

Aula 2

Estrutura de dados Lineares - Prática

Prof. Wesley Gonzaga Alves

ATAL



Ponto de partida

1. Estrutura de Dados Lineares

Listas

Filas

Pilhas

Deque

2. Implementações

Sequenciais

Encadeadas



Nosso objetivo

Realizar implementação de uma estrutura de dados do tipo
Sequencial

Listas

Filas

Pilhas

Deque



Listas

`add(E e)`: Adiciona um elemento ao final da lista.

`add(int index, E e)`: Adiciona um elemento em uma posição específica.

`get(int index)`: Retorna o elemento na posição especificada.

`remove(int index)`: Remove o elemento na posição especificada.

`remove(Object o)`: Remove a primeira ocorrência de um elemento.

`set(int index, E e)`: Substitui o elemento na posição especificada.

`size()`: Retorna o número de elementos na lista.

`isEmpty()`: Verifica se a lista está vazia.

`contains(Object o)`: Verifica se a lista contém o elemento especificado.

`clear()`: Remove todos os elementos da lista.

`indexOf(Object o)`: Retorna o índice da primeira ocorrência de um elemento.

Fila (Queue)

offer(E e): Insere um elemento na fila (retorna false se a fila estiver cheia, em implementações com capacidade limitada).

poll(): Remove e retorna o primeiro elemento da fila, ou null se a fila estiver vazia.

peek(): Retorna o primeiro elemento da fila, ou null se a fila estiver vazia, sem removê-lo.

add(E e): Similar ao offer(), mas lança uma exceção se a fila estiver cheia.

remove(): Remove e retorna o primeiro elemento da fila, lançando uma exceção se a fila estiver vazia.

Pilha (Stack)

`push(E e)`: Adiciona um elemento ao topo da pilha.

`pop()`: Remove e retorna o elemento do topo da pilha.

`peek()`: Retorna o elemento do topo da pilha sem removê-lo.

`isEmpty()`: Verifica se a pilha está vazia.

`search(Object o)`: Retorna a posição de um elemento na pilha, ou -1 se não for encontrado.

Deque (Deque)

addFirst(E e): Adiciona um elemento no início do deque.

addLast(E e): Adiciona um elemento no final do deque.

offerFirst(E e): Tenta adicionar um elemento no início do deque.

offerLast(E e): Tenta adicionar um elemento no final do deque.

removeFirst(): Remove e retorna o primeiro elemento do deque.

removeLast(): Remove e retorna o último elemento do deque.

pollFirst(): Remove e retorna o primeiro elemento, ou null se o deque estiver vazio.

pollLast(): Remove e retorna o último elemento, ou null se o deque estiver vazio.

getFirst(): Retorna o primeiro elemento sem removê-lo.

getLast(): Retorna o último elemento sem removê-lo.

peekFirst(): Retorna o primeiro elemento sem removê-lo, ou null se o deque estiver vazio.

peekLast(): Retorna o último elemento sem removê-lo, ou null se o deque estiver vazio.

Seu objetivo p/ próxima aula - **Valendo 0,5**

Realizar implementação de uma estrutura de dados do tipo sequencial

Listas

Filas

Pilhas

Deque



Atividade 1 - Valendo 0,5

Você está desenvolvendo um sistema de gerenciamento de pedidos de uma cafeteria, e foi escalado para criar a regra de negócio abaixo

Os pedidos chegam em uma sequência contínua e devem ser processados em ordem de chegada, mas há três situações especiais a serem consideradas:

- Clientes VIP devem ter seus pedidos processados com prioridade máxima.
- Clientes que cancelam um pedido devem ser removidos da lista de pedidos pendentes.
- Em determinados momentos, o gerente pode querer visualizar os pedidos em espera sem removê-los do sistema.

Atividade 1 - Valendo 0,5

Funcionalidades - *Features*

- Adicionar um novo pedido.
- Processar o próximo pedido.
- Visualizar os pedidos na fila sem removê-los.
- Remover um pedido cancelado.
- Atualizar a prioridade do pedido na fila.

Atividade 1 - Valendo 0,5

- a) Qual estrutura de dados você utilizaria para gerenciar os pedidos da cafeteria, considerando que os pedidos devem ser processados de forma eficiente e respeitando as prioridades dos clientes VIP? Explique a escolha.

- b) Qual operação específica você usaria para cada uma das features no contexto da estrutura de dados escolhida?

- c) Explique como o sistema de pedidos deve lidar com a entrada de um cliente VIP. A estrutura de dados escolhida suporta essa funcionalidade? Se sim, explique como. Se não, sugira uma alteração ou solução alternativa.

Próxima aula

Collections Framework