

### Exercício 04

1. Assinale verdadeiro ou falso:

- (F) Objetos são modelos para classes;
- (F) Atributos de uma classe devem ser obrigatoriamente inicializados para que as classes compilem;
- (F) Uma variável declarada dentro de um método deve ser inicializada para que a classe seja compilável;
- (V) Uma variável que seja uma classe declarada em um método é automaticamente inicializada com undefined;
- (V) Construtores são rotinas especiais que servem para inicializar e configurar os objetos no momento da instanciação;
- (V) Construtores não possuem tipo de retorno e podem ou não ter parâmetros;
- (V) Uma classe pode ter várias instâncias.

2. Suponha uma classe Hotel que sirva apenas para guardar a quantidade de solicitações de reservas feitas conforme abaixo:

```
class Hotel {
    quantReservas : number;
    adicionarReserva() : void {
        this.quantReservas++;
    }
}
```

Podemos afirmar que haverá um problema de compilação, pois a variável inteira não foi inicializada previamente? Justifique.

R: É possível afirmar que haverá um erro de compilação no modo strict do TypeScript, pois o atributo quantReservas foi declarado sem um inicializador e sem estar sendo atribuído dentro de um construtor. Isso aconteceu devido ao fato de que no modo strict o TypeScript exige que as propriedades de classe sejam inicializadas para garantir que elas não sejam undefined em tempo de execução. Quando eu testei fora do modo strict não teve problema.

3. Ainda sobre a classe do exemplo anterior, considere o código abaixo:

```
let hotel : Hotel = new Hotel(2);
console.log(hotel.quantReservas);
```

Adicione o construtor que aceite um parâmetro inteiro e faça a inicialização do atributo quantReservas.

R : O código dessa questão se encontra no arquivo (q3.ts).

4. Considere a classe Radio e as instruções que fazem seu uso abaixo:

```
class Radio {  
    volume : number;  
    constructor(volume : number) {  
        this.volume = volume;  
    }  
}  
  
let r : Radio = new Radio();  
r.volume = 10;
```

Justifique o erro de compilação e proponha uma solução.

R: O erro de compilação se deve ao fato de que a classe Radio possui um construtor com um parâmetro do tipo number, valor esse que não foi passado na instânciação dessa classe. Por isso houve um erro, já que a chamada não correspondeu a assinatura definida. A solução para esse erro seria passar um parâmetro do tipo number durante o processo de inicialização, assim fornecendo o valor inicial exigido pelo construtor.

A solução em código dessa questão se encontra no arquivo (q4.ts).

5. Considerando o uso da classe Conta apresentada em aula e seu uso abaixo:

```
let c1: Conta = new Conta("1",100);  
let c2: Conta = new Conta("2",100);  
let c3: Conta;  
c1 = c2;  
c3 = c1;  
c1.sacar(10);  
c1.transferir(c2,50);  
console.log(c1.consultarSaldo());  
console.log(c2.consultarSaldo());  
console.log(c3.consultarSaldo());
```

- a. Qual o resultado dos dois "prints"? Justifique sua resposta.

R: O resultado de todos os prints foi 90. Pois, depois das retribuições que foram realizadas, as três variáveis instanciadas de Conta passam a apontar para o mesmo objeto na memória, que no caso é o c2, fazendo com que c1 e c3 passem a ser somente "apelidos" para o mesmo objeto. Dessa forma na c1.sacar foi sacado 10 de c2 e no c1.transferir foi retirado 50 de c2 que foi devolvido de volta para ele e, por fim, os console.log realizados foram "prints" do valor final de c2, ou seja, 90.

O teste em código dessa questão se encontra no arquivo (q5.ts).

- b. O que acontece com o objeto para o qual a referência c1 apontava?

R: A linha c1 = c2; faz com que o objeto referenciado por c1 seja perdido, como nenhuma outra variável aponta para este objeto após essa linha, ele acaba se tornando um objeto sem referências e, por causa disso, futuramente o Garbage Collector do TypeScript irá excluir esse objeto da memória automaticamente liberando espaço.

O teste em código dessa questão se encontra no arquivo (q5.ts).

6. Crie uma classe chamada Triangulo que:

- Possua 3 atributos inteiros representando os lados;
- Crie um método que retorna true se os lados formarem um triângulo de acordo com a regra:  $|b-c| < a < b+c$ ;
- Crie 3 métodos: `ehIsocetes()`, `ehEquilatero()` e `ehEscaleno()` que retorne verdadeiro caso o triângulo seja um dos tipos relacionados ao nome do método. Eles devem chamar antes de tudo, o método da questão b. e retornar false se esse método já retornar false também;
- Instancie classes Triangulo de diferentes lados e seus métodos.

R: O código dessa questão se encontra no arquivo (q6.ts).

7. Uma classe Equipamento com:

- a. um atributo ligado (tipo boolean)
- b. dois métodos `ligar()` e `desligar()`. O método `liga` torna o atributo `ligado` true e o método `desliga` torna o atributo `ligado` false.
- c. Crie um método chamado `inverter()`, que muda o status atual (se ligado, desliga...se desligado, liga)
- d. Crie um método que `estaLigado()` que retorna o valor do atributo `ligado`
- e. Altere o comportamento do método `ligar()` para caso o equipamento já esteja ligado, não ligue novamente. Faça o mesmo com o método `desligar()`.
- f. Instancie uma classe Equipamento e teste todos os seus métodos.

R: O código dessa questão se encontra no arquivo (q7.ts).

8. Altere a classe conta dos slides conforme as instruções abaixo:

- Altere o método `sacar` de forma que ele retorne verdadeiro ou falso. Caso o saque deixe saldo negativo, o mesmo não será realizado, retornando falso;
- Altere o método `transferir()` para que retorne também um valor lógico e que não seja feita a transferência caso o `sacar()` na conta origem não seja satisfeito.

R: O código dessa questão se encontra no arquivo (q8.ts).

9. A abordagem da questão anterior é retornar códigos de erro ou acerto. Já a da questão 10 é desconsiderar a alteração. Quais das duas você acha mais correta?

R: Eu acho a abordagem da questão 8 mais correta, pois com ela há um feedback explícito para o resultado das ações executadas na classe.

10. Crie uma classe chamada Jogador e nela:

- Crie 3 atributos inteiros representando força, nível e pontos atuais;
- Crie um construtor no qual os 3 parâmetros são passados e inicialize os respectivos atributos;
- Crie um método chamado `calcularAtaque`. Nele, calcule e retorne o valor da multiplicação de força pelo nível. Esse resultado é o dano de ataque do jogador;

- Crie um método chamado atacar em que é passado um outro jogador (atacado) como parâmetro. Nele é feita a subtração do dano (método calcularAtaque) dos pontos do atacado;
- Crie um método chamado estaVivo que retorna true caso o atributo pontos do jogador seja maior que zero e falso caso contrário.
- Altere o método atacar para usar o método está vivo e desconsiderar a operação, ou seja, não atacar, caso o jogador passado por parâmetro não esteja vivo.
- Avalie em com testes dois jogadores instanciados e inicializados através do construtor. Utilize o método de ataque de cada jogador e ao final, verifique qual jogador tem mais pontos.

R: O código dessa questão se encontra no arquivo (q10.ts).