.StartSe
<tech/academy>

Programação Orientada a Objetos em JavaScript

Introdução

Programação Orientada a Objetos (POO)

Um estilo ou modo de programar

utilizando **objetos**

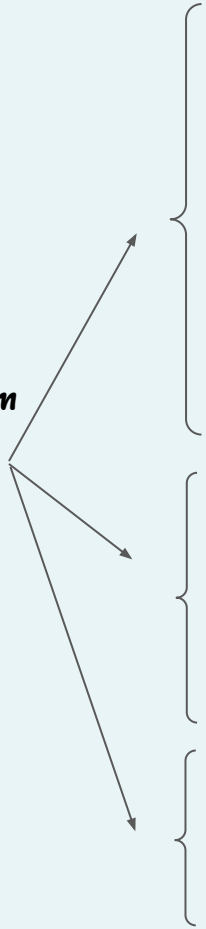
Programação Estruturada

Um estilo ou modo de programar

utilizando uma estrutura baseada em

módulos ou funções

Código de programação em JavaScript estruturado em **módulos ou funções**



```
function adicionaTarefa(textoDaTarefa) {
    const elementoLI = document.createElement('li');
    const elementoSPAN = document.createElement('span');
    elementoSPAN.setAttribute('id','tarefa');
    elementoSPAN.textContent = textoDaTarefa;
    elementoLI.className = 'naoRealizada';
    elementoLI.appendChild(elementoSPAN);
    elementoLI.appendChild(adicionaBotaoRemover());
    elementoSPAN.addEventListener('click',function() {
        if(this.id === 'tarefa') {
            if(this.parentNode.className === 'naoRealizada') {
                this.parentNode.className = 'realizada'
            } else {
                this.parentNode.className = 'naoRealizada'
            }
        }
    });
    return elementoLI;
}

function adicionaBotaoRemover() {
    const botaoRemover = document.createElement('button');
    botaoRemover.textContent = 'X';
    botaoRemover.className = 'remover';
    botaoRemover.addEventListener('click', function() {
            listaTarefas.removeChild(this.parentNode);
            exibeOcultaListaSuspensa();
        }
    );
    return botaoRemover;
}

function exibeOcultaListaSuspensa(){
    const elementoSPAN = document.querySelector('#tarefa');
    if(elementoSPAN === null) {
        listaSuspensa.setAttribute('hidden','hidden');
    } else {
        listaSuspensa.removeAttribute('hidden');
    }
}
```

.StartSe
<tech/academy>

Códigos de programação em JavaScript utilizando **classes** (Produto e Livro) e **objetos** (criados a partir dessas classes) **prod** e **liv**

```
class Produto {
    #tipo;
    constructor(tipo) {
        this.#tipo = tipo;
    }
    getDados() {
        return this.#tipo;
    }
}
```

```
const prod = new Produto('Generico');
console.log(prod.getDados());
const liv = new Livro('Livro','POO',50);
console.log(liv.getDados());
```

```
class Livro extends Produto {
    #titulo;
    #numPag;
    constructor(tipo,titulo,numPag) {
        super(tipo);
        this.#titulo = titulo;
        this.#numPag = numPag;
    }
    getDados() {
        return(`
        Tipo: ${super.getDados()}
        Titulo: ${this.#titulo}
        Num.Pág.: ${this.numPag}
        `);
    }
}
```

.StartSe
<tech/academy>

Programação Estruturada vs Programação Orientada a Objetos

Programação Estruturada (PE)	Programação Orientada a Objetos (POO)
Código dividido em **módulos** ou **funções**	Código baseado no conceito de **objetos**
Menos reusabilidade de código do que POO	Mais reusabilidade de código do que PE
Menos facilidade de modificar o código do que POO	Maior facilidade de modificar o código do que PE

Programação Orientada a Objetos (POO)

- Os **objetos** são definidos através um modelo ou template chamado **classe**
- As **classes** possuem características comuns a objetos do mesmo tipo
    - pessoa, carro, produto, etc
- As **classes** possuem comportamentos comuns a objetos do mesmo tipo
    - mover, parar, atualizar, comunicar, etc

Programação Orientada a Objetos (POO)

- As características das classes são chamadas de **atributos**
- Os comportamentos das classes são definidos pelos **métodos**
    - os métodos são funções declaradas dentro de uma classe
    - os métodos permitem isolar ou esconder os atributos de um objeto para que eles sejam acessados somente através dos métodos
        - assim os dados e os métodos de uma classe ficam encapsulados dentro de uma mesma entidade: o **objeto**

Programação Orientada a Objetos (POO)

- A partir da declaração de uma classe é possível criar **múltiplos** objetos
    - cada objeto representa uma **instância** de uma classe, por exemplo:
        - para uma classe estudante, podemos criar múltiplos objetos do tipo estudante
            - cada objeto representa um estudante em particular com seus próprios dados ou atributos (nome, endereço, telefone, etc)
        - a criação de um objeto é chamada de **instanciação**

Programação Orientada a Objetos (POO)

- A modularização de um código através de classes, permite reuso de código, uma vez que:
    - pode-se utilizar métodos de objetos de diferentes classes para produzir algum resultado sem precisar reescrevê-los

Programação Orientada a Objetos (POO)

- A modularização de um código através de classes, permite uma facilidade para modificar o código, uma vez que:
  - pode-se alterar parte do código de uma classe sem precisar alterar o código de outras classes
    - pode-se alterar a declaração de atributos e métodos privados (incluir, renomear, etc)
    - pode-se alterar o código de métodos públicos