Centro Paula Souza

Fatec Jahu

IMPLEMENTAÇÃO DE ARRAYS SEM FUNÇÕES NATIVAS

ESTRUTURA DE DADOS

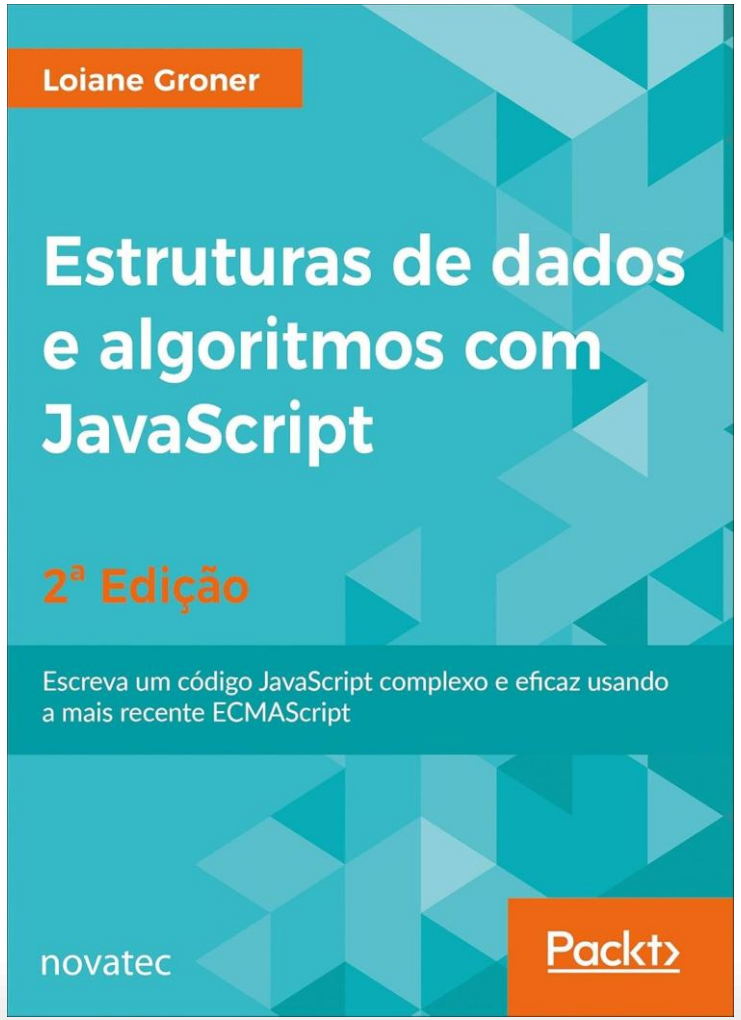
CST em Desenvolvimento de Software Multiplataforma

PROF. Me. TIAGO A. SILVA

LIVRO DE REFERÊNCIA DA DISCIPLINA

- **BIBLIOGRAFIA BÁSICA:**
  - GRONER, Loiane. **Estrutura de dados e algoritmos com JavaScript**: escreva um código JavaScript complexo e eficaz usando a mais recente ECMASript. **São Paulo: Novatec Editora**, 2019.

- **NESTA AULA:**
  - **Capítulo 3**



PARA SOBREVIVER AO JAVASCRIPT

Non-zero value

0

null

undefined

ARRAYS

Exemplo de Implementação em JavaScript,
sem o uso de funções nativas

RELEMBRANDO VETORES...

- Vetores no Visualg (***também chamados de arrays ou arranjos***) são estruturas de dados que permitem armazenar vários valores do mesmo tipo em uma única variável, **acessados por um índice numérico**.

