

Questões aula 07 – Alexandre Chaves Fernandes

Objetivo de aprendizado trabalhado na questão 01: Definir as estruturas necessárias para criar funções

- 1) A neuroengenharia desenvolve várias tecnologias que podem ser úteis, como por exemplo dispositivos de neuromodulação. Esses dispositivos precisam ser testados em estudos clínicos, em especial os ensaios clínicos randomizados. Nesses estudos basicamente existem dois grupos, um controle e um outro no qual será testada uma nova intervenção. Uma etapa importante no ensaio clínico é o processo de coleta e análise dos dados. Essa é uma etapa fundamental, uma vez que a segurança dos dados e a análise correta dos dados são etapas fundamentais. Para gerir esse fluxo de dados, pode ser utilizado softwares que utilizam funções, tal como parte do código abaixo. Sobre as funções podemos afirmar que:

```
7
8 def coleta_dados():
9     Paciente = input("Digite o nome do paciente")
10    Codigo_paciente = int(input("Digite o código do paciente"))
11    D0 = int(input("Digite o nível de dor do paciente pré tratamento"))
12    D1 = int(input("Digite o nível de dor do paciente pós tratamento "))
13    Grupo = int(input("Digite se pertence ao grupo 01 ou 02"))
14
15
16    with open('dados de neuromodulação com pandas.csv', 'a', newline= '') as csvFile: #Salvar em csv os dados coletados
17        spamWriter= csv.writer(csvFile, delimiter= ' ', quoting= csv.QUOTE_MINIMAL)
18
19        spamWriter.writerow ([Paciente] + [Codigo_paciente] + [Grupo] + [D0] + [D1])
20
21
22
23 def analise_dados():
24     grupo = 4
25     while grupo != 1 or 2 or 3:
26         grupo = int(input("Você gostaria de ver os dados do intragrupo 01, intragrupo 02 ou todos os dados?"))
27
28         df = pd.read_csv("dados de neuromodulação com pandas.csv", encoding = "UTF-8", sep = " ")
29
30         if grupo == 1:
31             df_grupo1 = df[df.Grupo == 1] # Filtrando o grupo 01
32             dados_tratamento_grupo_1 = df_grupo1["Dor_pré_tratamento"] #Dados de pré tratamento do grupo 01 que foi filtrado
33             dados_pos_tratamento_grupo_1 = df_grupo1["Dor_pós_tratamento"] #Dados do pós tratamento do grupo 01 que foi filtrado
34
```

- A) A função def coleta_dados não irá rodar, uma vez que não possui variável dentro do parêntese após o nome coleta_dados
- B) As funções são importantes porque organizam o código e podem ser reutilizadas ao longo do código
- C) Não existe na prática diferença em quebrar o programa em unidades menores como as funções.
- D) A função def analise_dados não precisará ser chamada para executar a ação
- E) A função def analise_dados só irá rodar pois apresenta no seu início um parâmetro de entrada de dados

Objetivo de aprendizado trabalhado na questão 02: Reescrever programas na forma de funções

