

Builder em TypeScript

Mario, Rafael, Robson e Wagner

Definição

Builder é um padrão de projeto criacional, que permite a construção de objetos que demandam bastante configuração.

```
class Product1 {  
    public parts: string[] = [];  
  
    public listParts(): void {  
        console.log(  
            `Product parts: ${this.parts.join(', ')}\n`  
        );  
    }  
}
```

```
interface Builder {  
    producePartA(): void;  
    producePartB(): void;  
    producePartC(): void;  
}
```

```
class ConcreteBuilder1 implements Builder {  
    private product: Product1;  
    constructor() {  
        this.reset();  
    }  
    public reset(): void {  
        this.product = new Product1();  
    }  
    [...]  
}
```

```
class ConcreteBuilder1 implements Builder {  
    [...]  
    public producePartA(): void {  
        this.product.parts.push('PartA1');  
    }  
    [...]  
    public producePartC(): void {  
        this.product.parts.push('PartC1');  
    }  
    [...]  
}
```

```
class ConcreteBuilder1 implements Builder {  
    [...]  
    public getProduct(): Product1 {  
        const result = this.product;  
        this.reset();  
        return result;  
    }  
}
```

```
class Director {  
    private builder: Builder;  
    public setBuilder(builder: Builder): void {  
        this.builder = builder;  
    }  
    public buildMinimalViableProduct(): void {  
        this.builder.producePartA();  
    }  
    public buildFullFeaturedProduct(): void {  
        this.builder.producePartA();  
        this.builder.producePartB();  
        this.builder.producePartC();  
    }  
}
```



```
function clientCode(director: Director) {  
    const builder = new ConcreteBuilder1();  
    director.setBuilder(builder);  
  
    console.log('Standard basic product:');  
    director.buildMinimalViableProduct();  
    builder.getProduct().listParts();  
  
    console.log(  
        'Standard full featured product:'  
    );  
    director.buildFullFeaturedProduct();  
    builder.getProduct().listParts();  
    [...]  
}
```

```
function clientCode(director: Director) {  
    [...]  
    console.log('Custom product:');  
    builder.producePartA();  
    builder.producePartC();  
    builder.getProduct().listParts();  
}
```

```
const director = new Director();  
clientCode(director);
```

Executando Código

Perguntas?