Filas e Pilhas em JS

Fila (Queue) em JavaScript:

Uma fila é uma estrutura de dados que segue o princípio "Primeiro a entrar, primeiro a sair" (FIFO - First In, First Out). Em JavaScript, você pode implementar uma fila usando arrays. Aqui está um exemplo básico:

```
// Implementação de uma Fila em JavaScript
class Fila {
  constructor() {
    this.items = [];
  }

  // Adiciona um elemento ao final da fila
  enfileirar(element) {
    this.items.push(element);
  }

  // Remove o elemento no início da fila e retorna-o
  desenfileirar() {
    if (this.isEmpty()) {
      return "A fila está vazia";
    }
    return this.items.shift();
  }

  // Retorna o elemento no início da fila sem removê-lo
  frente() {
    if (this.isEmpty()) {
      return "A fila está vazia";
    }
    return this.items[0];
  }

  // Verifica se a fila está vazia
  isEmpty() {
    return this.items.length === 0;
  }

  // Retorna o tamanho da fila
  tamanho() {
    return this.items.length;
  }
}

// Exemplo de uso da Fila
let fila = new Fila();
fila.enfileirar(1);
fila.enfileirar(2);
fila.enfileirar(3);
```

```
console.log(fila.desenfileirar()); // Saída: 1
console.log(fila.frente()); // Saída: 2
console.log(fila.tamanho()); // Saída: 2
```

Pilha (Stack) em JavaScript:

Uma pilha é uma estrutura de dados que segue o princípio "Último a entrar, primeiro a sair" (LIFO - Last In, First Out). Aqui está um exemplo de implementação de pilha em JavaScript:

```
// Implementação de uma Pilha em JavaScript
class Pilha {
  constructor() {
    this.items = [];
  }

  // Adiciona um elemento ao topo da pilha
  empilhar(element) {
    this.items.push(element);
  }

  // Remove o elemento do topo da pilha e retorna-o
  desempilhar() {
    if (this.isEmpty()) {
      return "A pilha está vazia";
    }
    return this.items.pop();
  }

  // Retorna o elemento do topo da pilha sem removê-lo
  topo() {
    if (this.isEmpty()) {
      return "A pilha está vazia";
    }
    return this.items[this.items.length - 1];
  }

  // Verifica se a pilha está vazia
  isEmpty() {
    return this.items.length === 0;
  }

  // Retorna o tamanho da pilha
  tamanho() {
    return this.items.length;
  }
}

// Exemplo de uso da Pilha
let pilha = new Pilha();
pilha.empilhar(1);
```

```
pilha.empilhar(2);
pilha.empilhar(3);

console.log(pilha.desempilhar()); // Saída: 3
console.log(pilha.topo()); // Saída: 2
console.log(pilha.tamanho()); // Saída: 2
```

Questão 1 (Implemente uma Fila):

Você está liderando uma equipe de desenvolvimento em uma empresa de software. Para otimizar a eficiência na resolução de problemas de clientes, você decidiu implementar um sistema de gerenciamento de tickets. Cada vez que um cliente relata um problema, um ticket é gerado e colocado em uma fila de atendimento. Desenvolva uma aplicação em JavaScript que utilize uma fila para simular esse sistema.

Desafios 1:

1. Implemente a funcionalidade de adicionar novos tickets à fila quando um cliente relatar um problema.
2. Crie uma função para atender o próximo ticket na fila, removendo-o após o atendimento.
3. Permita a visualização do próximo ticket a ser atendido sem removê-lo da fila.
4. Implemente uma contagem de quantos tickets estão atualmente na fila.

Questão 2 - Implemente uma Pilha:

Sua equipe está trabalhando em um novo recurso para o software da empresa de desenvolvimento. Para garantir a robustez do sistema, você decidiu implementar um mecanismo de histórico de ações. Desenvolva uma aplicação em JavaScript que utilize uma pilha para armazenar as ações dos desenvolvedores durante o desenvolvimento do software.

Desafios 2:

1. Crie uma função para empilhar cada ação realizada pelos desenvolvedores durante o trabalho no software.
2. Implemente a capacidade de desempilhar a última ação, revertendo-a.
3. Garanta que as ações desempilhadas possam ser refeitas, ou seja, empilhadas novamente.

4. Permita a verificação de quantas ações estão atualmente na pilha de histórico.