//INTERFACES:

//As interfaces são classes que nós definimos em POO com o único objetivo de interligar os objetos de classes diferentes gerando algum tipo de comportamento.

//Muitas pessoas defendem que as interfaces são o melhor método de interagir entre classes, até melhor que a herança, mas essa é uma questão de opinião pessoal.

//Mas usamos interfaces muitas vezes para gerar interação entre objetos de classes diferentes, vejamos como utilizá-la:

class Pessoa { //Temos uma classe para cadastrar clientes e funcionários...

    constructor(nome){

        this.\_nome = nome

        this.\_senha //A senha não pode ser cadastrada diretamente...

    }

    cadastrar\_senha(senha){ //É preciso que a instância chame o método para que uma senha seja cadastrada...

        this.\_senha = senha

    }

    autenticar(senha){ //O método autenticar é usado para fazer a conferência entre a senha passada e a cadastrada...

        return console.log(senha == this.\_senha)

    }

}

class Funcionario extends Pessoa { //Temos uma classe herdada para o Funcionário...

    constructor(nome){

        super(nome)

    }

}

class Cliente extends Pessoa { //E outra para o Cliente...

    constructor(nome){

        super(nome)

    }

}

class AutenticarSenha{ //Perceba que a classe "AutenticarSenha" não herda de nenhuma classe, ela vai trabahar como uma

    constructor(nome, senha){ //Composição, onde vai pegar o objeto gerado em alguma classe por através do parâmetro e

        nome.autenticar(senha) //"nome" vai comparar o valor da senha passada no parâmetro "senha". O que faz dela uma

    }                           //interface é a comparação se o objeto possuí o método "autenticar", qualquer classe

}                               //que tiver esse método vai conseguir ser operar na classe "AutenticarSenha"...

let romario = new Cliente('Romário') //Veja que geramos um objeto na classe Cliente...

romario.cadastrar\_senha('456') //Geramos uma senha, cadastrando ela...

console.log(romario.\_senha) //Veja que a senha realmente foi gerada...

/\*RESULTADO NO CONSOLE:

456

\*/

let autenticacaoRomario = new AutenticarSenha(romario, '456') //Veja que gerando uma nova instância e usando o objeto

/\*RESULTADO NO CONSOLE:                                         //"romario" com a senha dele, podemos usar a classe

true                                                            //"AutenticarSenha" normalmente...

\*/

let renato = new Funcionario('Romário') //O mesmo acontece na classe "Funcionario"...

renato.cadastrar\_senha('123')

console.log(renato.\_senha)

/\*RESULTADO NO CONSOLE:

123

\*/

let autenticacaoRenato = new AutenticarSenha(renato, '123')

/\*RESULTADO NO CONSOLE:

true

\*/