- **A chave parta entender vetores está no entendimento de que um valor guardado em vetor está em um determinado índice (posição)**

RELEMBRANDO VETORES...

```
1  Algoritmo "Visualg_Para_Js"
2
3  Var
4     nomes : vetor[1..3] de caracter
5     i : inteiro
6
7  Inicio
8
9     nomes[1] <- "João"
10    nomes[2] <- "Maria"
11    nomes[3] <- "José"
12
13    Para i de 1 até 3 Faça
14       Escreval(nomes[i])
15    FimPara
16
17 Fimalgoritmo
```

```
1  // Declaração do vetor
2  let nomes = [];
3  nomes[0] = "João";
4  nomes[1] = "Maria";
5  nomes[2] = "José";
6  // Acessando cada posição do vetor
7  for (let i = 0; i < 3; i++) {
8      console.log(nomes[i]);
9  }
```

O QUE SÃO ARRAYS?

- Um array é uma estrutura de dados que armazena uma coleção de elementos (geralmente do mesmo tipo) em **posições consecutivas de memória**.

- Características principais:
  - Os elementos são acessados por índices (geralmente começando em 0).
  - **O acesso é rápido**: buscar ou alterar um elemento pelo índice é operação de tempo constante **O(1).**
  - O tamanho costuma ser fixo: ao criar um array, você define quantos elementos ele terá (embora em algumas linguagens modernas, isso seja flexível).
  - Todos os elementos ocupam o mesmo espaço de memória (porque têm o mesmo tipo).

ARRAYS

Exemplo de Implementação em JavaScript,
sem o uso de funções nativas

IMPLEMENTAÇÃO DE UM ARRAY

- Essa implementação manual simula o comportamento de um array básico **sem recorrer** às funções nativas do JavaScript, como **`push()`**, **`pop()`**, ou **`length`**, e usa a lógica de manipulação de índices para armazenar e acessar os dados.
  - Nas futuras implementações de estrutura de dados poderemos usar as funções nativas.
  - Nosso objetivo em estudar sem o uso delas neste momento é entender seu funcionamento.

IMPLEMENTAÇÃO DE UM ARRAY

- Estrutura interna (**`items`**): Usamos um objeto simples para armazenar os itens do array, onde as chaves são os índices do array.
  - **`adicionar(elemento)`**: Adiciona um novo elemento ao final do array, usando o índice correspondente ao valor do tamanho atual (**`this.tamanho`**). Depois, incrementamos o tamanho.
  - **`remover()`**: Remove o último elemento do array, usando o índice que corresponde a tamanho - 1, e decrementa o tamanho.
  - **`obterElemento(indice)`**: Acessa e retorna o elemento no índice especificado, se ele for válido (entre 0 e tamanho - 1).
  - **`tamanhoArray()`**: Retorna o número de elementos no array.
  - **`limpar()`**: Remove todos os elementos do array, redefinindo o objeto **`items`** e o tamanho para zero.

IMPLEMENTAÇÃO DO PROTÓTIPO DA CLASSE

```
1   class MeuArray {
2
3       constructor() { }
4
5       adicionar(elemento) {  }
6
7       remover() { }
8
9       obterElemento(indice) { }
10
11      tamanhoArray() { }
12
13      limpar() { }
14  };
15
16  module.exports = MeuArray;
```

IMPLEMENTAÇÃO DO MÉTODO CONSTRUTOR

```
1   class MeuArray {
2
3       // Chamando quando um objeto é criado
4       constructor()
5       {
6           // Usamos uma lista para
7           // armazenar os itens do array
8           this.items = [];
9
10          // Mantemos o controle do
11          // tamanho do array
12          this.tamanho = 0;
13      }
```

IMPLEMENTAÇÃO DO MÉTODO ADICIONAR

```
15      // Adiciona um elemento ao final do array
16      adicionar(elemento)
17      {
18          // Insere o elemento na posição
19          // atual do tamanho
20          this.items[this.tamanho] = elemento;
21
22          // Incrementa o tamanho
23          this.tamanho++;
24      }
```

