

**Практическая работа. Абстрактный тип данных «комплексное число»**

Тема: Классы C++

*Цель: Сформировать практические навыки реализации абстрактных типов данных в соответствии с заданной спецификацией с помощью классов C++. Синтаксис классов: инкапсуляция, простые свойства.*

**Задание**

1. Реализовать абстрактный тип данных «комплексное число», используя класс C++, в соответствии с приведенной ниже спецификацией.
2. Протестировать каждую операцию, определенную на типе данных, используя средства модульного тестирования.
3. Если необходимо, предусмотрите возбуждение исключительных ситуаций.

**Спецификация типа данных «комплексное число».****ADT TComplex**

**Данные** Комплексное число TComplex - это неизменяемая пара вещественных чисел, представляющие действительную и мнимую части комплексного числа ( $a + i*b$ ).

**Операции**

Операции могут вызываться только объектом комплексное число (тип TComplex), указатель на который в них передаётся по умолчанию. При описании операций этот объект называется `this` «само число».

<b>КонструкторЧисло</b>	
Вход:	Пара вещественных чисел $a$ и $b$ .
Предусловия:	Нет.
Процесс:	Устанавливает значения $a$ , $b$ в поля экземпляра класса комплексное число (тип TComplex) <code>this.a</code> -

	действительной частью и this.b мнимая часть.  Например: <i>КонструкторЧисло</i> (6,3)=6 + i*3 <i>КонструкторЧисло</i> (3,0)=3 + i*0 <i>КонструкторЧисло</i> (0,0)=0 + i*0
Постусловия:	Поля объекта this инициализированы входными данными.
Выход:	Нет.
<b><i>КонструкторСтрока</i></b>	
Вход:	Строка f, представляющая комплексное число.
Предусловия:	Изображение комплексного числа во входной строке f должно быть представлено в заданном формате.
Процесс:	Выделяет из строки $f = 'a + i*b'$ , действительную часть (a) и комплексную часть (b) и преобразует их в число. Устанавливает значения a, b в поля экземпляра класса комплексное число (тип TComplex) this.a - действительной частью и this.b мнимая часть.  Например: <i>КонструкторСтрока</i> ('6+i*3') = 6+i*3 <i>КонструкторСтрока</i> ('0+i*3') = 0+i*3
Постусловия:	Поля объекта this инициализированы входными данными.
Выход:	Нет.
<b>Копировать:</b>	
Вход:	Нет.
Предусловия:	Нет.

Процесс	Создаёт и возвращает собственную копию - комплексное число (тип TComplex) с действительной и мнимой частями такими же, как у самого числа this.
Выход:	Комплексное число (тип TComplex). Например: $c = 6+i3$ , Копировать(c) = $6+i3$
Постусловия:	Нет.
<b>Сложить</b>	
Вход:	Комплексное число d (тип TComplex).
Предусловия:	Нет.
Процесс	Создаёт и возвращает комплексное число, полученное сложением самого числа this = $a1+i*b1$ с числом $d = a2+i*b2$ : $((a1+i*b1)+(a2+i*b2)=(a1+a2)+i*(b1+b2))$ . Например: $q = (2 +i*1)$ , $d = (2 +i*1)$ , $q.Сложить(d) = (4 +i*2)$ .
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Умножить</b>	
Вход:	Комплексное число d (тип TComplex).
Предусловия:	Нет.
Процесс	Создаёт и возвращает комплексное число, полученное умножением самого числа this = $a1+i*b1$ на число $d = a2+i*b2$ : $((a1+i*b1)*(a2+i*b2)=(a1*a2 - b1*b2)+i*(a1*b2+ a2*b1))$ .
Выход:	Комплексное число (тип TComplex).

Постусловия:	Нет.
<b>Квадрат</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Создаёт и возвращает комплексное число (тип TComplex), полученное умножением числа this на самого себя: $((a1+i*b1)*(a1+i*b1)=(a1*a1 - b1*b1)+i*(a1*b1 + a1*b1))$ .
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Обратное</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Создаёт и возвращает комплексное число (тип TComplex), полученное делением единицы на само число $1/((a1+i*b1) = a1/(a1**2 + b1**2) - i * b1/( a1**2 + b1**2 ))$ .
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Вычесть</b>	
Вход:	Комплексное число d (тип TComplex)..
Предусловия:	Нет.
Процесс	Создаёт и возвращает комплексное число (тип TComplex), полученное вычитанием d = a2 + i b2 из самого числа this = (a1+i*b1): $(a1+i*b1)-(a2+i*b2)=(a1-a2)+i*(b1-b2)$ .

	<p>Например:</p> $q = (2 + i*1), d = (2 + i*1)$ $q.\text{Вычесть}(d) = (0 + i0).$
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Делить</b>	
Вход:	Комплексное число d.
Предусловия:	Нет.
Процесс	Создаёт и возвращает комплексное число (тип TComplex), полученное делением самого числа this на число (d) $((a1+i*b1)/(a2+i*b2)=(a1*a2 + b1*b2)/(a2**2 + b2**2)+i*(a2*b1 - a1*b2)/(a2**2 + b2**2)).$
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Минус</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Создаёт и возвращает комплексное число (тип TComplex), являющееся разностью комплексных чисел z и i самого числа this, где z – комплексное число (0+i0).
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Модуль</b>	
Вход:	Нет.
Предусловия:	Нет.

