Situação	Finalizada
Iniciado	segunda-feira, 1 jul. 2024, 23:26
Concluído	terça-feira, 2 jul. 2024, 00:32
Duração	1 hora 5 minutos
Notas	23,00/23,00
Nota	**10,00** de um máximo de 10,00(**100**%)

Questão 1

Correto

Atingiu 2,00 de 2,00

Complete o código abaixo: (Aula_08_4a_App_Toplevel_OOP.py)

Arraste quais são as alternativas corretas.

Janela_Top.py

```
class Janela_Top( [ Toplevel ] ✓ ):
    def [ __init__ ] ✓ (self, Master, Str="Janela", px=0, py=0,
            dx=640, dy=480, cor= [ "ligthgray" ] ✓ ):
        super().__init__(Master)
        [ super() ] ✓ .title(Str)
        super().[ geometry ] ✓ ("%dx%d+%d+%d" % (dx, dy, px, py))
        super().[ configure ] ✓ (bg=cor)
        super().[ grab_set ] ✓ ()
        self.[ inicialize ] ✓ ()

    def action_exit(self):
        print("Destruindo a janela 2...")
        self.[ destroy() ] ✓

    def inicialize( [ self ] ✓ ):
        Lb1=Label(self, text="Nome")
        Et1=Entry(self, [ width=32 ] ✓ )
        Lb1.[grid(row=0, column=0, sticky=W, padx=2,]✓pady=2)
        Et1.[grid(row=1, column=0, sticky=W, padx=4,]✓pady=4)
        self.protocol("WM_DELETE_WINDOW", [ self.action_destroy ] ✓ )
```

Sua resposta está correta.

A resposta correta é:

Complete o código abaixo: (Aula_08_4a_App_Toplevel_OOP.py)

Arraste quais são as alternativas corretas.

Janela_Top.py

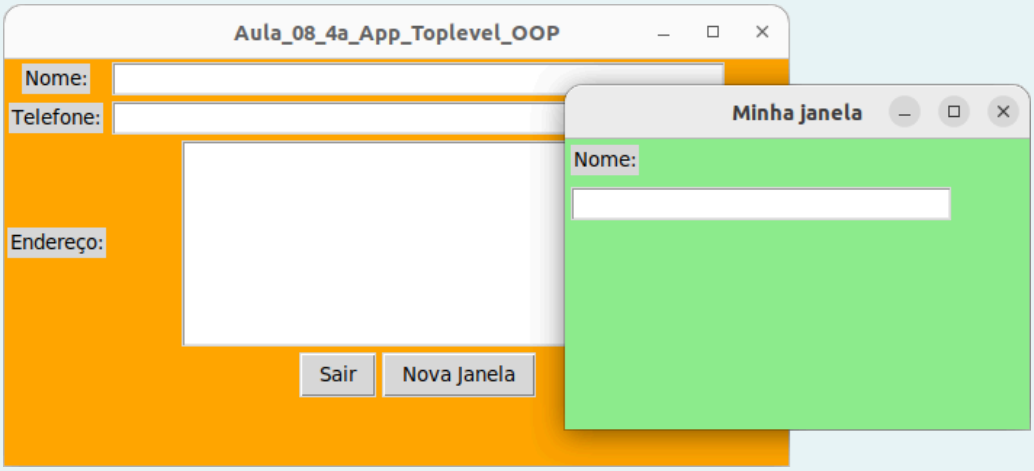
```
class Janela_Top([Toplevel]):
    def [__init__](self, Master, Str="Janela", px=0, py=0,
               dx=640, dy=480, cor=["ligthgray"]):
        super().__init__(Master)
        [super()].title(Str)
        super().[geometry]("%dx%d+%d+%d" % (dx, dy, px, py))
        super().[configure](bg=cor)
        super().[grab_set]()
        self.[inicialize]()

    def action_exit(self):
        print("Destruindo a janela 2...")
        self.[destroy()]

    def inicialize([self]):
        Lb1=Label(self, text="Nome")
        Et1=Entry(self, [width=32])
        Lb1.[grid(row=0, column=0, sticky=W, padx=2, pady=2)]
        Et1.[grid(row=1, column=0, sticky=W, padx=4, pady=4)]
        self.protocol("WM_DELETE_WINDOW", [self.action_destroy])
```

Questão 2

Correto

Atingiu 2,00 de 2,00

Complete o código abaixo:



O programa é continuação do programa apresentado anteriormente.
Arraste quais são as alternativas corretas.

