

Desenvolvimento de Sistemas : **Classes e Objetos**

Prof. Ovídio Francisco

Plano de aula

Introdução

Introdução

O contexto - Programação Orientada a Objetos

A **POO** é um paradigma de programação que define **estruturas de dados** e suas **operações** em **objetos**.

Cada **tipo de dado** é descrito em uma estrutura especial chamada **classe**.

O contexto - Classes

O termo **classe** *frequentemente* se refere a um grupo ou categoria de coisas.

Exemplo: Clientes, Veículos, Arquivos, Personagens e elementos de jogos.

Classes

Definições de Classe

Uma classe é um template ou modelo que determina os atributos e métodos comuns a todos os objetos de certo tipo. ¹

Uma classe representa computacionalmente um grupo ou categoria de objetos do mundo real.
Ela descreve o conteúdo dos objetos que pertencem a esse grupo, bem como suas ações. ²

Uma classe é uma modelo computacional que descreve algo do mundo real.
Essa descrição engloba suas características e ações.

¹ <http://www.iitk.ac.in/esc101/05Aug/tutorial/java/concepts/class.html>

² caml.inria.fr/pub/docs/oreilly-book/html/book-ora140.html

O contexto - Classes

Uma **classe** é um modelo computacional que **descreve algo do mundo real**.

Essa descrição engloba suas **características** e **ações**.

Às **características** chamamos de **atributos** e às **ações** de **métodos**

Atributos

Atributos

Podem ser entendidos como uma descrição dos dados que irá conter.

Por exemplo:

Pessoa altura, peso, CPF, profissão

Bola diâmetro, peso, cor, tipo

Carro número de portas, autonomia, cor

Animal espécie, número de patas, cor

Reunião data, hora, local



Cada indivíduo gato possui como atributos:

- raça
- altura
- peso
- cor

Cada funcionário de uma empresa possui como atributos:

- Nome
- CPF
- Data de nascimento
- Salário
- Número de dependentes
- Data de contratação

Métodos

Em POO precisamos que um objeto realize **ações** ou mesmo se **comunique** com outros objetos

Exemplos de ações de objetos

- Acelerar um carro
- Aumentar a temperatura de um ar condicionado
- Imprimir um documento
- Ataque de um personagem

Exemplos de comunicação entre objetos

- Consultar a idade de uma pessoa
- Consultar quantas folhas há na impressora
- Consultar o saldo de uma conta bancária

Métodos são ações comuns a todos os elementos (objetos) de uma classe.

Exemplo, cada colaborador de uma empresa possui como ações:

- Calcular idade
- Calcular tempo de serviço
- Ajustar salário
- Salvar no banco de dados
- Calcular valor de rescisão

Criando classes com Javascript

Relembrando...

- Cada classe irá representar um algo do mundo real ou um problema que será modelado computacionalmente.
- Define seus atributos (características) e seus métodos(ações).

Classe com Atributos e Métodos

Cada aluno de uma disciplina, em um período possui como atributos:

- Nome
- RA
- Data de nascimento
- Notas

E possui como ações:

- Calcular idade
- Calcular média
- Verificar situação final

Definições de Classe

Há várias formas de criar classes em Javascript, uma das mais formais é:

```
class Pessoa {  
  
    // Atributos da classe Pessoa  
    // Métodos da classe Pessoa  
  
}
```

```
class Bola {  
  
    // Atributos da classe Bola  
    // Métodos da classe Bola  
  
}
```

Instâncias

Definições de Classe

Até agora, vemos uma classe como um **molde**.

E como todo molde, as classes servem para criar coisas.

