

**Nome: Lucas Augusto Araújo Aguiar**

**Data: 05/10/2021**

## **Orientação a objetos**

**1) Em relação ao conceito de orientação a objetos, a afirmativa de que o estado de um objeto não deve ser acessado diretamente, mas sim por intermédio de métodos de acesso (ou propriedades) está diretamente relacionada ao conceito de:**

- ☐ Herança
- ☐ Interface
- ☐ Classe
- ☐ Polimorfismo
- ☒ Encapsulamento

**2) É o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura) mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse.**

- ☐ Classe
- ☐ Herança
- ☒ Polimorfismo
- ☐ Encapsulamento

**3) No contexto da Orientação a Objetos, A definição das características (estado) e do comportamento de um objeto é conhecido por:**

- ☒ Classe
- ☐ Herança
- ☐ Polimorfismo
- ☐ Encapsulamento

#### 4) Considere o código abaixo:

```
01. for (int i = 0; i < 3; i++) {  
02.     for (int j = 1; j < 3; j++) {  
03.         if (i % j == 0) {  
04.             continue;  
05.         } else {  
06.             System.out.println("i = " + i + ", j = " + j);  
07.         }  
08.     }  
09. }
```

- ( continue ) i = 0, j = 1
- ( continue ) i = 0, j = 2
- ( continue ) i = 1, j = 1
- ( ocorrência de print ) i = 1, j = 2
- ( continue ) i = 2, j = 1
- ( continue ) i = 2, j = 2

#### 5) Que situação o código abaixo indica?

```
01. class Carro{  
02.     String cor = "branco";  
03.     void mudaCor(String cor){  
04.         this.cor = cor;  
05.     }  
06. }  
07.  
08. class Pessoa{  
09.     final Carro carro = new Carro();  
10.     void mudaCorDoCarro(String cor){  
11.         carro.mudaCor(cor);  
12.     }  
13. }
```

- ( ) Que uma pessoa pode trocar de carro, mas um carro não pode trocar de cor.
- ( ) Que uma pessoa não pode trocar de carro e não pode trocar a cor do carro.
- ( ) Que uma pessoa pode trocar de carro e pode trocar a cor do carro.
- ( X ) Que uma pessoa não pode trocar de carro, mas pode trocar a cor do carro.

**06) Quatro cientistas sentam-se a jantar. Os nomes são Shelly, Frank, Corbin e Mel. Os quatro colocam cartas na mesa com apenas os seus sobrenomes: Infinito, Radiano, Tissue, e Ósmio.**

Será capaz de descobrir os nomes completos dos cientistas, sabendo apenas que:

- Nenhum cientista tem um sobrenome em que apareça a inicial do primeiro nome;
- O sobrenome de Corbin é também um elemento;
- O primeiro nome de Radiano contém um R;

Corbin Ósmio, visto que ósmio é o único elemento.

Frank Radiano, uma vez que para o Radiano havia 2 opções e uma delas foi retirada acima.

Shelly Infinito, pois Tissue possui a inicial de Shelly.

Mel Tissue.

**7) Determine o próximo número da sequência: 5,11,19,29,41,...**

5 (+6) 11 (+8) 19 (+10) 29 (+12) 41 (+14) => **55**

**08) Para qual tarefa será mais apropriado usar o comando DISTINCT?**

- ☐ identificar linhas duplicadas na tabela
- ☐ identificar quais colunas possuem dados únicos
- ☐ eliminar colunas duplicadas na tabela
- ☒ eliminar linhas duplicadas no resultado

**09) Considerando a consulta abaixo,o que é mostrado na coluna SALARY quando um valor NULL é retornado?**

*SELECT name, NVL(salary, 0) FROM employee WHERE salary IS NULL ORDER BY name;*

- ☒ 0
- ☐ NULL
- ☐ Spaces
- ☐ nothing

**10) Para qual tarefa você precisará usar o operador BETWEEN?**

- ☐ Consulta de tabelas com valores desconhecidos.
- ☒ Consulta de tabelas para uma faixa de valores
- ☐ Consulta de tabelas para um tipo de caracter
- ☐ Consulta de tabelas para valores específicos de uma lista

## 11. Vida útil da bateria do laptop para jogos

Você comprou um novo laptop para jogos e gosta de jogar nele o dia todo. Requer carregamento frequente devido ao alto uso de gráficos. Você quer saber quanta bateria será deixada após vários eventos de reprodução e carregamento.

Escreva uma função que receba  $n$  registros do consumo de bateria do laptop e *eventos de carregamento*. Um valor de *eventos*  $[i]$  representa o número de minutos gastos carregando o laptop (valor positivo) ou jogando um jogo – consumo (valor negativo). A cada minuto, o laptop consome *1% de bateria*. A carga da bateria não pode exceder *100%*. Retorne a porcentagem de cobrança final do laptop, pois a carga inicial é de *50%*.

### Exemplo

$n = 4$

*eventos* =  $[10, -20, 61, -15]$

Inicialmente, o laptop é carregado com 50%. Ele está conectado a *eventos*  $[0] = 10$  minutos a carga final é  $50 + 10 = 60$ . Um jogo é jogado por *20 minutos em eventos*  $[1] = -20$  levando a carga para 40. O carregamento por outros 61 minutos faz com que a bateria atinge 100 e, em seguida, 15 minutos de jogo resultam em uma carga final de 85.

### Descrição da função

Complete a função *getBattery*. A função deve retornar um número inteiro.

*getBattery* possui o seguinte parâmetro:

***eventos* [*eventos*  $[0]$ ,... *eventos*  $[n-1]$ ]: uma matriz de números inteiros**

```
public Integer getBattery(ArrayList<Integer> events) {
    Integer battery = 50;
    int eventsSize = events.size();

    for (int i=0; i<eventsSize ; i++) {
        if (events.get(i) == null) {
            continue;
        }

        battery = battery + events.get(i);

        if (battery < 0) {
            battery = 0;
        } else if (battery > 100) {
            battery = 100;
        }
    }

    return battery;
}
```

**12.** Para  $A = V$ ,  $B = V$  e  $C = F$ , qual o resultado da avaliação das seguintes expressões:

a)  $(A \text{ e } B) \text{ ou } (A \text{ xou } B)$       **V**

b)  $(A \text{ ou } B) \text{ e } (A \text{ e } C)$       **F**

c)  $(A \text{ ou } C) \text{ e } (B \text{ xou } A) \text{ e } (\text{não } B)$       **F**