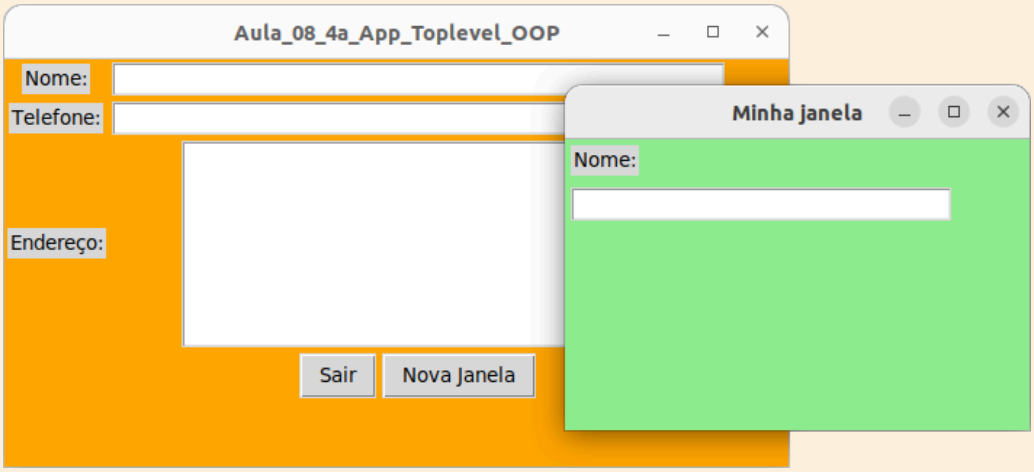
Janela.py

```
from Janela_Top import Janela_Top

class Janela( Tk ✓ ):
    def __init__(self, Str="Janela", px=0, py=0, dx=640, dy=480 ✓ , cor="ligthgray"):
        super(). __init__() ✓
        super().title( Str ✓ )
        super().geometry("%dx%d+%d+%d" % (dx, dy, px, py))
        super().configure(bg=cor)
        self. inicialize ✓ ()

    def action_new_window(self):
        Jan2 = Janela_Top(self, "Minha janela", 500, 350, 320, 200, "lightgreen" ✓ )

    def action_exit(self):
        self.destroy()
        sys.exit(0)

    def inicialize(self):
        Lb1=Label(self, text="Nome:")
        Lb2= Label(self, text="Telefone:") ✓
        Lb3=Label(self, text="Endereço:")

        Et1=Entry(self, width=52)
        Et2= Entry(self, width=52) ✓

        Txt1=Text(self, height=8, width=40) ✓

        Bt1=Button(self, text='Sair', command=self.action_exit)
        Bt2=Button(self, text='Nova Janela', command= self.action_new_window ✓ )

        Lb1.grid(row=0, column=0, padx=2, pady=2)
        Lb2.grid(row=1, column=0, padx=2, pady=2)
        Lb3.grid(row=3, column=0, padx=2, pady=2) ✓
        Et1.grid(row=0, column=1, columnspan=2, padx=2, pady=2)
        Et2.grid(row=1, column=1, columnspan=2, padx=2, pady=2)
        Txt1.grid(row=3, column=1, columnspan=2, padx=2, pady=2)
        Bt1.grid(row=4, column=1, sticky=E, padx=2, pady=2)
        Bt2.grid ✓ (row=4, column=2, sticky=W, padx=2, pady=2)
        self.protocol("WM_DELETE_WINDOW", self.action_exit ✓ )
```

Sua resposta está correta.

A resposta correta é:

Complete o código abaixo:



O programa é continuação do programa apresentado anteriormente.
Arraste quais são as alternativas corretas.