Процесс	Вычисляет и возвращает модуль самого комплексного числа <code>this</code> . Например: $q = (2 + i*1)$ , <code>q. Модуль</code> = $\sqrt{2^2+1^2}$ . $q = (i*17)$ , <code>q. Модуль</code> = $\sqrt{0^2+17^2}$ .
Выход:	Вещественное число.
Постусловия:	Нет.
<b>УголРад</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Возвращает аргумент <code>fi</code> самого комплексного числа <code>this</code> (в радианах). $fi = (\arctg(b/a), a > 0; \pi/2, a = 0, b > 0; \arctg(b/a) + \pi, a < 0; -\pi/2, a = 0, b < 0)$ . Например: $q = (1 + i*1)$ , <code>q. УголРад</code> = 0,79.
Выход:	Вещественное число.
Постусловия:	Нет.
<b>УголГрад</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Возвращает аргумент самого комплексного числа <code>this</code> (в градусах). Например: $q = (1 + i*1)$ , <code>q. Град</code> = 45.
Выход:	Вещественное число.
Постусловия:	Нет.

<b>Степень</b>	
Вход:	Целое n.
Предусловия:	Нет.
Процесс	Возвращает целую положительную степень n самого комплексного числа this. $\text{this}^n = r^n(\cos(n \cdot fi) + i \cdot \sin(n \cdot fi))$ .
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Корень</b>	
Вход:	Целое n, целое i.
Предусловия:	Нет.
Процесс	Возвращает i-ый корень целой положительной степени n самого комплексного числа this. $\sqrt[n]{\text{this}} = \sqrt[n]{r} \cdot (\cos((fi + 2 \cdot k \cdot \pi)/n) + i \cdot \sin((fi + 2 \cdot k \cdot \pi)/n))$ . При этом коэффициенту k придается последовательно n значений: $k = 0, 1, 2, \dots, n - 1$ и получают n значений корня, т.е. ровно столько, каков показатель корня.
Выход:	Комплексное число (тип TComplex).
Постусловия:	Нет.
<b>Равно</b>	
Вход:	Комплексное число d.
Предусловия:	Нет.
Процесс	Сравнивает само комплексное число this с числом d. Возвращает значение True, если они - тождественные комплексные числа, и значение False - в противном случае.
Выход:	Булевское значение.

Постусловия:	Нет.
<b>НеРавно</b>	
Вход:	Комплексное число d.
Предусловия:	Нет.
Процесс	Сравнивает само комплексное число this с числом d. Возвращает значение True, если само число $\neq$ d, - значение False - в противном случае.
Выход:	Булевское значение.
Постусловия:	Нет.
<b>ВзятьReЧисло</b>	
Вход:	Нет
Предусловия:	Нет.
Процесс	Возвращает значение действительной части самого комплексного числа this в числовом формате.
Выход:	Вещественное значение.
Постусловия:	Нет.
<b>ВзятьIm Число</b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Возвращает значение мнимой части самого комплексного числа this в числовом формате.
Выход:	Вещественное значение.
Постусловия:	Нет.
<b>ВзятьReСтрока</b>	
Вход:	Нет.



Предусловия:	Нет.
Процесс	Возвращает значение вещественной части самого комплексного числа <code>this</code> в строковом формате.
Выход:	Строка.
Постусловия:	Нет.
<b><i>ВзятьImСтрока</i></b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Возвращает значение мнимой части самого комплексного числа <code>this</code> в строковом формате.
Выход:	Строка.
Постусловия:	Нет.
<b><i>ВзятьКомплексноеСтрока</i></b>	
Вход:	Нет.
Предусловия:	Нет.
Процесс	Возвращает значение самого комплексного числа <code>this</code> в строковом формате.
Выход:	Строка.
Постусловия:	Нет.

***end TComplex***

Рекомендации к выполнению

1. Тип данных реализовать, используя класс C++ в составе проекта Консольное приложение.
2. Для возбуждения исключений опишите класс исключительных ситуаций, как наследника библиотечного класса.
3. Для тестирования используйте проект модульного теста.
4. Тип данных реализовать в отдельном модуле в консольном приложении.

## Содержание отчета

1. Задание.
2. Текст программы.
3. Тестовые наборы данных для тестирования типа данных.

## Контрольные вопросы

1. Что такое инкапсуляция?
2. Как синтаксически представлено поле в описании класса?
3. Как синтаксически представлен метод в описании класса?
4. Как синтаксически представлено простое свойство в описании класса?
5. Особенности описания методов класса?
6. Особенности описания и назначение конструктора класса?
7. Видимость идентификаторов в описании класса?
8. Особенности вызова методов применительно к объектам класса?