

//SISTEMA DE VERIFICAÇÃO DE OBJETOS:

//Sistemas de Verificação são blocos condicionais que podemos implementar nas nossas interfaces para conferir se atributos e métodos realmente existem nos objetos que estão interagindo nas nossas interfaces.

//Além disso, eles facilitam a verificação e manutenção do código, pois, em vez de mostrar uma mensagem de Error que vai parar a nossa execução, podemos colocar outros tipos de mensagem que nos ajudem a entender por que determinado objeto não pode operar sobre uma determinada classe.

//Vejamos como podemos utilizá-la:

```
class Pessoa{ //Veja que temos uma classe Pessoa onde é possível cadastrar o nome e a senha de um usuário...
    constructor(nome){
        this._nome = nome
        this._senha
    }

    cadastrarSenha(senha){
        return this._senha = senha
    }

    autenticar(senha){ //Esse é o detalhe importante, dentro de Pessoa temos uma chave "autenticar" e ela é uma função...
        return this._senha == senha
    }
}

class Animal { //Já na classe "Animal" podemos só cadastrar o nome do bichinho e mostrá-lo por através de uma função...
    constructor(nome){
        this._nome = nome
    }

    mostrarNome(){
        return this._nome
    }
}
```

```
//Nossa missão é implementar um sistema de autenticação que verifique se a classe autenticar realmente existe dentro do nosso objeto, e se ela é mesmo uma função e não um atributo...

class SistemaDeAutenticacao{ //Aqui temos a interface com o sistema de verificação de tipo de objeto ativa, perceba que
    static login(nome, senha){ //ela possui dentro da função estática login() um bloco condicional que chama outra
        if(SistemaDeAutenticacao.verificacaoDeFuncao(nome)){ //função estática dessa classe, a "verificacaoDeFuncao" que
            return nome.autenticar(senha) //verifica se a função "autenticar" é mesmo uma chave do
        } else { //objeto passado como argumento, e também, se essa chave
            return 'Esse objeto não possui a função de autenticação.' //é uma função...
        }
    }

    static verificacaoDeFuncao(nome){ //Veja que essa função faz uma comparação dupla, ela vê se a chave "autenticar"
        return "autenticar" in nome && nome.autenticar instanceof Function //realmente existe no objeto passado e também
    } //se "autenticar" é uma função mesmo, e não
} //um atributo...

let roberto = new Pessoa('Roberto') //Como roberto é um objeto da classe Pessoa ele pode possuir senha e ser autenticado
roberto.cadastrarSenha('123')
console.log(roberto._senha)
/*RESULTADO NO CONSOLE:
123
*/

let autenticarRoberto = SistemaDeAutenticacao.login(roberto, '123')
console.log(autenticarRoberto)
/*RESULTADO NO CONSOLE:
true
*/

let black = new Animal('Black') //Mas Black só possui nome, e a sua classe não possui uma chave autenticar que seja
console.log(black.mostrarNome()) //uma função...
/*RESULTADO NO CONSOLE:
Black
*/
```

```
let autenticarBlack = SistemaDeAutenticacao.login(black, '123')
console.log(autenticarBlack)
/*RESULTADO NO CONSOLE:
Esse objeto não possui a função de autenticação.
*/
```