

Introdução

As estruturas de dados são elementos fundamentais para a organização e armazenamento de dados em programas de computador. Dentre essas estruturas, destaca-se a fila, uma estrutura linear que segue o princípio FIFO (First In, First Out), onde o primeiro elemento a entrar é o primeiro a sair. Nesta apostila, discutiremos a implementação de uma fila em JavaScript usando arrays, bem como as operações básicas que podem ser realizadas.

Desenvolvimento

Em JavaScript, é possível implementar uma fila utilizando um array. Para isso, podemos utilizar dois métodos nativos do array: push e shift.

Para criar uma fila vazia, basta criar um array vazio:

```
1 let queue = [];
```

A operação de enfileiramento (inserção de elementos na fila) é realizada através do método push. Esse método adiciona um ou mais elementos ao final do array. No caso da fila, adicionamos um elemento ao final do array para representar a adição de um elemento na fila.

```
1 function enqueue(element) {  
2   queue.push(element);  
3 }
```

A operação de desenfileiramento (remoção de elementos da fila) é realizada através do método shift. Esse método remove o primeiro elemento do array e retorna o valor removido. Na fila, é importante remover o primeiro elemento, por isso utilizamos o método shift.

```
1 function dequeue() {  
2   return queue.shift();  
3 }
```

Para verificar o tamanho da fila, podemos utilizar a propriedade length do array:

```
1 function size() {  
2   return queue.length;  
3 }
```

Para verificar se a fila está vazia, basta verificar se o tamanho é igual a zero:

```
1 function isEmpty() {  
2   return queue.length === 0;  
3 }
```

Para mostrar o primeiro elemento da fila em JavaScript, basta utilizar a propriedade `queue[0]`, que retorna o primeiro elemento do array, que representa o primeiro elemento da fila. Caso a fila esteja vazia, basta retornar `null` ou um valor indicativo de que a fila está vazia.

Segue abaixo o código-fonte da função `front()`, que retorna o primeiro elemento da fila:

```
1 function front() {  
2   if (queue.length === 0) {  
3     return null; // ou um valor indicativo de que a fila está vazia  
4   }  
5   return queue[0];  
6 }
```

Exemplo completo:

```
1 let queue = [];  
2  
3 function enqueue(element) {  
4   queue.push(element);  
5 }  
6  
7 function dequeue() {  
8   return queue.shift();  
9 }  
10  
11 function size() {  
12   return queue.length;  
13 }  
14  
15 function isEmpty() {  
16   return queue.length === 0;  
17 }  
18  
19 enqueue(1);  
20 enqueue(2);  
21 enqueue(3);  
22 console.log(queue); // [1, 2, 3]  
23 console.log(dequeue()); // 1  
24 console.log(queue); // [2, 3]  
25 console.log(size()); // 2  
26 console.log(isEmpty()); // false
```

Conclusão:

Em resumo, a fila é uma estrutura de dados linear que segue o princípio FIFO, onde o primeiro elemento a entrar é o primeiro a sair. É possível implementar uma fila em JavaScript utilizando arrays e dois métodos nativos: `push` e `shift`. Essa estrutura é muito útil em diversas aplicações, como gerenciamento de tarefas em sistemas operacionais, processamento de eventos em tempo real e muito mais. É importante ressaltar que as operações de inserção e remoção na fila possuem complexidade $O(1)$, o que significa que elas são muito eficientes.