

Programação Modular - Exercícios

- Montar uma Biblioteca de vetores com as funções:
 - `Public static double [] lerVetorDouble(int n)`
 - Ler um vetor de double com “n” elementos, entrada o número de elementos do vetor (n), saída um vetor double preenchido;
 - `Public static void mostraVetorDouble (double [] v)`
 - Imprimir um vetor de double, entrada um vetor de double, não possui saída;
 - `Public static int [] somaVetores(int [] v1, int [] v2)`
 - Recebe dois vetores de inteiros (v1,v2) de mesmo tamanho e soma os vetores, elemento por elemento, retorna um novo vetor resultante;
 - `Public static int [] subtraiVetores(int [] v1, int [] v2)`
 - Recebe dois vetores de inteiros (v1,v2) de mesmo tamanho e subtrai os vetores, elemento por elemento, retorna um novo vetor resultante;

Programação Modular - Exercícios

- Montar uma Biblioteca de vetores com as funções:
 - `Public static int maiorVetor(int [] v)`
 - Recebe um vetor de inteiros e retorna o maior valor encontrado;
 - `Public static int posicaoMaiorVetor(int [] v)`
 - Recebe um vetor de inteiros e retorna a posição do maior valor encontrado;
 - `Public static int menorVetor(int [] v)`
 - Recebe um vetor de inteiros e retorna o menor valor encontrado;
 - `Public static int posicaoMenorVetor(int [] v)`
 - Recebe um vetor de inteiros e retorna a posição do maior valor encontrado;
 - `Public static double calculaMediaVetor (int [] v)`
 - Recebe um vetor de inteiros de entrada, calcula e retorna a média do vetor;

Programação Modular - Exercícios

- Montar uma Biblioteca de vetores com as funções:
 - `Public static int [] ordenaCrescenteVetor(int [] v)`
 - Recebe um vetor de inteiros e ordena o vetor na ordem crescente, retorna um novo vetor ordenado;
 - `Public static int [] ordenaDecrescenteVetor(int [] v)`
 - Recebe um vetor de inteiros e ordena o vetor na ordem decrescente, retorna um novo vetor ordenado;
 - `Public static void ordenaCrescenteVetor(int [] v)`
 - Recebe um vetor de inteiros e ordena o vetor na ordem crescente, sem retorno;
 - `Public static void ordenaDecrescenteVetor(int [] v)`
 - Recebe um vetor de inteiros e ordena o vetor na ordem decrescente, sem retorno;

Programação Modular - Exercícios

- Montar uma Biblioteca de vetores com as funções:
 - `Public static int [] interseccaoVetores(int [] v1, int [] v2)`
 - Recebe dois vetores de inteiros e realiza a operação de intersecção de vetores, não deve sobrar posições no vetor intersecção, retorna um novo com a intersecção dos vetores.
 - `Public static int [] uniaoVetores(int [] v1, int [] v2)`
 - Recebe dois vetores de inteiros e realiza a operação de união dos vetores, não deve sobrar posições no vetor união, retorna um novo com a união dos vetores.

Programação Modular - Exercícios

- Mote programas para testar as funções da biblioteca de Vetores;