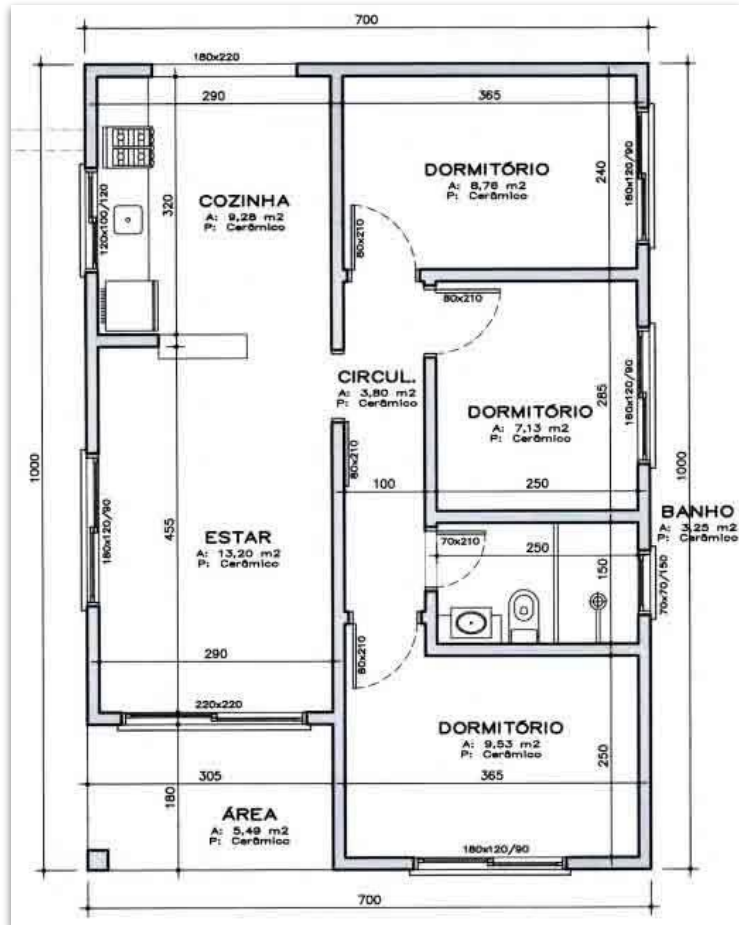
Essas coisas criadas vão seguir o mesmo formato do **molde** utilizado na **criação**.

Chamamos de **objeto** ou **instância** aquilo que criamos a partir de uma classe.

Uma classe pode ser **instanciada várias vezes**, ou seja, podem criar **vários objetos**

Os objetos de uma **mesma classe** compartilham o mesmo **conjunto de atributos**, porém com **valores diferentes**

Classes e Objetos



Classes e Objetos

Para criarmos objetos usamos a palavra new seguida do construtor da classe. (Vermos mais sobre construtores a seguir).

```
class NomeDaClasse {  
    // Atributos e Métodos  
}  
  
var obj = new NomeDaClasse();
```

Classes e Objetos

```
class Pessoa {  
    // Atributos da classe Pessoa  
    // Métodos da classe Pessoa  
}  
  
class Bola {  
    // Atributos da classe Bola  
    // Métodos da classe Bola  
}  
  
var p = new Pessoa();  
var b = new Bola();
```

p é um objeto do tipo pessoa pois foi criado a partir da classe pessoa.
Da mesma forma b é um objeto do tipo Bola.

Criando atributos

Classes e Objetos

Uma forma bem simples de criar atributos e simplesmente atribui valores a um identificador dentro da declaração da classe.

```
class NomeDaClasse {  
    atributo = valor;  
}
```

Classes e Objetos

Uma forma bem simples de criar atributos e simplesmente atribui valores a um identificar dentro da declaração da classe.

```
class Pessoa {  
    nome = "Nicanor";  
    idade = 13;  
}  
  
var p = new Pessoa();  
  
document.write("Nome: " + p.nome + "<br>");  
document.write("Idade: " + p.idade + "<br>");
```

Classes e Objetos

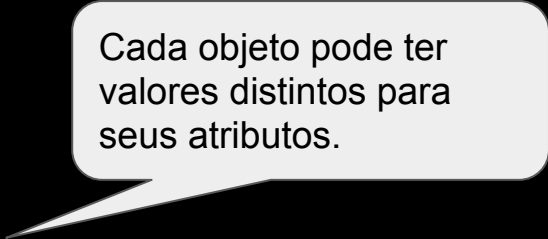
As instâncias podem acessar os atributos por meio de um ponto '.' seguindo o formato `objeto.atributo`. Dessa forma podem ser usado com uma variável comum.

```
class Pessoa {  
    nome = undefined;  
    idade = undefined;  
}  
  
var p1 = new Pessoa();  
var p2 = new Pessoa();  
  
p1.nome = "Ataliba"  
p1.idade = 22;  
  
p2.nome = "Emenengarda"  
p2.idade = 19;  
  
document.write("Nome: " + p1.nome + "<br>");  
document.write("Idade: " + p1.idade + "<br>");  
  
document.write("Nome: " + p2.nome + "<br>");  
document.write("Idade: " + p2.idade + "<br>");
```

Classes e Objetos

As instâncias podem acessar os atributos por meio de um ponto '.' seguindo o formato `objeto.atributo`. Dessa forma podem ser usado com uma variável comum.

```
class Pessoa {  
    nome = undefined;  
    idade = undefined;  
}  
  
var p1 = new Pessoa();  
var p2 = new Pessoa();  
  
p1.nome = "Ataliba"  
p1.idade = 22;  
  
p2.nome = "Emenengarda"  
p2.idade = 19;  
  
document.write("Nome: " + p1.nome + "<br>");  
document.write("Idade: " + p1.idade + "<br>");  
  
document.write("Nome: " + p2.nome + "<br>");  
document.write("Idade: " + p2.idade + "<br>");
```



Cada objeto pode ter valores distintos para seus atributos.

