

Lista 9

- 1) Analise o código e entenda-o, depois execute-o para ver os resultados.

Obs: Lembre-se da diferença de passagem de parâmetro por valor e por referência. Vetores em Java são sempre passados por referência.

```
public class Teste {
    public static void selecao(int[] array, int n) {
        for (int i = 0; i < (n - 1); i++) {
            int menor = i;
            for (int j = (i + 1); j < n; j++){
                if (array[menor] > array[j]){
                    menor = j;
                }
            }
            int temp = array[menor];
            array[menor] = array[i];
            array[i] = temp;
        }
    }

    public static void main(String[] args) {
        int[] vet = {1,6,0,3,7};
        selecao(vet, vet.length);
        for(int i = 0; i < vet.length; i++){
            System.out.print(vet[i]+" ");
        }
    }
}
```

- 2) Implemente o método de ordenação Bolha. No método main, solicite ao usuário que informe o tamanho do vetor que ele deseja ordenar, crie um vetor de inteiros com o tamanho informado, depois solicite ao usuário os números para preencher o vetor. Por fim, utilize o método Bolha para ordenar o vetor e imprima o vetor ordenado.
- 3) Implemente o método de ordenação Inserção. No método main, solicite ao usuário que informe o tamanho do vetor que ele deseja ordenar, crie um vetor de inteiros com o tamanho informado, depois solicite ao usuário os números para preencher o vetor. Por fim, utilize o método Inserção para ordenar o vetor e imprima o vetor ordenado.
- 4) Implemente o método de ordenação ShellSort. No método main, solicite ao usuário que informe o tamanho do vetor que ele deseja ordenar, crie um vetor de inteiros com o tamanho informado, depois solicite ao usuário os números para preencher o vetor. Por fim, utilize o método ShellSort para ordenar o vetor e imprima o vetor ordenado.

5) Ordene o vetor [10, 1, 3, 20, 5, 0, 1, -3, 9, 2, 2, 7, 3, 4] mostrando o conteúdo do vetor a cada passo intermediário. Utilize os seguintes algoritmos:

- a) Seleção: liste o vetor para cada elemento que atinja sua posição definitiva.
- b) Bolha: liste o vetor para cada elemento que atinja sua posição definitiva.
- b) Inserção: liste o vetor para cada elemento incluído na ordenação parcial até o momento.
- c) Shellsort: liste o vetor para cada novo valor de h , enquanto $h > 1$. Quando $h = 1$, liste o vetor para cada elemento inserido na ordem parcial.

Obs: O intuito desse exercício é que os alunos entendam a lógica de cada algoritmo de ordenação. Nesse exercício deverá ser feita a execução passo a passo dos algoritmos. Assim, nenhum código deve ser entregue, apenas a descrição do conteúdo do vetor a cada passo intermediário de cada algoritmo.

6) Considere os seguintes vetores:

- Vetor crescente: [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
- Vetor decrescente: [20,19,18,17,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1]
- Vetor aleatório: [20,14,1,3,2,4,11,7,8,9,17,18,16,15,6,5,13,12,10,19]

Preenche as duas tabelas a seguir, com o número de comparações e movimentações de registros do vetor que cada algoritmo faz para ordenar cada um dos três vetores.

Número de comparações

	Seleção	Bolha	Inserção	ShellSort
Vetor crescente				
Vetor decrescente				
Vetor aleatório				

Número de movimentações

	Seleção	Bolha	Inserção	ShellSort
Vetor crescente				
Vetor decrescente				
Vetor aleatório				

Dica: para preencher a tabela, adicione contadores nos métodos e execute os métodos para ordenar um dos três vetores.