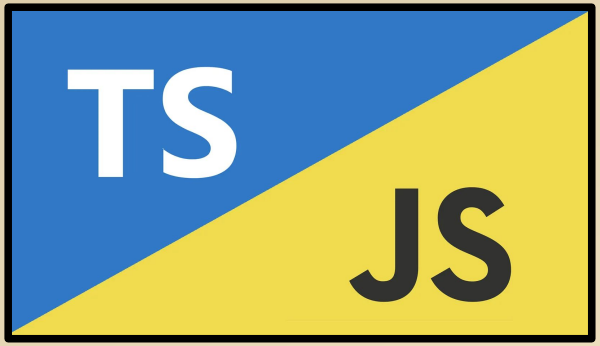
- Herança
- Associação
- Agregação
- Composição
- Dependência



JavaScript - Parte 3

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

TS JS

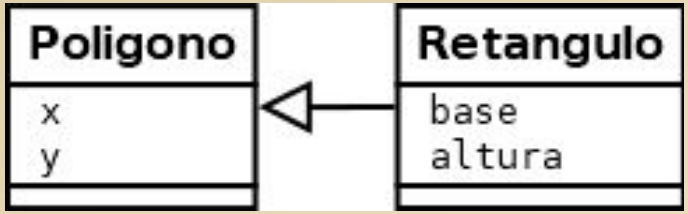
Herança

Herança

- ➔ Capacidade de classes específicas compartilharem atributos e métodos de classe genérica
- ➔ Classe genérica estabelece “molde” ou “base” para que outra classe adiciona a propriedade
- ➔ instanceof → operador retorna verdadeiro se um objeto “é” do mesmo tipo classe

TS JS

Herança



Herança

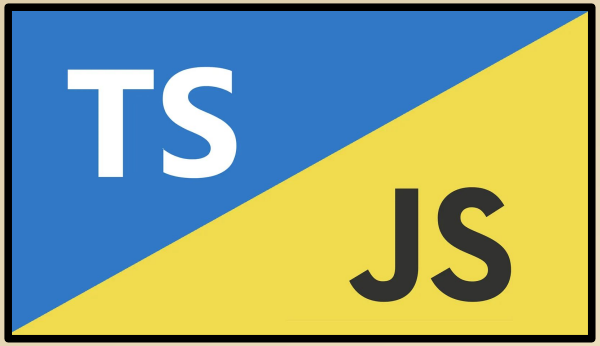
```
class Poligono{
  constructor(){
    this.x = 0;
    this.y = 0;
  }  }
class Retangulo
 extends Poligono{
  constructor(){
    super();
    this.base=0;
    this.altura=0;
  }  }
```

ES6

```
function Poligono(){
  this.x = 0;
  this.y = 0; }
function Retangulo(){
  Poligono.call(this);
  this.base=0;
  this.altura=0;  }
```

ES5

- Herança
- Associação
- Agregação
- Composição
- Dependência



JavaScript - Parte 3

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

TS JS

Associação

Associação

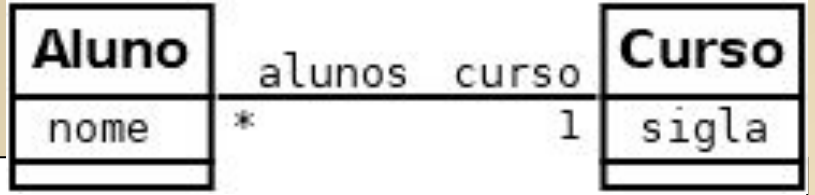
- A associação ocorre quando uma classe usa outra como um atributo ou referência, mas ambas podem existir separadamente.
- Permite que um objeto se relacione com outro sem dependência total.
- O objeto associado não precisa ser destruído se o objeto principal for removido.
- A classe associada pode ser compartilhada por múltiplos objetos.

