $/$ 之一）是否可用于复数·实数和实数·复数两种情况，验证自增、自减操作符是否能改变复数的值，比较前自增（减）和后自增（减）操作符的结果差异，以及测试插入流运算符的输出是否符合格式。

结果见 Table 1。

b. 二元运算符的测试

对于上表每两个不相同的复数，分别做如下测试：输出它们的值；复数与复数四则运算。这些测试主要是为了测试复数之间重载的四则运算符。

结果见 Table 2。

Table 1 涉及一个复数的运算符验证结果

序号	cout << z	z+1.5	z-1.5	z*1.5	z / 1.5	z++的值	++z 的值	z--的值	--z 的值
		1.5+z	1.5-z	1.5*z	1.5 / z	z++后 z 的值	++z 后 z 的值	z--后 z 的值	--z 后 z 的值
z ₁	1.5+2i	3+2i	2i	6.75+9i	0.33+0.44i	1.5+2i	2.5+2i	2.5+2i	1.5+2i
		3+2i	-2i	6.75+9i	0.33+0.44i	2.5+2i	1.5+2i	2.5+2i	1.5+2i
z ₂	1.5	3	0	6.75	0.33	1.5	2.5	2.5	1.5
		3	0	6.75	4.5	2.5	1.5	2.5	1.5
z ₃	0	1.5	-1.5	0	0	0	1	1	0
		1.5	1.5	0	NAN ¹	1	0	1	0
z ₄	1.5i	1.5+1.5i	-1.5+1.5i	6.75i	0.33i	1.5i	1+1.5i	1+1.5i	1.5i
		1.5+1.5i	1.5-1.5i	6.75i	-4.5i	1+1.5i	1.5i	1+1.5i	1.5i
z ₅	-1-i	0.5-i	-2.5-i	-4.5-4.5i	-0.22-0.22i	-1-i	-i	-i	-1-i
		0.5-i	2.5+i	-4.5-4.5i	-3.18+3.18i	-i	-1-i	-i	-1-i

¹ NAN 为除以零异常，是本程序意料之内。下文同。详见 1.c.ii.

Table 2 涉及两个复数的运算符验证结果

第一操作数	第二操作数	相加	相减	相乘	相除
z_1	z_2	$3+2i$	$2i$	$2.25+3i$	$1.5+2i$
	z_3	$1.5+2i$	$1.5+2i$	0	NAN
	z_4	$1.5+3.5i$	$1.5+0.5i$	$-3+2.25i$	$2-1.5i$
	z_5	$0.5+i$	$2.5+3i$	$0.5-3.5i$	$-2.47-0.35i$
z_2	z_1	$3+2i$	$-2i$	$2.25+3i$	$0.9-1.2i$
	z_3	1.5	1.5	0	NAN
	z_4	$1.5+1.5i$	$1.5-1.5i$	$2.25i$	$-1.5i$
	z_5	$0.5-i$	$2.5+i$	$-1.5-1.5i$	$-1.06+1.06i$
z_3	z_1	$1.5+2i$	$-1.5-2i$	0	0
	z_2	1.5	-1.5	0	0
	z_4	$1.5i$	$-1.5i$	0	0
	z_5	$-1-i$	$1+i$	0	0
z_4	z_1	$1.5+3.5i$	$-1.5-0.5i$	$-3+2.25i$	$1.2+0.9i$
	z_2	$1.5+1.5i$	$-1.5+1.5i$	$2.25i$	$1.5i$
	z_3	$1.5i$	$1.5i$	0	NAN
	z_5	$-1+0.5i$	$1+2.5i$	$1.5-1.5i$	$-1.06-1.06i$
z_5	z_1	$0.5+i$	$-2.5-3i$	$0.5-3.5i$	$-1.4+0.2i$
	z_2	$0.5-i$	$-2.5-i$	$-1.5-1.5i$	$-1-i$
	z_3	$-1-i$	$-1-i$	0	NAN
	z_4	$-1+0.5i$	$-1-2.5i$	$1.5-1.5i$	$-1+i$

