

JavaScript



ANA CAROLINA N R GRACIOSO
carol.nrg@gmail.com

Agenda

- Programação Orientada a Objetos:
 - Objeto, Classe
 - Propriedades e Métodos
 - Herança;
 - Encapsulamento;
 - Abstração;
 - Especialização;
 - Polimorfismo;
 - Associação.

Herança

- Herança é uma maneira de criar uma classe como uma versão especializada de uma ou mais classes
- **JavaScript suporta apenas herança de classe única.**
- A classe especializada é comumente chamada de **filha**, e a outra classe é comumente chamada de **pai**.
- Em JavaScript isso é feito nomeando uma instância da classe pai para a classe filha, e então especializa-a.
- Em navegadores modernos também pode-se usar **Object.create** para implementar herança.

Herança

```
// Define a classe Pessoa
function Pessoa () {
    var nome;
    this.setNome = function(vNome) {
        this.nome = vNome;
    }
    this.getNome = function() {
        return this.nome;
    }
}

// Define a classe Estudante
function Estudante () {
    // Chama o método pai
    Pessoa.call(this);
}

var estudante = new Estudante();
estudante.setNome("Fulano");
document.write(estudante.getNome());
```

Atividade

1 - Implemente uma classe Aluno e outra Professor que herdem atributos de uma classe Pessoa:

– Pessoa:

- nome
- sobrenome
- email
- data_nascimento
- nomeCompleto()

– Aluno:

- curso

– Professor:

- area_atuacao
- link_lattes

Encapsulamento

- No exercício anterior, nem a classe **Estudante** nem a classe **Professor** precisavam saber como o método **nomeCompleto()** da classe **Pessoa** seria implementado, mas ainda puderam utilizar este método;
- Isso se chama **encapsulamento**, pelo qual cada classe herda os métodos de seu pai e só precisa definir as coisas que deseja mudar.

Abstração

- A junção de herança, métodos, propriedades de um objeto devem refletir adequadamente um modelo da realidade.
- No exercício, abstraímos informações dos professores e alunos em uma classe Pessoa.
 - Outro exemplo:
 - Bola de Futebol de Couro;
 - Bola de Futebol;
 - Bola.

Especialização

- “Herança é uma maneira de criar uma classe como uma versão **especializada** de uma ou mais classes”.
 - No exercício, a classe Professor continha dados especializados (area_atuacao, link_lattes) de uma pessoa.
 - Outro exemplo:
 - Bola;
 - Bola de Futebol;
 - Bola de Futebol de Couro.

Polimorfismo

- Classes diferentes podem definir métodos com o mesmo nome; os métodos têm como escopo a classe a qual foram definidos, a menos que duas classes possuam uma relação pai-filho.

Polimorfismo

```
function Pessoa () {  
    var nome;  
    this.setNome = function(vNome) {  
        this.nome = vNome;  
    }  
    this.getNome = function() {  
        return this.nome;  
    }  
}  
  
function Carro () {  
    var nome;  
    this.setNome = function(vNome) {  
        this.nome = vNome;  
    }  
    this.getNome = function() {  
        return this.nome;  
    }  
}
```

Associação

- É o mecanismo pelo qual um objeto utiliza os recursos de outro.
- Pode tratar-se de uma associação simples "*usa um*" ou de um acoplamento "*parte de*".
- Exemplo:
 - Um humano *usa um* telefone;
 - A tecla "1" é *parte de* um telefone;
 - O pneu é *parte de* uma moto.

Associação

```
function Moto() {
    var pneu1 = new Pneu();
    var pneu2 = new Pneu();
    this.setPressao = function(vP1,vP2){
        pneu1.setPressao(vP1);
        pneu2.setPressao(vP2);
    }
    this.getPressao = function(){
        document.write(pneu1.getPressao());
        document.write(pneu2.getPressao());
    }
}

function Pneu (vPressao) {
    var pressao;
    this.setPressao = function(vPressao){
        this.pressao = vPressao;
    }
    this.getPressao = function(){
        return this.pressao;
    }
}
```

Atividade

2 - Considerando um objeto a sua escolha do mundo real e implemente um código JavaScript para exemplificar os conceitos de **Polimorfismo** e **Associação**.

Atividade

600

Formulário de Cadastro:

Nome:
Digite seu nome completo

Email:
fulano.com

Data Nascimento:
dd/mm/aaaa

Email inválido

Telefone Fixo:
(99)99999-9999

Telefone Celular:
(99)99999-9999

☒ Professor
☐ Aluno

Área:
Digite sua área de atuação

Matrícula:
Digite sua matrícula

Lattes:
Digite aqui o endereço para seu Lattes

Enviar

Redefinir

Atividade

600

Formulário de Cadastro:

Nome:
Digite seu nome completo

Email:
fulano.com

Data Nascimento:
dd/mm/aaaa

Email inválido

Telefone Fixo:
(99)99999-9999

Telefone Celular:
(99)99999-9999

☐ Professor

☒ Aluno

Curso:
Digite seu curso

Matricula:
Digite sua matrícula

Enviar

Redefinir

Atividade

- Seguir o padrão de formatação e posicionamento dos elementos segundo os wireframes;
 - Título: Helvetica, 26px, #000;
 - Label: Helvetica, 10px, #222, transparência 90%;
 - Input: Helvetica, 12px, #999, borda inferior solid, 1px, #ccc;
 - Botões: Helvetica, 12px, fundo #eee, borda #aaa, cantos arredondados 4px;
- Todos os campos são obrigatórios e devem ser validados com o evento onBlur. Caso ao perder o foco o campo não tenha sido preenchido corretamente deverá ser exibida a mensagem de erro do mesmo em vermelho abaixo do campo (vide exemplo campo email). O evento onSubmit também deve fazer parte da validação para os casos em que ao carregar o formulário o mesmo já seja submetido sem que nenhum campo tenha sido focado.
 - Validações:
 - Nome: xxx xxx
 - Email: xxx@xxx.xxx
 - Data Nascimento: dd/mm/aaaa
 - Telefone Fixo: (xx)xxxx-xxxx preencher os parênteses e traço automaticamente na digitação
 - Telefone Celular: (xx)xxxx-xxxx preencher os parênteses e traço automaticamente na digitação
 - Curso, Área de Atuação e Curso: obrigatórios para respectivos perfis
 - Matrícula professor: 5 dígitos / Matrícula aluno: 10 dígitos
- Todos os dados preenchidos devem ser armazenados em um objeto da classe Aluno/Professor que herdará dados da classe Pessoa.

Referência

- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript