

ELE078 - Programação Orientada a Objetos

Atividade Prática 03

Constantes, Funções inline e Atributos static:

Escreva uma classe Ponto2D em C++ para manipular pontos no espaço cartesiano bidimensional. Os objetos Ponto2D devem ter as coordenadas x e y do tipo *double*.

Recomendações:

- Use sobrecarga de funções sempre que possível;
- Funções membro que não alteram o estado dos objetos devem ser `const`.
- Use, sempre que possível, passagem de argumentos por referência implícita, com o modificador `const` para argumentos cujo valor não deve ser alterado.
- avaliem a possibilidade de retornar por objetos por cópia ou por referência.
- avaliem a possibilidade de usarem funções inline para otimização do código.

A classe deve possuir:

- construtores que permitem a inicialização de objetos do tipo Ponto2D das seguintes formas:

In []:

```
Ponto2D p1;           // inicializa o ponto2D p1 com as coordenadas x = 0.0 e y = 0.0;
Ponto2D p2(3.0, 4.0); // inicializa o ponto2D p2 com as coordenadas x = 3.0 e y = 4.0;
```

Cada objeto Ponto2D inicializado deve ter um identificador (*id*) que é incrementado cada vez que um novo objeto é criado e decrementado cada vez que um novo Objeto é destruído; O primeiro Objeto Ponto2D criado deve ter o *id* = 1, o segundo *id* = 2, e assim por diante.

- vcs podem criar uma função membro (`private`) que retorna o próximo *id* disponível: `getNextId()`, para facilitar o processo de inicialização dos Ponto2D.

Funções membro *get* and *set* para acessar e alterar os atributos dos objetos:

In []:

```
// Função membro para escrever (imprimir) as coordenadas do ponto na tela
p1.print();
```

In []:

```
// Função membro para calcular a distância do ponto a origem do sistema de coordenadas cartesiano
double d = p1.distToOrig();
```

In []:

```
// Função membro para calcular a distância entre dois pontos p1 e p2, onde p1 é o objeto que chama a função  
double d = p1.distTo(p2);
```

In []:

```
// Função membro que modifica as coordenadas do objeto corrente p1 somando às suas coordenadas um novo ponto p2, que deve ser passado por argumento.  
p1.sumOf(p2);
```

In []:

```
// Função membro que retorna um ponto p3 que é o resultado da soma entre dois pontos p1 e p2, onde p1 é o objeto que chama a função.  
p3 = p1.sumOf(p2);
```

Escreva um programa teste para testar todas as operações de sua classe.