

QUESTÕES DA AULA 7

Elaborar 4 questões objetivas alinhadas com os 4 objetivos de aprendizagem da aula 7.

1- O professor André precisa criar um sistema que realize o cálculo do Índice de massa corpórea (IMC) para ser utilizado pelos profissionais da saúde do Centro de Educação e Pesquisa em Saúde Anita Garibaldi para verificar os pacientes que se encontram muito acima do peso normal, para isso ele elegeu você para criar a função do IMC, com isso, dentre as soluções abaixo qual possui a sintaxe (Na linguagem Python) correta para resolver esse problema:

a)

```
def Funcao_Imc():  
    condicao=1  
    resultado=[]  
    while(condicao):  
        peso=float(input('Insira seus peso '))  
        altura=float(input('Insira seus altura '))  
        imc= round(peso/math.pow(altura,2),2)  
        resultado.append(imc)  
        if (imc<17.0):  
            print ("Imc:",imc,"Muito abaixo do peso")  
        elif (imc>=17.0 and imc<18.5):  
            print ("Imc:",imc,"Abaixo do peso")  
        elif (imc>=18.5 and imc<25.0):  
            print ("Imc:",imc,"Peso dentro do normal")  
        elif (imc>=25 and imc<30):  
            print ("Imc:",imc,"Acima do peso normal")  
        elif (imc>=30):  
            print ("Imc:",imc,"Muito acima do peso normal")  
        condicao=int(input("Gostaria de fazer outra leitura, 1 para  
sim e 0 para não?"))  
    return resultado
```

b)

```
def Funcao_Imc():  
    condicao=1  
    resultado=[]  
    while(condicao)  
        peso=float(input('Insira seus peso '))
```

```

altura=float(input('Insira seus altura '))
imc= round(peso/altura)
resultado.append(imc)
if (imc<17.0):
    print ("Imc:",imc,"Muito abaixo do peso")
elif (imc>=17.0 and imc<18.5):
    print ("Imc:",imc,"Abaixo do peso")
elif (imc>=18.5 and imc<25.0):
    print ("Imc:",imc,"Peso dentro do normal")
elif (imc>=25 and imc<30):
    print ("Imc:",imc,"Acima do peso normal")
elif (imc>=30):
    print ("Imc:",imc,"Muito acima do peso normal")
condicao=int(input("Gostaria de fazer outra leitura, 1 para sim e 0 para
não?"))

```

```

c)
fuction Funcao_Imc():
condicao=1
resultado=[]
while(condicao)
    peso=float(input('Insira seus peso '))
    altura=float(input('Insira seus altura '))
    imc= round(peso/altura)
    resultado.append(imc)
    if (imc<17.0):
        print ("Imc:",imc,"Muito abaixo do peso")
    elif (imc>=17.0 and imc<18.5):
        print ("Imc:",imc,"Abaixo do peso")
    elif (imc>=18.5 and imc<25.0):
        print ("Imc:",imc,"Peso dentro do normal")
    elif (imc>=25 and imc<30):
        print ("Imc:",imc,"Acima do peso normal")
    elif (imc>=30):
        print ("Imc:",imc,"Muito acima do peso normal")
    condicao=int(input("Gostaria de fazer outra leitura, 1 para sim e 0 para
não?"))

```

```

    return resultado

```

```

d)
Funcao_Imc(){
condicao=1
resultado=[]
while(condicao)

```

```

    peso=float(input('Insira seus peso '))
    altura=float(input('Insira seus altura '))
    imc= round(peso/altura)
    resultado.append(imc)
    if (imc<17.0):
        print ("Imc:",imc,"Muito abaixo do peso")
    elif (imc>=17.0 and imc<18.5):
        print ("Imc:",imc,"Abaixo do peso")
    elif (imc>=18.5 and imc<25.0):
        print ("Imc:",imc,"Peso dentro do normal")
    elif (imc>=25 and imc<30):
        print ("Imc:",imc,"Acima do peso normal")
    elif (imc>=30):
        print ("Imc:",imc,"Muito acima do peso normal")
    condicao=int(input("Gostaria de fazer outra leitura, 1 para sim e 0 para
não?"))
    return resultado}
e)
fuction Funcao_Imc():
    condicao=1
    resultado=[]
    while(condicao)
        peso=float(input('Insira seus peso '))
        altura=float(input('Insira seus altura '))
        imc= round(peso/altura)
        resultado.append(imc)
        if (imc<17.0):
            print ("Imc:",imc,"Muito abaixo do peso")
        elif (imc>=17.0 and imc<18.5):
            print ("Imc:",imc,"Abaixo do peso")
        elif (imc>=18.5 and imc<25.0):
            print ("Imc:",imc,"Peso dentro do normal")
        elif (imc>=25 and imc<30):
            print ("Imc:",imc,"Acima do peso normal")
        elif (imc>=30):
            print ("Imc:",imc,"Muito acima do peso normal")
        condicao=int(input("Gostaria de fazer outra leitura, 1 para sim e 0 para
não?"))
    input(resultado)