TS

Associação

ES6

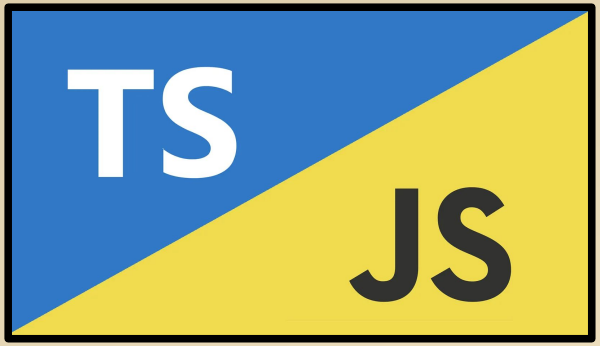
```
class Aluno{
  constructor(nome){
  this.nome = nome;
  this.cursos = null; }}
class Curso{
  constructor(sigla){
    this.sigla = sigla;
    this.alunos = [];}}
var ze=new Aluno("José");
var m=new Aluno("Maria");
var c=new Curso("Pblc01");
c.alunos.push(ze);
c.alunos.push(m);
console.log(c.alunos[0].nome)
console.log(c.alunos[1].nome)
```



ES5

```
function Aluno(nome) {
  this.nome=nome;
  this.curso=null; }
function Curso(sigla) {
  this.sigla=sigla;
  this.alunos=[]; }
var ze=new Aluno("José");
var m=new Aluno("Maria");
var c=new Curso("Pblc01");
c.alunos.push(ze);
c.alunos.push(m);
console.log(c.alunos[0].nome);
console.log(c.alunos[1].nome);
```

- Herança
- Associação
- Agregação
- Composição
- Dependência



JavaScript - Parte 3

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

TS JS

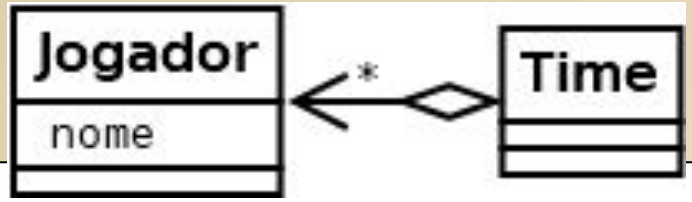
Agregação

Agregação

- A agregação é um tipo de associação onde um objeto contém outro, mas o objeto agregado pode existir separadamente.
- O objeto agregado não é destruído se o objeto principal for removido.
- A classe principal “possui” a outra, mas a outra pode existir sozinha.

TS JS

Agregação



Agregação

```
class Jogador {
  constructor(nome) {
  this.nome = nome; }}
class Time {
  constructor() {
  this.jogadores = [];}}
const sts = new Time();
sts.jogadores.push(
new Jogador("Neymar"));
console.log(
  sts.jogadores[0].nome
);
```

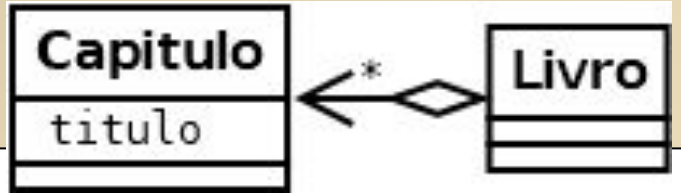
ES6

```
function Jogador(nome) {
  this.nome = nome;
}
function Time() {
  this.jogadores = [];
}
var sts = new Time();
sts.jogadores.push(
new Jogador("Neymar"));
console.log(
  sts.jogadores[0].nome
);
```

ES5

TS JS

Agregação



Composição

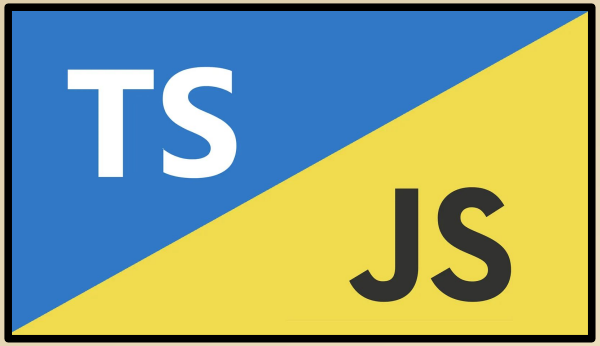
```
class Livro {
  constructor() {
    this.capitulo= [] }}
class Capitulo {
  constructor(titulo) {
    this.titulo = titulo;}}
const meuLivro = new
Livro();
meuLivro.capitulo.push(new
Capitulo("Introdução"));
console.log(meuLivro.capitu
lo[0].titulo);
```

ES6

```
function Livro(){
  this.capitulo = []
}
function Capitulo(titulo){
  this.titulo = titulo;
}
var meuLivro = new Livro();
meuLivro.capitulo.push(new
Capitulo("Introdução"));
console.log(meuLivro.capitu
lo[0].titulo);
```

