

Encapsulamento

Vejamos a seguinte definição da classe `Conta` :

```
class Conta {  
  
  constructor(titular, conta) {  
  
    this.titular = titular;  
    this.conta = conta;  
    this.saldo = 0.0;  
  }  
  
  deposita(valor) {  
    console.log('Valor depositado: ' + valor);  
    this.saldo+=valor;  
  }  
}  
  
var conta = new Conta('Mingau', 171);  
conta.deposita(100);
```

Para depositarmos um valor, chamamos o método `deposita` . É neste método que logamos no console o valor depositado na conta. **Logar essa informação é imprescindível para um sistema de auditoria.**

Podemos dizer que:



Por mais que seja importante atualizarmos o `saldo` da conta somente através do método `deposita` , para que a operação de log seja realizada, ninguém impede que o programador altere o `saldo` diretamente na propriedade da classe, conforme o exemplo abaixo:

```
var conta = new Conta('Mingau', 171);  
conta.saldo+= 100;
```

Com isso, a operação de log não é realizada.

B

JavaScript possui nativamente uma sintaxe para declarar atributos privados. É só prefixar os atributos com `_` .



JavaScript não possui nativamente uma sintaxe para declarar atributos privados.



Uma solução barata é apelar para a **convenção** do mundo javascript na qual atributos prefixados por `_` não devem ser acessados diretamente pelos desenvolvedores. Eles devem utilizar as próprias classes para poder acessá-los. Sendo assim, a única maneira **convencionada** para acessar esses atributos é através de métodos. No caso da nossa classe `Conta` , poderíamos fazer:

```
class Conta {  
  
  constructor(titular, conta) {  
  
    this.titular = titular;  
    this.conta = conta;
```

```
    this._saldo = 0.0;
  }

  deposita(valor) {
    console.log('Valor depositado: ' + valor);
    this._saldo+=valor;
  }

  getSaldo() {
    return this._saldo;
  }
}

var conta = new Conta('Mingau', 171);
conta.deposita(100);
console.log(conta.getSaldo()); // se não pode acessar this._saldo, é preciso aces
```

CURIOSIDADE: a linguagem TypeScript da Microsoft permite definir atributos privados com o modificador `private`. TypeScript nada mais é do que um superset do ES2015+ com a adição de novos recursos, inclusive aquele que define atributos privados.

PRÓXIMA ATIVIDADE