

06-05-2024 - Laboratório AEDS2 { Tipos Abstratos de Dados Básicos Nativos do Java }

Interface em Java

- Define e padroniza como classes distintas interagem entre si. Por exemplo, os cálculos de pagamentos nas classes Funcionario e Mercadoria
- Não especifica qualquer detalhe de implementação

Exemplo de Interface: Comparable

```
public class Personagem implements Comparable{
    public int compareTo(Personagem outro){
        //Algoritmo de comparação
    }
}
```

Vector, ArrayList e List

```
Vector<String> ve = new Vector<string>();
ArrayList<String> al = new ArrayList<String>();
LinkedList<string> ll = new LinkedList<String>();
```

Acessar todos os valores

```
for(Iterator i = ve.iterator(); i.hasNext();) System.out.println((String)i.next());
```

ArrayList vs Linkedlist

- ArrayList/Vector são arrays redimensionáveis e têm comportamento similar
 - Implementação sequencial
 - `get(index)` é eficiente
 - Inserir/remover elementos no meio é ineficiente, pois movimentam elementos
 - Eficiente para caminhar entre elementos
- LinkedList são listas duplamente encadeadas
 - Implementação flexível
 - `get(index)` não eficiente
 - Inserir/remover elementos no meio é eficiente
 - Ineficiente para caminhar entre elementos

Métodos da classe Stack

- Elemento `pop()`: desempilha o topo da pilha
- `void push(E elemento)`: empilha um elemento
- `boolean empty()`: retorna se a pilha está vazia
- Elemento `peek()`: retorna o topo da pilha, contudo sem removê-lo
- `int search(Object o)` retorna a posição de um elemento da pilha

Métodos da interface Queue

- `boolean add(E e)`: insere o elemento se existir espaço, retornando true em caso de sucesso.
- `boolean offer(E e)`: insere o elemento e retorna true se existir espaço
- `E element()`: recupera a cabeça da fila, sem removê-lo
- `E peek()`: recupera a cabeça, sem removê-lo. Se a fila vazia, retorna null
- `E poll()`: recupera e remove a cabeça. Se a fila vazia, retorne null

- E `remove()`: recupera e remove a cabeça