UNINOVE

Programação Orientada a Objetos

INTRODUÇÃO A ORIENTAÇÃO A OBJETOS EM JAVASCRIPT

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR

Orientação a Objetos em JavaScript

Neste tópico iremos abordar o tema da programação orientada a objetos (POO) em JavaScript. Apesar de ser um tema extenso e amplos, vamos tratar dos principais assuntos de forma simples e bem direta.

A Orientação a objetos teve sua criação inspirada no mundo real, nas coisas que nos cercam. Diferentemente do estilo da programação estruturada, a orientação a objetos procura agrupar, a exemplo do que fazemos na vida real, unir estas coisas semelhantes no que chamamos de classe. Nesta classe, tentamos definir e representas a ideia de alguma coisa, real ou abstrata, porém ainda no âmbito conceitual. Podemos reunir informações a seu respeito, como suas características e ações.

Neste universo, diversas classes interagem entre si, e se "comunicam" por meio de mensagens, trocando informação. Entre estas informações, as suas características podem ser apresentadas ou transformadas, e a partir deste ponto vamos chamar as características desta classe de atributos. Assim os atributos são algo como as definições de uma classe. Junta-se a eles, suas ações, que podemos definir como comportamentos das classes, coisas que uma classe pode fazer, e vamos chamá-los de métodos.

No entanto, uma classe é uma ideia, um conceito, um modelo de alguma coisa. Para que possamos sair deste modelo e criar algo de fato devemos representa-los coisas de fato, que iremos chamar de objetos. Assim, o objeto seria a parte concreta ou "real" de uma classe, desta ideia ou conceito inicial.

Vamos exemplificar. Imagine a ideia de um carro, podemos pensar em vários aspectos de um carro, mas com certeza antes de criar um devemos projeta-lo, desenha-lo, e somente depois de tudo bem adequado poderíamos criar um carro real.

Na programação orientada a objetos é da mesma forma. O nosso carro, essa ideia ou conceito de carro iremos chamar de classe carro. Nesta classe iremos definir como será este carro, por exemplo, a sua cor, quantas portas ele terá, qual o modelo, seu ano de fabricação, se terá direção hidráulica, câmbio automático, ar-condicionado, enfim, iremos definir quais são as características deste carro, ou quais são os seus atributos. Uma vez definido como ele será (atributos) devemos definir o que ele fará, quais ações, por exemplo este carro deverá acelerar, frear, virar para direita, virar para a esquerda, ligar, desligar, em resumo, definir o que ele faz, quais suas ações, ou melhor, quais são os seus métodos.

Uma vez que esta ideia de carro esteja completa, ou seja, a sua classe com atributos e métodos, podemos iniciar a criação ou montagem dos carros, agora sim podemos “fabricar” nossos automóveis. Neste universo da programação orientada a objetos, chamamos o resultado desta fabricação, cada carro criado de objeto, ou seja, um carro fabricado ou criado é um objeto, e o processo de fabricação ou criação, ou seja, o ato de criar um carro chamamos de instância um objeto.

Classe

Uma classe é a definição ou conceito de alguma coisa, ou de objetos. Neste projeto de objetos podemos definir características e ações correspondentes, ou atributos e métodos.
Uma classe serve de modelo ou molde para a futura criação de objetos, ou seja, ele é utilizado para instanciar futuros objetos.
Exemplo:

```
1. // Definição das Classes
2.          function Pessoa(a, b){
3.                 this.nome = a;
4.                 this.idade = b;
5.           }
```

Atributos

Os atributos são característica ou propriedades de um objeto, definidos em uma classe. Estes atributos visam definir as características, a forma de um objeto.
Exemplo:

```
1.   this.nome = a;
2.   this.idade = b;
```

Métodos

Os métodos são as ações ou funções possíveis de um objeto, definidos em uma classe.
Exemplo:

```
1.  // Método exibir
2.    Pessoa.prototype.exibirPessoa = function(){
3.         return 'Nome = ' + this.getNome() + ', Idade = ' + this.getIdade(
4.    }
```

Objeto

É um exemplar, o representante real de uma classe, a instância de uma classe.

Exemplo:

```
1. var pessoa1 = new Aluno('Joao da Silva', 30, '123456', 'Ciência da Computa
```

Construtor

O construtor é o método solicitado toda vez que um objeto é criado.

