

1.

Pessoa
+ nome + idade + deitado + emPe - testePrivada
+ levantar +deitar

// Criação de CLASSE: Pessoa

```
class Pessoa {  
    // Crio uma função para as PROPRIEDADES da classe  
    constructor(nome, idade, deitado, emPe) {  
        // atributos públicos na instancia da classe: this.variável  
        // atributos privados na instancia da classe: var variável =  
        ""  
  
        this.nome = nome;  
        this.idade = idade;  
        this.deitado = deitado;  
        this.emPe = emPe;  
    }  
}
```

```

        var testePrivada = 55;
    }
    // Crio os METODOS
    levantar() {
        // Confiro se está deitado, se sim ativa a função
        if (this.deitado) {
            this.deitado = false;
            this.emPe = true;
            console.log(this.nome + ' levantou.');
```

```
        }
```

```
        else {
```

```
            console.log(this.nome + ' já está em pé.');
```

```
        }
```

```
    }
```

```
    deitar() {
```

```
        if (this.deitado) {
```

```
            console.log(this.nome + ' já está deitado.');
```

```
        }
```

```
        else {
```

```
            this.emPe = false;
```

```
            this.deitado = true;
```

```
            console.log(this.nome + ' deitou.')
```

```
        }
```

```
    }
```

```
}
```

```
// Criação de OBJETO
```

```
const p1 = new Pessoa('Joao', 40, true, false);
```

```
// Imprimo todas as propriedades do objeto
console.log(p1); // Pessoa {nome: 'Joao', idade: 40, deitado: true,
emPe: false}

// Imprimo uma única propriedade
console.log(p1.deitado); // true

// Usar métodos: objeto.metodo();
p1.levantar(); // Joao levantou.
p1.levantar(); // Joao ja está em pé.
p1.deitar(); // Joao deitou.
p1.deitar(); // Joao ja está deitado.


// teste para atributo privado onde apenas a classe atual tem acesso
console.log(p1.testePrivada); // undefined


const p2 = new Pessoa('Paulo', 23, false, true);


console.log(p2); // Pessoa {nome: 'Paulo', idade: 23, deitado: false,
emPe: true}
p2.deitar(); // Paulo deitou.
p2.levantar(); // Paulo levantou.
console.log(p2.idade); // 23
```

ContaCorrente
+ numero + saldo + statusEspecial + limite + movimentacoes
+ sacar + depositar + solicitarAlmentoLimite + verSaldo

```
// crio a classe
```

```
class ContaCorrente {
```

```
    // crio os atributos da classe
```

```
    constructor(numero, saldo, statusEspecial, limite, movimentacoes)
{
```

```
    this.numero = numero;
```

```
    this.saldo = saldo;
```

```
    this.statusEspecial = statusEspecial;
```

```
    this.limite = limite;
```

```
    this.movimentacoes = movimentacoes;
```

```
}
```

```
// crio os métodos da classe
```

```
sacar(valorSacar) {
```

```
    // se o valor saque é menor ou igual ao saldo
```

```
    if(valorSacar <= this.saldo) {
```

```
        this.saldo = this.saldo - valorSacar;
```

```
        console.log("Saldo atual: R$" + this.saldo);
```

```
        this.movimentacoes++;
```

```

    }
    else {
        console.log("Saldo insuficiente.");
    }
}

depositar(valorDepositar) {
    this.saldo = this.saldo + valorDepositar;
    this.movimentacoes++;
    console.log("Saldo atual: R$" + this.saldo);
}

solicitarAlmentoLimite() {
    if(this.statusEspecial){
        console.log("Status especial, limite almentado.");
        this.limite += 500;
        console.log("Limite atual: R$" + this.limite);
    }
    else {
        console.log("Você não status especial para que seja feita
essa solicitação.");
        console.log("Limite atual: R$" + this.limite);
    }
}
}

// crio o objeto1
const cc1 = new ContaCorrente('252409', 1500.99, false, 5000.00, 0);
cc1.sacar(200.90); //Saldo atual: R$1300.09
cc1.sacar(300.05); //Saldo atual: R$1000.04
cc1.depositar(500.00); //Saldo atual: R$1500.04
cc1.solicitarAlmentoLimite(); //Você não status especial para que seja
feita essa solicitação.

```

```
console.log(cc1); //ContaCorrente {numero: '252409', saldo: 1500.04,
statusEspecial: false, limite: 5000, movimentacoes: 3}
```

```
const cc2 = new ContaCorrente('123459', 200.00, true, 4000.00, 0);
```

```
cc2.depositar(800); //Saldo atual: R$1000
```

```
cc2.solicitarAlmentoLimite(); //Status especial, limite almentado.
Limite atual: R$4000
```

```
cc2.solicitarAlmentoLimite(); //Status especial, limite almentado.
Limite atual: R$5000
```

```
console.log(cc2); //ContaCorrente {numero: '123459', saldo: 1000,
statusEspecial: true, limite: 5000, movimentacoes: 1}
```

3.

Produto
+ nome + preco + quantidade
+ valorProduto

```
// crio classe
```

```
class Produto {
```

```
  // crio atributos
```

```
  constructor(nome, preco, quantidade) {
```

```
    this.nome = nome;
```

```

        this.preco = preco;
        this.quantidade = quantidade;
    }
    valorProduto(){
        if(this.quantidade <= 10) {
            console.log("Valor " + this.nome + "s sem desconto: R$" +
            (this.quantidade*this.preco));
        }
        else if(10 < this.quantidade && this.quantidade <= 20) {
            console.log("Valor " + this.nome + "s com 10% de desconto:
R$" + (this.quantidade*this.preco -
(this.quantidade*this.preco)*0.10));
        }
        else if(20 < this.quantidade && this.quantidade <= 50) {
            console.log("Valor " + this.nome + "s com 20% de desconto:
R$" + (this.quantidade*this.preco -
(this.quantidade*this.preco)*0.20));
        }
        else if(50 < this.quantidade) {
            console.log("Valor " + this.nome + "s com 25% de desconto:
R$" + (this.quantidade*this.preco -
(this.quantidade*this.preco)*0.25));
        }
    }
}

// crio os objetos (produtos)
const m1 = new Produto("leite", 4, 5);
m1.valorProduto(); //Valor leites sem desconto: R$20

const m2 = new Produto("carne", 50.00, 20);
m2.valorProduto(); //Valor carnes com 10% de desconto: R$900

```

```
const m3 = new Produto("ovo", 0.5, 30);  
m3.valorProduto(); //Valor ovos com 10% de desconto: R$13.5
```

```
const m4 = new Produto("folha", 0.10, 100);  
m4.valorProduto(); //Valor folhas com 25% de desconto: R$7.5
```