ES5

- Herança
- Associação
- Agregação
- Composição
- Dependência



JavaScript - Parte 3

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

TS JS

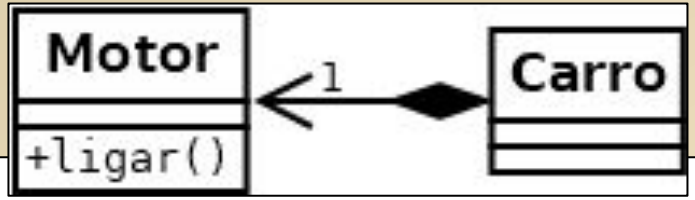
Composição

Composição

- A composição é um tipo forte de associação, onde um objeto não pode existir sem o outro.
- O objeto composto é parte essencial do objeto principal.
- Se o objeto principal for destruído, o objeto dependente também será.

TS JS

Composição



Associação

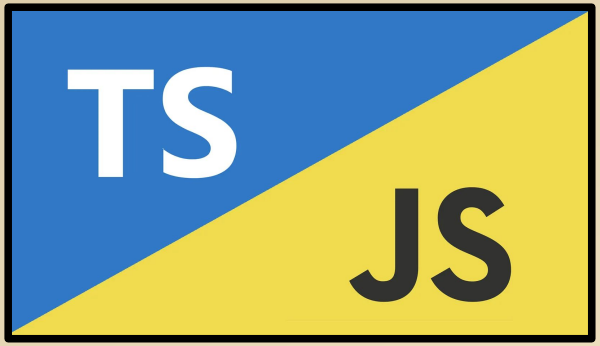
```
class Motor {
  ligar() {
  console.log("Motor
ligado");
}}
class Carro {
  constructor() {
    this.motor=new
Motor(); }}
const fusca = new
Carro(new Motor());
fusca.motor.ligar();
```

ES6

```
function Motor() {
  this.ligar = function(){
  console.log("Motor
ligado");
}}
function Carro() {
    this.motor=new Motor();
}
var fusca= new Carro(new
Motor());
fusca.motor.ligar();
```

ES5

- Herança
- Associação
- Agregação
- Composição
- Dependência



JavaScript - Parte 3

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano

TS JS

Dependência

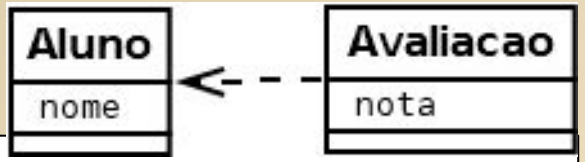
Dependência

- A dependência ocorre quando uma classe precisa da existência de outra para funcionar, mesmo que não a contenha como parte permanente
- Se o objeto dependente não existir, a funcionalidade da classe principal não é possível.
- A classe não armazena o outro objeto como atributo.
- A existência do objeto auxiliar é fundamental para a execução da ação.

TS JS

Dependência

Dependência



ES6

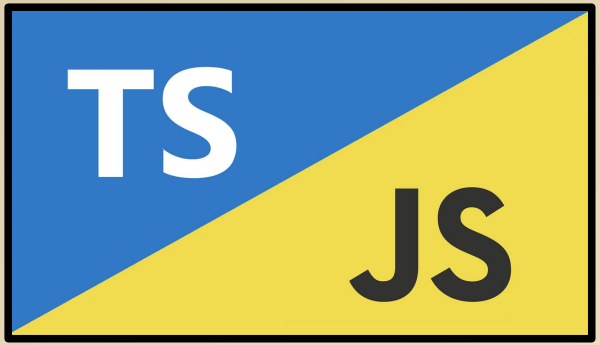
```
class Aluno{
  constructor(nome) {
  this.nome = nome;  }}
class Avaliacao{
  constructor(aluno) {
  this.aluno = aluno;
  this.nota = 0; }}
var ze = new Aluno("José")
var prova = new
Avaliacao(ze);
prova.nota = 9;
console.log(`${prova.aluno.
nome} tirou
${prova.nota}`);
```

```
function Aluno(nome){
  this.nome = nome; }
function Avaliacao(aluno){
  this.aluno = aluno;
  this.nota = 0; }
var ze = new Aluno("José")
var prova = new
Avaliacao(ze);
prova.nota = 9;
console.log(prova.aluno.nome
+ " tirou " + prova.nota);
```

ES5

Prof. Enzo Seraphim

Profa. Bárbara Pimenta Caetano



JavaScript - Parte 3

Os logotipos, marcas comerciais e nomes de produtos citados nesta publicação tem apenas o propósitos de identificação e podem ser marcas registradas de suas respectivas companhias.