Janela.py

```
from Janela_Top import Janela_Top

class Janela([Tk]):
    def __init__(self, Str="Janela", [px=0, py=0, dx=640, dy=480], cor="ligthgray"):
        super().[__init__()]
        super().title([Str])
        super().geometry("%dx%d+%d+%d" % (dx, dy, px, py))
        super().configure(bg=cor)
        self.[inicialize]()

    def action_new_window(self):
        Jan2 = Janela_Top(self, "Minha janela", [500, 350, 320, 200, "lightgreen"])

    def action_exit(self):
        self.destroy()
        sys.exit(0)

    def inicialize(self):
        Lb1=Label(self, text="Nome:")
        Lb2=[Label(self, text="Telefone:")]
        Lb3=Label(self, text="Endereço:")

        Et1=Entry(self, width=52)
        Et2=[Entry(self, width=52)]

        Txt1=[Text(self, height=8, width=40)]

        Bt1=Button(self, text='Sair', command=self.action_exit)
        Bt2=Button(self, text='Nova Janela', command=[self.action_new_window])

        Lb1.grid(row=0, column=0, padx=2, pady=2)
        Lb2.grid(row=1, column=0, padx=2, pady=2)
        Lb3.[grid(row=3, column=0, padx=2, pady=2)]
        Et1.grid(row=0, column=1, columnspan=2, padx=2, pady=2)
        Et2.grid(row=1, column=1, columnspan=2, padx=2, pady=2)
        Txt1.grid(row=3, column=1, columnspan=2, padx=2, pady=2)
        Bt1.grid(row=4, column=1, sticky=E, padx=2, pady=2)
        [Bt2.grid](row=4, column=2, sticky=W, padx=2, pady=2)
        self.protocol("WM_DELETE_WINDOW", [self.action_exit])
```

Questão 3

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:

Programa: Aula_10_4_App_TkTable

Dados.py

```
001... ######################################################
002...
003... Dados = [
004...             ['01', 'Antonio', '18', '2019'],
005...             ['02', 'Beatriz', '19', '2018'],
006...             ['03', 'Carlos', '20', '2017'],
007...             ['04', 'Darlan', '17', '2020'],
008...             ['05', 'Eduardo', '21', '2016'],
009...             ['06', 'Fernando', '20', '2015'],
010...             ['07', 'Guilherme', '19', '2017'],
011...             ['08', 'Humberto', '18', '2018'],
012...             ['09', 'Isabel', '17', '2021'],
013...             ['10', 'Joao', '21', '2016'],
014...             ['11', 'Karen', '23', '2015'],
015...             ['12', 'Luciana', '19', '2018'],
016...             ['13', 'Maria', '18', '2020'],
017...             ['14', 'Nicole', '19', '2019']
018...         ]
019...
020... ######################################################
```