Exemplo:

```
1. function Funcionario(a, b, c, d){
2.         Pessoa.call(this, a, b);
3.         this.matricula = c;
4.         this.setor = d;
5. }
```

Exemplo

Vamos analisar o programa a seguir, com uma simples implementação em JavaScript, onde utiliza-se a orientação a objetos.

```
<!DOCTYPE html>
<html>
<head>
    <title>Tópico 19</title>
    <meta charset="UTF-8">
</head>
<body>
    <script>
        // Classe
        function Carro(a, b, c, d){
            this.cor = a;
            this.ano = b;
            this.portas = c;
            this.modelo = d;
            this.velocidadeAtual = 0;
            this.ligado = false;
        }

        // Método
        Carro.prototype.passear = function(){
            this.ligado = true;
            this.velocidadeAtual = 50;
            return this.velocidadeAtual;
        }

        // Método
        Carro.prototype.parar = function(){
            this.velocidadeAtual = 0;
        }

        // Método
        Carro.prototype.estacionar = function(){
            this.parar();
            this.ligado = false;
        }

        // Método
        Carro.prototype.acelerar = function(){
            this.velocidadeAtual += 10;
        }
```

```
41.
42.         // Método
43.         Carro.prototype.informacao = function(){
44.             if(this.ligado)
45.                 var motorLigado = 'ligado';
46.             else
47.                 var motorLigado = 'desligado';
48.
49.             return 'Velocidade atual = ' + this.velocidadeAtual + ' km/h,
50.             ' o meu ' + this.modelo + ' ' + this.cor + ' está ' + motorLig
51.
52.         }
53.
54.         // Instanciar objeto
55.         var veiculo1 = new Carro('preto', 2017, 5, 'SUV');
56.
57.         // Exibir
58.         console.log(veiculo1.informacao());
59.         veiculo1.passear();
60.
61.         console.log(veiculo1.informacao());
62.         veiculo1.parar();
63.         console.log(veiculo1.informacao());
64.         veiculo1.passear();
65.         console.log(veiculo1.informacao());
66.         veiculo1.estacionar();
67.         console.log(veiculo1.informacao());
68.         veiculo1.passear();
69.         veiculo1.acelerar();
70.         veiculo1.acelerar();
71.         console.log(veiculo1.informacao());
72.
73.     </script>
74. </body>
75. </html>
```

```
Velocidade atual = 0 km/h, o meu SUV preto está desligado.          aula19_a.html:58
Velocidade atual = 50 km/h, o meu SUV preto está ligado.           aula19_a.html:61
Velocidade atual = 0 km/h, o meu SUV preto está ligado.            aula19_a.html:63
Velocidade atual = 50 km/h, o meu SUV preto está ligado.           aula19_a.html:65
Velocidade atual = 0 km/h, o meu SUV preto está desligado.         aula19_a.html:67
Velocidade atual = 70 km/h, o meu SUV preto está ligado.           aula19_a.html:71
> |
```

Na imagem acima, temos o resultado do exemplo anterior.

A explicação detalhada deste código exemplo está apresentada nos vídeos a seguir.
Assista até o final e veja a implementação comentada.

> ## DICA:
>
> Para saber mais sobre orientação a objetos veja em:
> https://developer.mozilla.org/pt-PT/docs/Javascript_orientado_a_objetos
> (https://developer.mozilla.org/pt-PT/docs/Javascript_orientado_a_objetos)

Conclusão

A orientação a objetos é um paradigma de programação poderoso e ágil. Como utilizado em diversas linguagens modernas, o JavaScript não poderia ficar de fora.
Os novos conceitos e estrutura de programação são fáceis de serem implementados, no entanto requer atenção e cuidado.

ATIVIDADE

Em orientação a objetos, devemos instanciar os objetos antes de utilizar seus métodos e atributos.

Escolha a opção correta de instanciar um objeto:

A. int pessoa1 = new Aluno('Joao da Silva', 30, '123456', 'Ciência da Computação');

B. var pessoa1 = Aluno('Joao da Silva', 30, '123456', 'Ciência da Computação');

C. var pessoa1 = new Aluno('Joao da Silva', 30, '123456', 'Ciência da Computação');

D. String pessoa1 = Aluno('Joao da Silva', 30, '123456', 'Ciência da Computação');

ATIVIDADE

Segundo o código de exemplo abaixo, selecione a alternativa que utiliza corretamente o método de um objeto.

```
Carro.prototype.parar = function(){
    this.velocidadeAtual = 0;
}
var veiculo1 = new Carro('preto', 2017, 5, 'SUV');
```

A. veiculo1.parar;

B. veiculo1.parar();

C. veiculo1.parar.prototype;

D. parar.veiculo1();

ATIVIDADE

Qual alternativa exibe um método no console do navegador?

A. console.log(veiculo1.informacao());

B. console(veiculo1.informacao());

C. console.log(informacao());

D. log(veiculo1.informacao());

REFERÊNCIA

MORRISON, M. Use a cabeça JavaScript. 5º Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO. C. A. J. Faça um site JavaScript orientado por projeto. 6º ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8º Ed. São Paulo: Novatec, 2010. 245 p.

TERUEL, Evandro Carlos. Programação orientada a objetos com JAVA sem mistérios. 1º ed. São Paulo, 2016: Universidade Nove de Julho - UNINOVE. 386 p.