

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

Exercício 03

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

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

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

2. Ainda sobre a classe do exemplo anterior, considere o método main 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.

3. Considere a classe Radio e as instruções 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.

4. 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);
c1 = c2;
c1.sacar(10);
c1.transferir(c2,50);
```

```
console.log(c1.saldo);  
console.log(c2.saldo);
```

- a. Qual o resultado dos dois "prints"? Justifique sua resposta.
 - b. O que acontece com o objeto para o qual a referência c1 aponta?
5. Crie uma classe chamada Jogador e nela:
- a. Crie 3 atributos inteiros representando força, nível e pontos atuais;
 - b. Crie um construtor no qual os 3 parâmetros são passados e inicialize os respectivos atributos;
 - c. Crie um método que calcule os pontos relativos a um ataque que são calculados pela multiplicação de força pelo nível;
 - d. Crie um método chamado atacar em que é passado um outro jogador como parâmetro e é feita a subtração de pontos de tal jogador baseado na quantidade de pontos do jogador atual ("this").
 - e. 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.
6. Altere a classe conta dos slides conforme as instruções abaixo:
- a. 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;
 - b. Altere o método transferir() para que o mesmo use os métodos sacar() e depositar(). Visto pelo prisma da "proteção do saldo", chamar outros métodos em vez de acessar o saldo diretamente é mais seguro?
 - c. 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;
 - d. Verifique as diferentes operações implementadas.
7. Crie uma classe chamada Produto e nela:
- a. Crie os atributos codigo, descricao, valor e um construtor que os inicialize;
 - b. Crie os métodos baixar(quantidade : number) e repor(quantidade : number) que reduzem e incrementam a quantidade disponível do produto;
 - c. Crie um atributo quantidadeMinima e reescreva o método baixar para que não seja possível realizar a baixa caso a operação deixe a quantidade abaixo da quantidade mínima;
 - d. Crie um método da classe Produto chamado reajusta(taxa : number) que reajusta em x% o valor do produto.
 - e. Crie um método chamado toString() que retorna a representação textual do produto concatenando todos os atributos.
 - f. Crie um método equals(Produto produto) que retorna true ou false se o produto passado como parâmetro possui o mesmo código;
 - g. Verifique as diferentes operações implementadas com testes.