

Avaliação Prática 01

Estrutura de Dados - Prof. Hamilton José Brumatto

Problema C: Números complexos

Arquivo: complexo.[h,hpp,py]

O Problema:

O conjunto de números \mathcal{C} - complexo, representa um grupo que permite operações não atingidas pelos números reais. Cada número é representado por duas partes, uma parte real e uma parte puramente complexa: $n = a + bi$. Neste caso, i é a base dos complexos e podemos simplificar como: $i = \sqrt{-1}$. Na numeração complexa, as partes a e b , são representada por números reais. Com base neste conjunto pode-se estabelecer as operações básicas:

Atribuição	$(a + bi) \rightarrow (a + bi)$
Simétrico	$-(a + bi) \rightarrow (-a - bi)$
Soma	$(a + bi) + (c + di) = (a + c) + (b + d)i$
Subtração	$(a + bi) - (c + di) = (a - c) + (b - d)i$
Multiplicação	$(a + bi) \times (c + di) = (ac - bd) + (ad + bc)i$
Divisão*	$\frac{a + bi}{c + di} = \frac{(ac + bd) + (-ad + bc)i}{c^2 + d^2}$
Módulo	$ a + bi = \sqrt{a^2 + b^2}$
Real	$(a + bi) \rightarrow a$
Imaginário	$(a + bi) \rightarrow b$

* A operação de divisão é fácil entender ao multiplicar divisor e dividendo pelo complemento do divisor: $(c - di)$.

Em especial, não há comparação de ordem entre números complexos. Foi pedido a você para criar um tipo abstrato / classe complexo, e as funções / operadores para estes números. Ou seja, na linguagem C deve-se criar o tipo complexo e será necessário implementar os seguintes protótipos:

```
typedef complexo {  
    ...  
} complexo;  
void cria(complexo *s, double a, double b);  $\rightarrow s = a + bi$   
void atribui(complexo *s, complexo a);  $\rightarrow s = a$   
void simetrico(complexo *s, complexo a);  $\rightarrow s = -a$   
void soma(complexo *s, complexo a, complexo b);  $\rightarrow s = a + b$   
void subtrai(complexo *s, complexo a, complexo b);  $\rightarrow s = a - b$   
void multiplica(complexo *s, complexo a, complexo b);  $\rightarrow s = a \times b$   
void divide(complexo *s, complexo a, complexo b);  $\rightarrow s = \frac{a}{b}$   
double modulo(complexo c);  $c \rightarrow |c|$   
double real(complexo c);  $c = (a + bi) \rightarrow a$   
double imaginario(complexo c);  $c = (a + bi) \rightarrow b$ 
```

Ou, então, na linguagem C++, deve-se criar a classe complexo e será necessário implementar os seguintes protótipos de métodos e sobreposições de operações:

```
class complexo {  
    ...  
public:  
    complexo(double a, double b); criação  $\rightarrow *this = a + bi$   
    complexo operator=(complexo c); atribuição  $\rightarrow *this = c$   
    complexo operator-(); simétrico  $\rightarrow -*this$   
    complexo operator+(complexo c); soma  $\rightarrow *this + c$   
    complexo operator-(complexo c); subtração  $\rightarrow *this - b$   
    complexo operator*(complexo c); multiplicação  $\rightarrow *this \times c$   
    complexo operator/(complexo c); divisão  $\rightarrow \frac{*this}{c}$   
    double modulo();  $\rightarrow |*this|$   
    double real();  $*this = (a + bi) \rightarrow a$   
    double imaginario();  $*this = (a + bi) \rightarrow b$   
};
```

Ou, então, em Python, deve-se criar a classe complexo e será necessário implementar os seguintes protótipos de métodos e sobreposições de operações:

```

class complexo:
    ...
    def __init__(self, a, b): criação  $\rightarrow self = a + bi$ 
    def __neg__(self): simétrico  $\rightarrow -self$ 
    def __add__(self, c): soma  $\rightarrow self + c$ 
    def __sub__(self, c): subtração  $\rightarrow self - b$ 
    def __mul__(self, c): multiplicação  $\rightarrow self \times c$ 
    def __truediv__(self, c): divisão  $\rightarrow \frac{self}{c}$ 
    def modulo(self):  $\rightarrow |self|$ 
    def real(self):  $self = (a + bi) \rightarrow a$ 
    def imaginario(self):  $self = (a + bi) \rightarrow b$ 

```