06

Estendendo interfaces

Transcrição

Excelente, mas a implementação de `Imprimivel` e `Igualavel` é algo bem comum de implementarmos para praticamente todos nossas classes do modelo. É por isso que podemos criar uma interface que estende essas duas que criamos, bastando implementarmos a nova interface agregado.

Vamos criar a interface `MeuObjeto` :

```
import { Imprimivel, Igualavel } from './index';

export interface MeuObjeto<T> extends Imprimivel, Igualavel<T>  { }
```

```
export * from './Negociacao';
export * from './Negociacoes';
export * from './NegociacaoParcial';
export * from './Imprimivel';
export * from './Igualavel';
export * from './MeuObjeto';
```

Uma interface pode estender quantas interfaces forem necessárias. No caso, `MeuObjeto` teve que usar o tipo genético `<T>` para que esse seja considerado na interface `Igualave` .

Agora, vamos fazer com que `Negociacao` e `Negociacoes` implementem `MeuObjeto` :

```
import { MeuObjeto } from './index';

export class Negociacao implements MeuObjeto<Negociacao> {

   // código omitido
}
```

Por fim, vamos implementar em `Negociacoes` :

```
import { Negociacao, MeuObjeto } from './index';

export class Negociacoes implements MeuObjeto<Negociacoes> {

    private _negociacoes: Negociacao[] = [];

    adiciona(negociacao: Negociacao): void {

        this._negociacoes.push(negociacao);
    }

    paraArray(): Negociacao[] {
```

```
        return ([] as Negociacao[]).concat(this._negociacoes);
    }

    paraTexto(): void {
        console.log('-- paraTexto --');
        console.log(JSON.stringify(this._negociacoes));
    }

   // precisamos implementar o método ehIgual
    ehIgual(negociacoes: Negociacoes): boolean {

        return JSON.stringify(this._negociacoes) == JSON.stringify(negociacoes.paraArray());
    }
}
```