- 2) A caixa de condicionamento operante é útil em uma série de estudos em neurociências, especialmente estudos comportamentais. Uma possibilidade de estruturação de programa para uma caixa de condicionamento operante é a exposta na imagem abaixo. Sobre as possibilidades de escrita desse código e as funções, podemos afirmar que:

```

1 # Habituação
2 print ("## HABITUAÇÃO ##")
3 habituacao_valor = "nao"
4 while habituacao_valor != "sim":
5     habituacao_valor = input("Foi realizada as sessões de habituação no animal? (Responda 'sim' ou 'nao', sem as aspas)")
6     if habituacao_valor == "sim":
7         print ("\nAnimal habituado. Iniciar as sessões de treinamento")
8         sessao_habitacao = True
9     else:
10        print ("\nrealizar a etapa de habituação")
11        sessao_habitacao = False
12
13 # Regime de aproximações sucessivas
14 if sessao_habitacao == True:
15     print ("\n## REGIME DE APROXIMAÇÕES SUCESSIVAS ##")
16     aproximacao = 30
17     while aproximacao >= 30:
18         aproximacao = float(input("Qual a aproximação do animal em cm? "))
19         if aproximacao < 30:
20             print ("\nDar 0,5 ml de recompensa ao animal")
21             animal_aproximado = True
22         else:
23             print ("\nAguardar a aproximação ficar menor que 30 cm")
24             animal_aproximado = False
25
26 # TOCAR NA BARRA 20 x PARA O ANIMAL PASSAR PARA A PRÓXIMA ETAPA. O CÓDIGO TEM UM TOTAL DE 50 REPETIÇÕES PARA TENTAR Atingir OS 20 TOQUES
27 if animal_aproximado == True:
28     numero_toques = 0
29     while numero_toques < 20:
30         tocar = input("O animal tocou na barra? (Responda 'sim' ou 'nao', sem as aspas)")
31         if (tocar == "sim"):
32             print ("Dar 0,5 ml de recompensa ao animal")
33             numero_toques = numero_toques + 1
34         if numero_toques == 20:
35             print ("\n## O experimento passou para a próxima etapa ##")
36             toques = True
37
38 # RECOMPENSA COM O LADO DO SOM #
39
40 if toques == True:
41     repeticoes_experimento = 0
42     while repeticoes_experimento <= 50:
43         som = int(input("O som emitido foi o 1 ou 2 ?(Responda '1' ou '2', sem as aspas)"))
44         lado = input("Qual lado da barra o animal tocou? (Responda 'direita' ou 'esquerda', sem as aspas)")
45         if som == 1 and lado == "esquerda":
46             print ("Dar 0,5 ml de recompensa ao animal")
47             repeticoes_experimento = repeticoes_experimento + 1
48         elif som == 2 and lado == "direita":
49             print ("Dar 0,5 ml de recompensa ao animal")
50             repeticoes_experimento = repeticoes_experimento + 1
51         else:
52             repeticoes_experimento = repeticoes_experimento + 1
53             print ("Não liberar nenhuma recompensa")
54
55 # Tempo e repetições do experimento
56 if repeticoes_experimento == 50:
57     duracao = int(input("Qual foi a duração do experimento em minutos?"))
58     if duracao <= 30 and repeticoes_experimento >= 50:
59         print ("O experimento seguirá para a próxima fase!")
60

```

- A) Não daria certo utilizar funções para esse programa
- B) O programa poderia ser escrito novamente com funções para cada fase
- C) No caso de ter função, não seria verdade que uma variável declarada dentro de uma função somente existirá durante a execução desta mesma função
- D) No caso de ter função, não seria verdade que uma segunda chamada da mesma função, o programa não se lembrará do valor anterior para as mesmas variáveis
- E) Só seria possível definir uma função se ela tiver parâmetro de entrada ou saída

Objetivo de aprendizado trabalhado na questão 03: Definir classe e sua aplicação

- 3) A optogenética é uma técnica cada vez mais utilizada devido a sua alta precisão temporal e espacial, propiciando um estudo eficiente sobre estruturas e vias neurais. Na optogenética se utiliza luz led de um RGB. Um software para gerenciar um experimento em optogenética e o funcionamento dos Led é complexo e para isso requer a criação de entes chamados “classes”. Sobre as classes no contexto de programação, podemos afirmar que:

- A) A classe não é importante na programação orientada a objetos
- B) A classe não descreve as propriedades e atributos que um determinado objeto terá
- C) A classe é um padrão para definição de objetos
- D) A classe não descreve qual o comportamento de objetos da classe
- E) Na programação procedural o foco é a escrita de classes, e não de funções

Objetivo de aprendizado trabalhado na questão 04: Reconhecer estruturas da linguagem que utilizam classe

- 4) Na neuroengenharia frequentemente temos que saber definir classes para resolver os problemas que aparecem. Abaixo segue a definição de uma classe. Sobre essa programação e as classes em programação podemos afirmar que:

```
class Point:
    def __init__(self, initX, initY):
        self.x = initX
        self.y = initY

    def getX(self):
        return self.x

    def getY(self):
        return self.y
```

```
def distanceFromOrigin(self):
    return ((self.x ** 2) + (self.y ** 2)) ** 0.5

def __str__(self):
    return "x=" + str(self.x) + ", y=" + str(self.y)

def halfway(self, target):
    mx = (self.x + target.x)/2
    my = (self.y + target.y)/2
    return Point(mx, my)
```

- A) Não existe utilidade em definir a classe point. Tal definição torna a programação mais complexa que o necessário, sem ganhos
- B) O programa não irá rodar, pois não se deve escrever uma função dentro de uma classe
- C) A classe defini objetos, e o objeto possui um estado e uma coleção de métodos que ele pode executar
- D) O comando self não é importante
- E) Um objeto é um elemento computacional que representa apenas entidade abstratas

GABARITO: 1 – B // 2 – B // 3 – C // 4 - C