```

2- Um professor do segundo semestre do curso de Neurociências pediu para seus alunos criarem uma função, na linguagem Python, que realize o cálculo da diferença de potencial entre a parte intra e extracelular de um neurônio através da equação de Nernst (Figura 1). Sabendo disso, qual das opções abaixo é a melhor escolha para realizar a atividade pedida pelo professor:

Figura 1 : Fórmula de Nernst

Equação de Nernst

$$V_K = \frac{RT}{zeA} \ln \left(\frac{[\text{Íon}]_{\text{fora}}}{[\text{Íon}]_{\text{dentro}}} \right)$$

Fonte: Azevedo, 2017.

a)

```
import math
function LeiNernst(){
    T=float(input("Digite a temperatura absoluta:"))
    z=float(input("Digite a valência do íon:"))
    e=float(input("Digite a carga elétrica do elétron:"))
    IONf=float(input("Digite a concentração do íon fora:"))
    IONd=float(input("Digite a concentração do íon dentro:"))
    R=8.315
    A=6.02 * 10^(23)
    V= math.sqrt (math.pow(R*T/z*e*A,2) +math.pow(IONf/IONd,4)) }
```

b)

```
import math
def LeiNenst(T, z, e, IONf, IONd):
    R=8.315
    A=6.02 * 10^(23)
    V=((R*T)/(z*e*A)) * math.log10( IONf/IONd)
    return V
```

c)

```
import csv
```

```
def LeiNenst(T, z, e, IONf, IONd):
    R=8.315
    A=6.02 * 10^(23)
    V=((R*T)/(z*e*A)) * csv.log10( IONf/IONd)
    return V
```

d)

```
import math
def LeiNenst(T, z, e, IONf, IONd):
    R=8.315
    A=6.02 * 10^(23)
    V= math.sqrt (math.pow(R*T/z*e*A,2) +math.pow(IONf/IONd,4)) }
```

e)

```
import math
function LeiNenst(T, z, e, IONf, IONd){
    R=8.315
    A=6.02 * 10^(23)
    V=((R*T)/(z*e*A)) * math.log10( IONf/IONd)
    return V
}
```

3- Sabe-se que as classes são de suma importância quando se está criando sistemas computacionais, pois facilita a revisão e ajustes do código. Além disso, são imprescindíveis quando forem aplicadas em sistemas para área da neurociências, pois esse código precisa ser entendido por profissionais da áreas da saúde, bem como das exatas, isto é, precisando cada vez mais de organização. Com isso, dentre as afirmativas, abaixo, sobre as classes está **correta**:

- a) Uma classe podem ser consideradas como um conjunto de instruções, as quais realizam uma tarefa específica;
- b) Uma classe pode ser conhecido como uma estrutura que tem por único objetivo descrever os atributos, isto é, as propriedades que o objeto terá e não levando em consideração seu comportamento.
- c) Uma classe pode ser conhecido como uma estrutura que tem por objetivo descrever os atributos, isto é, as propriedades que o objeto terá, e também levando em consideração seu comportamento.**
- d) De acordo com o paradigma de programação orientada a objeto, a classe pode ser considerada como um conjunto de atributos que compartilham de um mesmo objeto.
- e) Programação imperativa considera a classe é um conjunto de objetos que compartilham os mesmos atributos.

4- o professor André quer fazer uma sistema de controle de um dispositivo de optogenética para isso ele quer fazer uma classe que define os atributos, os quais

são as intensidades de luz RGB e uma matriz que contém 32 eletrodos, de um dispositivo. Como isso, o qual das opções abaixo possui a sintaxe (na linguagem Python) mais correta para a realização uma classe para o dispositivo em questão.

a) **class Dispositivo :**

```
def __init__(self,Matriz-Eletrodos,RGB_red,RGB_green,RGB_blue):  
    self.Matriz-Eletrodos=Matriz-Eletrodos  
    self.RGB_red=RGB_red  
    self.RGB_green=RGB_green  
    self.RGB_blue=RGB_blue
```

b) **class Dispositivo (self,Matriz-Eletrodos,RGB_red,RGB_green,RGB_blue):**

```
print(Matriz-Eletrodos)  
print(RGB_red)  
print(RGB_green)  
print(RGB_blue)
```

c) **class Dispositivo (self,Matriz-Eletrodos,RGB_red,RGB_green,RGB_blue):**

```
Matriz-Eletrodos=list(input("Digite a matriz de letrodos"))  
RGB_red=int(input("Digite a intensidade do RGB da cor vermelha"))  
RGB_green=int(input("Digite a intensidade do RGB da cor verde"))  
RGB_blue=int(input("Digite a intensidade do RGB da cor azul"))
```

d) **def Dispositivo :**

```
def __init__(self,Matriz-Eletrodos,RGB_red,RGB_green,RGB_blue):  
    self.Matriz-Eletrodos=Matriz-Eletrodos  
    self.RGB_red=RGB_red  
    self.RGB_green=RGB_green  
    self.RGB_blue=RGB_blue
```

e) **class Dispositivo :**

```
def __init__(self,Matriz-Eletrodos,RGB_red,RGB_green,RGB_blue)  
self.Matriz-Eletrodos=Matriz-Eletrodos  
self.RGB_red=RGB_red  
self.RGB_green=RGB_green  
self.RGB_blue=RGB_blue
```