

Estrutura de dados – Java

Algoritmos e Orientação a Objetos

Os Algoritmos são sequências de passos que existem para resolver determinado problema, a ordem dos passos interfere diretamente no resultado.

O paradigma de Orientação a Objetos consiste em interpretar tudo como objetos (pessoas, objetos, espaço e ações). Com isso, a estrutura, e como os 'passos' são formados devem seguir as regras do paradigma.

Vetores (Arrays)

Os vetores são como gavetas de um armário, onde o mesmo, só pode guardar determinado tipo, em que cada posição (gaveta), que você determina quando monta a variável, pode guardar um valor ou objeto do mesmo tipo.

Exemplo:

```
Int[] vetor = new int[3]
```

Int[] -> tipo da variável

'vetor' -> nome da variável

new int -> instanciando como novo objeto int.

[3] -> tamanho do array (Quantidade de gavetas que o armário possui)

Listas Ligadas (LinkedList)

As listas ligadas são um conjunto de objetos ou valores, como em um array, mas todos são conectados na memória, assim, é possível obter valores próximos a outro determinado, ou colocar valores no início ou no fim da lista, também é possível obter o maior 'menor valor' ou o menor 'maior valor' tendo como referência um valor da lista, pois todos eles estão ligados.

Pilhas (Stack)

As pilhas seguem o padrão LIFO (Last In, First Out) onde o último objeto ou valor a entrar na pilha será o primeiro a sair. Elas possuem métodos específicos como colocar um item no topo da pilha, recuperar o último elemento, ou o topo da pilha, entre outros.

Filas (Queue)

As filas seguem o padrão FIFO (First In, First Out) onde o primeiro objeto ou valor a entrar na lista será o primeiro a sair. As filas possuem métodos adicionais de inserção, extração e inspeção de duas formas diferentes, onde uma lança exceção e outra um valor booleano.

Mapa (Map)

Em java os mapas são objetos que armazenam um conjunto de dados como Chave e Valor, onde, principalmente a chave é usada para encontrar os valores, tanto chave como valor podem possuir o tipo distintos ou iguais.

Exemplo:

```
Map<Integer, String> person = new HashMap<Integer, String>();  
person.put(01512664874, Pedro Kawan Silva dos Santos);
```

No exemplo, imagine que o valor do tipo 'Integer' seja o CPF (É uma boa prática usar números de CPF, RG, CNPJ, onde não se faça calculos com seus valores, como valores 'String'), e o tipo String o nome de uma determinada pessoa registrada no mapa, é possível encontrá-la dentro do Map (Atual banco de dados) através do número do CPF com o método:

person.get(Parameter key) -> retorna o valor registrado com a chave do parametro definido no método.

person.containsKey(Parameter key) -> retorna true ou false caso a 'key' passada exista dentro do Map.

person.containsValue(Parameter value) -> retorna true ou false caso o 'value' exista dentro do Map.