AC1 - Programação Orientada a Objetos

Seu e-mail será registrado quando você enviar este formulário.

*Obrigatório

Selecione a opção que contém os nomes corretos dos operadores aritméticos binários +, -, *, /, //, % e **, respectivamente:*

1 ponto

adição, subtração, multiplicação, divisão inteira, divisão real, exponenciação e resto.

adição, subtração, exponenciação, divisão real, divisão inteira, resto e multiplicação.

adição, subtração, multiplicação, divisão real, divisão inteira, resto e exponenciação.

adição, subtração, multiplicação, divisão inteira, divisão real, resto e exponenciação.

adição, subtração, multiplicação, divisão real, divisão inteira, exponenciação e resto.

Marque a alternativa correta sobre funções em Python.*

1 ponto

Funções não devem ser reaproveitadas em diferentes pontos do código.

São estruturas de dados lineares.

São a forma de criar comportamentos para nosso software, que podem ser reaproveitados em diferentes contextos.

Funções não podem chamar outras funções.

Funções não podem ter variáveis internas.

Marque a alternativa correta sobre listas em Python.*

1 ponto

Elementos contidos em listas NÃO podem ser acessados diretamente (nem pelo índice).

Listas são imutáveis.

Listas são estruturas de dados não lineares (sem ordem garantida).

Listas são estruturas de dados lineares, isto é, coleções sequenciais cuja ordem (posição na lista) é garantida e cada elemento pode ser acessado através de seu índice).

Listas são pares chave-valor.

Qual a estrutura de mapeamento (pares chave-valor) do Python?*

1 ponto

Não é possível trabalhar com pares chave-valor em python.

Vetor sequencial.

Lista.

Função.

Dicionário.

Selecione a saída correta para o código Python abaixo:*

1 ponto

```
valores = [1, 2, 3, 4]
novo = []

for n in valores:
    novo.append(n**2)

print(novo)
```

[1, 2, 3, 4, 1, 4, 9, 16]

[1, 2, 3, 4]

[4, 3, 2, 1]

[1, 4, 9, 16]

[2, 4, 6, 8]

Selecione a saída correta para o código Python abaixo:*

1 ponto

```
### Funções ###
def eh_palindromo(texto):
    for i in range(len(texto)):
        if texto[i] != texto[-1-i]:
            return False
    return True

### Código principal ###
entradas = ['arara', 'elefante', 'radar', 'banana']
palindromos = []

for palavra in entradas:
    if(eh_palindromo(palavra)):
        palindromos.append(palavra)

print(palindromos)
```

None

['arara', 'elefante', 'radar', 'banana']

[ ]

['arara', 'radar']

['arara']

]]

Marque a alternativa que NÃO É um dos 4 pilares de Programação Orientada a Objetos (POO)*

1 ponto

Herança

Diagramação

Abstração

Polimorfismo

Encapsulamento

Marque a alternativa FALSA em relação às *classes* de POO em Python que são criadas pelo programador.*

1 ponto

Representam a abstração de um conjunto de objetos, definindo quais são as especificações em comum destes objetos.

Definem um tipo personalizado de dado, e são usadas para criar objetos desse tipo.

Em Python, possuem um método especial chamado __init__, que é responsável por inicializar os objetos que serão criados a partir dela.

São criadas em Python com o uso da palavra chave class.

Ao criarmos uma classe nova, devemos informar previamente na definição da classe o número total de objetos que serão criados a partir dela.

Marque a alternativa FALSA em relação aos *objetos* de POO em Python que são criados pelo programador.*

1 ponto

são entidades que agrupam apenas características de um objeto real, não sendo possível a representação de comportamentos ou ações.

são mutáveis, isto é, após criados, podem ser modificados. Por exemplo, é possível adicionar um novo atributo a um objeto já criado.

são entidades que agrupam características e comportamentos, representando no programa objetos do mundo real.

são criados em tempo de execução, ao instanciarmos uma classe.

normalmente chamamos as características de atributos e os comportamentos de métodos de um objeto.

Avalie o trecho de código a seguir e marque a afirmação VERDADEIRA*

1 ponto

```
class Aluno:
    pass

aluno = Aluno()
aluno.nome = 'Megan'
```

Cria uma classe Aluno, que recebe um nome no inicializador (__init__) e em seguida cria um objeto a partir dessa classe já com o nome 'Megan'

Cria um objeto Aluno, cria uma classe Megan e cria uma instância dessa classe com o nome "aluno".

Cria uma classe Aluno, instancia um objeto dessa classe e em seguida cria um atributo 'nome' neste objeto com o valor 'Megan'.

Cria uma classe Aluno, cria uma nova classe que herda de Aluno e atribui o valor 'Megan' ao atributo 'nome'.

Cria uma classe Aluno, instancia um objeto dessa classe e em seguida exibe na tela o nome do aluno.