

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUI</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ Curso: ADS Disciplina: Programação Orientada a Objetos Professor: Ely</p>
--	--

Exercício 04

1. Assinale verdadeiro ou falso:

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

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.

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.

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());

```

- Qual o resultado dos dois "prints"? Justifique sua resposta.
- O que acontece com o objeto para o qual a referência c1 apontava?

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.

7. Uma classe Equipamento com:

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

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.
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?
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 e 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.