



# Lógica de Programação Orientada a Objetos

# Exercício 1

---

```
def function_1(a,b):  
    a.append(b)  
    return XXXXXX
```

O que deveríamos colocar no lugar de 'XXXXXX', para que a função retornasse a quantidade de elementos em a?

- A. count(a)
- B. a.count()
- C. a.len()
- D. len(a)

# Exercício 2

---

```
Def remove(m,n, lista):  
    for i in range (m,n+1):  
        XXXX
```

A função acima remove os elementos das posições m até n (inclusive). Com essas informações, o que deveríamos substituir por XXXX para que a função funcionasse corretamente

- A. `del(i)`
- B. `del(n)`
- C. `del(n+1)`
- D. `del(m)`

# Exercício 3

---

Class prova:

```
def __init__(self,nome):
    self.aluno = nome
    self.nota = 0
    self.questoes_corrigidas = [] # para cada questão se adicionará 1 para certo e 0 para erradas

def corrigir(self):
    self.nota = self.questoes_corrigidas.sum()
    print("O aluno", self.aluno, "tirou nota",self.nota)
```

Qual o nome técnico correto para se referir as variáveis “aluno”, “nota” e “questões” da classe prova ?

- A. variáveis
- B. atributos
- C. métodos
- D. funções

# Exercício 4

---

Class prova:

```
def __init__(self,nome):  
    self.aluno = nome  
    self.nota = 0  
    self.questoes_corrigidas = [] # para cada questão se adicionará 1 para certo e 0 para erradas  
  
def corrigir(self):  
    self.nota = self.questoes_corrigidas.sum()  
    print("O aluno", self.aluno, "tirou nota",self.nota)
```

Qual o nome técnico correto para se referir a função “corrigir” da classe prova ?

- A. variável
- B. atributo
- C. método
- D. função

# Exercício 5

---

```
## a função fará via soma a multiplicação 2*k
### Por exemplo se k=3, a função deverá retornar 6
def tabuada_do_2(k):
    if k < 1:
        return 0
    return 2+tabuada_do_2(k-1)
```

Qual o nome dado para o tipo de função que no seu corpo (escopo), chama a ela mesma. Como por exemplo a função `tabuada_do_2`, acima?

- A. construtora
- B. recursiva
- C. método
- D. função

# Exercício 6

---

Def remove\_retorna(n, lista):

    return XXXX

A função acima, deve remover o elemento da posição 'n' da 'lista' e o retornar. Com essas informações, o que deveríamos substituir por XXXX para que a função funcionasse corretamente

- A. del(lista[n])
- B. remove(lista[n])
- C. lista.pop(n)
- D. lista.remove(n)

# Exercício 7

---

```
try:
    print('Linha 1')
    ### linha A
    try:
        print('Linha 2')
        ### linha B
    except:
        ### linha C
        print('Linha 3')
    print('Linha 4')
except:
    print('Linha 5')
print('Linha 6')
```

Dado a saída do código acima, em que lugar (caso necessário), devemos adicionar `raise RuntimeError('Erro 1')` para que o `print('Linha 3')` apareça na saída (caso ele não esteja aparecendo ainda).

- A. linha A
- B. linha B
- C. linha C
- D. Não é necessário adicionar nada



# Exercício 8

---

Crie uma função com 3 argumentos que representarão os lados de um triângulo. Considere que não importa a ordem que serão fornecidos os valores, podendo ser fornecido primeiro a hipotenusa e depois os catetos, ou primeiro os catetos e depois a hipotenusa, etc. O retorno da função é uma lista/tupla com os lados informados na ordem [hipotenusa, cateto1, cateto2] e um booleano que informa se o triângulo é retângulo ou não.

Você pode utilizar o teorema de Pitágoras para auxiliar na resolução:  $hipotenusa^2 = cateto1^2 + cateto2^2$ .

# Exercício 9

---

Crie uma classe que modele uma pessoa:

Atributos: nome, idade, peso e altura

Métodos: Envelhecer, engordar, emagrecer, crescer.

O programa deve pedir o nome da pessoa e ano e calcular sua altura com base na seguinte informação:

A cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm.

# Exercício 10

---

```
class Felinos:
    sound = 'meow'
    def make_sound(self):
        print('Gatos fazem',self.sound)

class Gato(Felinos):
    def __init__(self):
        self.sound = 'meow '
    def make_sound(self,n=1):
        som = lambda x,k: x*k
        print('Gatos fazem',som(self.sound,n))
```

Utilizando o código acima, crie o objeto do tipo Gato e o chame a função make\_sound(), de modo a obter (sem modificar nada), as seguintes saídas:

- Gatos fazem meow
- Gatos fazem meow meow meow meow meow
- Gatos fazem