Criando métodos

Classes e Objetos

Um método é uma função criada dentro da declaração da classe.

```
class Conta {  
    saldo = 0;  
    numero = undefined;  
  
    deposito(valor) {  
        this.saldo += valor;  
    }  
  
    saque(valor) {  
        this.saldo -= valor;  
    }  
}
```

Classes e Objetos

Um método é uma função criada dentro da declaração da classe.

```
class Conta {  
    saldo = 0;  
    numero = undefined;  
  
    deposito(valor) {  
        this.saldo += valor;  
    }  
  
    saque(valor) {  
        this.saldo -= valor;  
    }  
}  
  
var conta = new Conta();  
  
conta.deposito(1200.00);  
conta.deposito( 600.00);  
conta.saque(500);  
  
document.write("Saldo: " + conta.saldo);
```


Classes e Objetos

Um método é uma função criada dentro da declaração da classe.

this é utilizada para reforçar que o atributo pertence ao objeto.

Pode ser traduzida como “isto” ou “ele próprio”.

```
class Conta {  
    saldo = 0;  
    depositar(valor) {  
        this.saldo += valor;  
    }  
    saque(valor) {  
        this.saldo -= valor;  
    }  
}  
  
var conta = new Conta();  
  
conta.deposito(1200.00);  
conta.deposito( 600.00);  
conta.saque(500);  
  
document.write("Saldo: " + conta.saldo);
```

O método constructor

Classes e Objetos

Um constructor é um método especial responsável por criar objetos.

```
class Conta {  
  
    constructor (saldoInicial, numero) {  
        this.saldo = saldoInicial;  
        this.numero = numero;  
    }  
  
    deposito(valor) {  
        this.saldo += valor;  
    }  
  
    saque(valor) {  
        this.saldo -= valor;  
    }  
}
```

Classes e Objetos

O construtor é chamado sempre que um novo objeto é criado.

```
var conta = new Conta(800, "123-89");  
  
conta.deposito(1200.00);  
conta.deposito( 600.00);  
conta.saque(500);  
  
document.write("Saldo: " + conta.saldo);
```

Classes e Objetos

```
class Pessoa {
    constructor(nome, sobrenome, idade) {
        this.nome = nome;
        this.sobrenome = sobrenome;
        this.idade = idade;
    }

    nomeCompleto() {
        return this.nome + " " + this.sobrenome;
    }
}

var p1 = new Pessoa("Ataliba", "Fisher", 22);
var p2 = new Pessoa("Emenengarda", "Poper", 19);

document.write("Nome: " + p1.nomeCompleto() + "<br>");
document.write("Idade: " + p1.idade + "<br>");

document.write("Nome: " + p2.nomeCompleto() + "<br>");
document.write("Idade: " + p2.idade + "<br>");
```

Resuminho

- Uma classe descreve algo do mundo real por meio de seus atributos e métodos.
- Uma classe é um tipo de dado, e portanto, a partir dela, podemos criar ‘variáveis’, que chamamos objetos.

- Uma classe pode ser instanciada várias vezes, gerando os objetos da classe.
- Todos os atributos e métodos comuns de um mesmo tipo são definidos na classe.
- Objetos da mesma classe comem os mesmo atributos e métodos.

Podemos dizer que uma **classe** descreve algo real de forma **genérica**, ou abstrata. Por outro lado, um **objeto** leva em conta as **especificidades** de cada elemento.

Para saber mais...

- <https://blog.schoolofnet.com/tudo-o-que-voce-precisa-saber-sobre-classes-no-javascript/>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes>
- https://www.w3schools.com/js/js_classes.asp
- <https://www.programiz.com/javascript/classes>
- <https://www.samanthaming.com/tidbits/39-es6-classes/>

Exercícios

Exercícios

1. Crie uma classe que representa um produto. A classe deve ter os atributos, nome, marca, preço de custo e preço de venda. Deve ter também um método que calcula e retorna o lucro.

Crie objetos dessa classe e escreva na tela seus atributos e o valor do lucro.