

1. Classe Ponto2D:

Reescreva classe Ponto2D em C++ para manipular pontos no espaço cartesiano bidimensional. A classe deve possuir:

- construtor que permite a inicialização de objetos do tipo Ponto2D das seguintes formas:
 - Ponto2D p1; // inicializa as coordenadas x = 0.0 e y = 0.0 com o valor 0;
 - Ponto2D p2(3.0, 4.0);
 - Cada objeto inicializado 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.
 - Obs: use sobrecarga e argumentos default sempre que possível;
 - Obs: vcs podem criar uma função membro 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;
- Função membro para calcular a distância do ponto a origem;
- Função membro para calcular a distância entre dois pontos p1 e p2, onde p1 é o objeto que chama a função
- Função membro para escrever as coordenadas do ponto na tela;
- Função membro que modifica as coordenadas do objeto corrente p1 somando às suas coordenadas um novo ponto p2, que deve ser passado por argumento.
 - Ex: p1.somaCom(p2);
- 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.
 - Ex. p3 = p1.somaCom(p2);

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.

2. Construtor de cópia:

Crie uma classe contendo um double *. O construtor inicializa o double * chamando new double e associando ao conteúdo apontado pelo ponteiro o valor recebido como parâmetro. O destrutor imprime o conteúdo apontado pelo ponteiro e chama o delete para o ponteiro. Agora crie uma função que receba um objeto desta classe por valor e chame esta função no main(). O que acontece? Conserte o código criando o construtor de cópia