

Material de Apoio Aula 14

Ordenação

Um problema frequentemente enfrentado em diversas aplicações é a ordenação de elementos que se encontram armazenados em um array. A ordenação visa organizar os elementos segundo alguma ordem pré-definida. Exemplos:

- Ordenar um array de inteiros em ordem crescente ou decrescente.
- Ordenar um array de tarefas em ordem de prioridade.
- Ordenar um array de veículos em uma frota em ordem de quilometragem percorrida.

Existem diversos algoritmos de ordenação. Os mais populares são o Quick Sort, Bubble Sort, Merge Sort, etc. Um algoritmo bastante simples é o algoritmo abaixo.

Algoritmo de troca simples

```
// ordenar um vetor v de inteiros
for( i = 0 ; i < v.length - 1 ; i++ )
    for( j = i ; j < v.length ; j++ )
        if( v[j] < v[i] ) {
            aux = v[i];
            v[i] = v[j];
            v[j] = aux;
        }
```

Exercícios

1. Implemente uma classe que manipule um array de inteiros. O número de elementos deste vetor deve ser passado no construtor e também no construtor deve inicializado cada elemento do vetor com um valor randômico entre 0 e 100. Implemente métodos para:
 - a) imprimir todo o array.
 - b) ordenar este array em ordem crescente.
 - c) ordenar este array em ordem decrescente.
2. Implemente uma classe `Tarefa`. Esta classe possui dois atributos, prioridade (`int`) e descrição (`String`). Estes dois atributos devem ser inicializados no construtor do objeto. Construa um array de `Tarefas`, inicialize este array com 5 tarefas (entradas pelo usuário). Por fim, imprima o array com suas tarefas organizadas em ordem crescente.
3. Faça o mesmo exercício 2, considerando uma classe `Aluno`, onde os nomes dos alunos devem ser apresentados de duas formas:
 - a) Ordenados por nota (maior nota para a menor).
 - b) Ordenados por nome (do menor para o maior).
4. Implementar o exercício 18 da lista de exercícios 7 do prof. Aníbal (disponível em <http://www.inf.unisinos.br/~anibal/prog1lis7.pdf>).