

POO - Atividade de Implementação 02

1. As matrizes correspondem a um componente matemático fundamental da Álgebra Linear. Na Engenharia de Controle, o uso de matrizes e os conceitos relacionados de Álgebra Linear desempenham papel importante, por exemplo, no projeto e implementação de controladores em Espaço de Estados. Sabendo disso, implemente uma classe denominada `Matriz` com o objetivo de abstrair alguns aspectos importantes desta entidade da matemática. A sua classe deve cobrir, pelo menos, os seguintes aspectos:

- Atributos para indicar o número de linhas e de colunas da matriz;
- A matriz do objeto da classe `Matriz` deve ser armazenada em uma matriz bidimensional alocada dinamicamente;
- Construtor que recebe um único parâmetro de entrada, que indica a ordem de uma matriz quadrada;
- Construtor que recebe como parâmetros os números de linhas e de colunas da matriz;
- A matriz do objeto deve ser inicializada com todos os seus elementos iguais a zero;
- O destrutor do objeto `Matriz` deve liberar a memória alocada dinamicamente para o objeto;
- Método que exibe em tela todos os elementos do objeto `Matriz`;
- Método que pode ser chamado para requisitar que o usuário informe todos os elementos da matriz do objeto, de acordo com a sua vontade. Este método pede que o usuário digite elemento a elemento da matriz;
- Método que preenche a matriz do objeto com números aleatórios entre um determinado valor mínimo e um valor máximo. Estes valores devem ser informados por meio dos parâmetros do método;
- Sobrecarga do operador `+`, que permite a realização da adição matricial no formato `objetoMatrizA + objetoMatrizB`;
- Sobrecargas do operador `+`, que permite a adição de uma matriz por um escalar nos seguintes formatos: `2 + objetoMatrizA` ou `objetoMatrizA + 2`;
- Sobrecarga do operador `-`, que permite a realização da subtração matricial no formato `objetoMatrizA - objetoMatrizB`;
- Sobrecargas do operador `-`, que permite a subtração de uma matriz por um escalar nos seguintes formatos: `2 - objetoMatrizA` ou `objetoMatrizA - 2`;
- Sobrecarga do operador `*`, que realiza a multiplicação matricial no formato `objetoMatrizA * objetoMatrizB`;

- Sobrecargas do operador `*`, que permite a multiplicação de uma matriz por um escalar nos seguintes formatos: `2 * objetoMatrizA` ou `objetoMatrizA * 2`;
- Sobrecarga do operador `=`, que realiza uma cópia adequada dos atributos de um objeto `Matriz` para um outro objeto `Matriz`. Por exemplo, `objetoMatrizA = objetoMatrizB`;
- Sobrecarga do operador `<<`, utilizado com o objeto `cout` para imprimir em tela um objeto da classe `Matriz`, exibindo os elementos de sua matriz. Exemplo: `cout << objetoMatriz`;
- Sobrecarga do operador `>>`, utilizado com o objeto `cin` para permitir que o usuário preencha todas as posições da matriz com os valores desejados. Exemplo: `cin >> objetoMatriz`;
- Sobrecarga do operador `[]`, que permite recuperar um elemento de uma determinada posição da matriz do objeto ou atribuir um valor a esta posição. Exemplos: `objetoMatriz[0][1] = 2` e `double numero = objetoMatriz[0][1]`;
- Método que retorna `true` se a matriz do objeto é diagonal e `false` caso contrário;
- Métodos acessores (`get` e `set`) que considerar necessários;
- Método que retorna a **matriz escalonada** do objeto da classe `Matriz`.