IMPLEMENTAÇÃO DO MÉTODO REMOVER

```
26      // Remove o último elemento do array
27      remover()
28      {
29          // Se o array estiver vazio, não há o que remover
30          if (this.tamanho === 0) {
31              return undefined;
32          }
33
34          // Armazena o último item
35          const ultimoItem = this.items[this.tamanho - 1];
36
37          // Remove o último item do array
38          delete this.items[this.tamanho - 1];
39
40          // Decrementa o tamanho
41          this.tamanho--;
42
43          return ultimoItem;  // Retorna o item removido
44      }
```

IMPLEMENTAÇÃO DO MÉTODO OBTERELEMENTO

```
46      // Acessa o elemento em um índice específico
47      obterElemento(indice)
48      {
49          if (indice < 0 || indice >= this.tamanho) {
50
51              // Se o índice estiver fora do alcance,
52              // retorna undefined
53              return undefined;
54          }
55
56          // Retorna o item no índice solicitado
57          return this.items[indice];
58      }
```

IMPLEMENTAÇÃO DO MÉTODO TAMANHOARRAY

```
60      // Retorna o tamanho do array
61      tamanhoArray()
62      {
63          // Retorna o valor do
64          // tamanho atual do array
65          return this.tamanho;
66      }
```

IMPLEMENTAÇÃO DO MÉTODO LIMPAR

```
68      // Remove todos os elementos do array
69      limpar()
70      {
71          // Limpa o objeto
72          this.items = [];
73
74          // Reinicializa o tamanho
75          this.tamanho = 0;
76      }
```

COMO USAR A CLASSE

```
const MeuArray = require("./MeuArray.js");

// Exemplo de uso
let minha_variavel = new MeuArray();

minha_variavel.adicionar(10);
console.table(minha_variavel.items);

minha_variavel.adicionar(20);
console.table(minha_variavel.items);

minha_variavel.adicionar(30);
console.table(minha_variavel.items);

console.log(minha_variavel.obterElemento(1));  // Saída: 20
console.log(minha_variavel.tamanhoArray());  // Saída: 3

// Saída: 30 (Remove o último elemento)
console.log(minha_variavel.remover());

console.log(minha_variavel.tamanhoArray());  // Saída: 2
```

EXERCÍCIOS

Use a classe implementada acima para resolver os exercícios

EXERCÍCIO 1

- Uma empresa deseja criar um sistema simples para gerenciar as tarefas da equipe. Cada tarefa será armazenada em um array utilizando a classe MeuArray.

- **Requisitos**:
  - Criar uma instância da classe MeuArray.
  - Adicionar cinco tarefas ao array.
  - Remover a última tarefa adicionada.
  - Exibir todas as tarefas armazenadas.

- **Perguntas**:
  - O que acontece quando tentamos acessar um índice fora do tamanho do array?
  - Como garantir que a ordem das tarefas seja mantida ao adicionar e remover itens?

EXERCÍCIO 2

- O setor de Recursos Humanos de uma empresa deseja armazenar os nomes dos funcionários que participaram de um treinamento.

- **Requisitos**:
  - Criar uma instância da classe MeuArray.
  - Adicionar os nomes de quatro funcionários ao array.
  - Obter o nome do terceiro funcionário que participou.
  - Limpar todos os registros do array.

- **Perguntas**:
  - O que acontece se tentarmos acessar um índice inexistente após limpar o array?
  - Como modificar a classe para garantir que os nomes armazenados sejam únicos?

TRABALHO SALVA-VIDAS

Orientações de como realizar o trabalho dessa semana

TRABALHO SALVA-VIDAS DESSA AULA

- Instruções:
  - Escolha um dos exercícios para gravar.
  - Orientações gerais de como enviar: veja o tópico “Trabalho Salva-vidas da Aula 1.
    - Não se esqueça de marcar os perfis e das hashtags.
  - Duração do Vídeo: aproximadamente **90 segundos**.
  - Adicione legendas
  - **Prazo: 04/09/2025**

OBRIGADO!

- Encontre este **material on-line** em:
  - Slides: Plataforma Microsoft Teams

- Em caso de **dúvidas**, entre em contato:
  - **Prof. Tiago**: tiago.silva238@fatec.sp.gov.br