Janela.py

```
021... import sys
022... from tkinter import *
023... from tkinter import messagebox
024... from Dados import Dados
025...
026... ################################################
027...
028... class Janela(Tk):
029...     Cnv1 = None
030...     inter = None
031...     inter_id = None
032...     Matriz = None
033...
034...     def __init__(self, Str="Janela", px=0, py=0, dx=640, dy=480, cor="ligthgray"):
035...         super().__init__()
036...         super().title(Str)
037...         super().geometry("%dx%d+%d+%d" % (dx, dy, px, py))
038...         super().configure(bg=cor)
039...         self.inicialize()
040...
041...     def action_exit(self):
042...         print("Destruindo a janela...")
043...         self.destroy()
044...         sys.exit(0)
045...
046...     def action_print(self):
047...         ######################################
048...         ##                                  ##
049...         ##         Complete o código        ##
050...         ##             do evento            ##
051...         ##                                  ##
052...         ######################################
053...
054...     def calculo_percentual(self):
055...         total=0
056...         Lb_height = self.Lb_cod.winfo_height() + 2 + 2
057...         print("Label height: %f" % Lb_height)
058...         total += Lb_height
059...         for row in self.Matriz:
060...             if (len(row)>0):
061...                 cell = row[0]
062...                 Et_height = cell.winfo_height()+2+2
063...                 total+=Et_height
064...
065...         Cnv_height = self.Cnv1.winfo_height()
066...         perc = (total-Cnv_height) / total
067...
068...         print("Canvas height: %f" % Cnv_height)
069...         print("Total: %f" % total)
070...         print("Percentual: %f %%" % perc)
071...         return(perc)
072...
073...     # track changes to the canvas and frame width and sync them,
074...     # also updating the scrollbar
075...     def _configure_interior(self, event):
076...         # update the scrollbars to match the size of the inner frame
077...         print("_configure_interior: %s" % event)
078...         posy = -event.y / event.height
079...         print("moveTo: %f %%" % posy)
080...
081...         perc = self.calculo_percentual()
082...
083...         if (posy < 0.0):
084...             self.Cnv1.yview_moveto(0.0)
```

```
085...         if (posy > perc):
086...             self.Cnv1.yview_moveto(perc)
087...
088...         size = (self.inter.winfo_reqwidth(), self.inter.winfo_reqheight())
089...         self.Cnv1.config(scrollregion="0 0 %s %s" % size)
090...         if self.inter.winfo_reqwidth() != self.Cnv1.winfo_width():
091...             # update the canvas's width to fit the inner frame
092...             self.Cnv1.config(width=self.inter.winfo_reqwidth())
093...
094...     def _configure_canvas(self, event):
095...         if self.inter.winfo_reqwidth() != self.Cnv1.winfo_width():
096...             # update the inner frame's width to fill the canvas
097...             self.Cnv1.itemconfigure(self.inter_id, width=self.Cnv1.winfo_width())
098...
099...     def inicialize(self):
100...         self.Frm1 = Frame(self, width=100, height=80, bd=4)
101...         self.Frm1.configure(bg='cyan')
102...
103...         self.Cnv1 = Canvas(self.Frm1, width=300, height=260, confine=False,
104...                            yscrollincrement=10, scrollregion=(0, 0, 300, 260))
105...         self.Cnv1.configure(bg='yellow', scrollregion="0 0 200 160")
106...         self.Cnv1.xview_moveto(0)
107...         self.Cnv1.yview_moveto(0)
108...
109...         self.inter = Frame(self.Cnv1)
110...         self.inter_id = self.Cnv1.create_window(0, 0, window=self.inter, anchor=NW);
111...
112...         self.inter.bind('<Configure>', self._configure_interior)
113...
114...         self.Cnv1.bind('<Configure>', self._configure_canvas)
115...
116...         Sb1 = Scrollbar(self.Frm1, orient="vertical")
117...
118...         self.Frm2 = Frame(self, width=100, bd=4)
119...         self.Frm2.configure(bg='red')
120...
121...         self.Lb_cod = Label(self.inter, text="Cod.")
122...         ########################################
123...         ##    Alocar os demais componentes    ##
124...         ########################################
125...
126...         self.Lb_cod.configure(bg='yellow', anchor=W)
127...         Lb_nome.configure(bg='yellow')
128...         Lb_idade.configure(bg='yellow')
129...         Lb_ingresso.configure(bg='yellow')
130...
131...         self.Lb_cod.grid(row=0, column=0, sticky=W, padx=2, pady=2)
132...         Lb_nome.grid(row=0, column=1, sticky=W, padx=2, pady=2)
133...         Lb_idade.grid(row=0, column=2, sticky=W, padx=2, pady=2)
134...         Lb_ingresso.grid(row=0, column=3, sticky=W, padx=2, pady=2)
135...
136...         ########################################
137...         ##                                    ##
138...         ##                                    ##
139...         ##                                    ##
140...         ##                                    ##
141...         ##    Preencha os campos da tabela    ##
142...         ##                                    ##
143...         ##                                    ##
144...         ##                                    ##
145...         ##                                    ##
146...         ########################################
147...
148...         Sb1.set(0, 10)
```

```
149...         Sb1.config(command=self.Cnv1.yview)
150...         self.Cnv1.config(yscrollcommand=Sb1.set)
151...
152...         Bt_print = Button(self.Frm2, text='Print', anchor='e', command=self.action_print)
153...         Bt_exit = Button(self.Frm2, text='Exit', anchor='w', command=self.action_exit)
154...
155...         self.Frm1.grid(row=0, column=0)
156...         self.Frm2.grid(row=1, column=0)
157...
158...         self.Cnv1.grid(row=0, column=0)
159...         Sb1.grid(row=0, column=ncol, rowspan=nlin, sticky=NS)
160...         Bt_print.grid(row=0, column=0, sticky=E, padx=8, pady=2)
161...         Bt_exit.grid(row=0, column=1, sticky=W, padx=8, pady=2)
162...
163...         self.grid_anchor("center")
164...
165...         self.protocol("WM_DELETE_WINDOW", self.action_exit)
166...
167... ##################################################
```

Aula_10_4_App_TkTable.py

```
168... from Janela import Janela
169...
170... ##################################################
171...
172... Jan1=Janela("Aula_10_4_App_TkTable", 400, 200, 720, 340, "orange")
173... Jan1.mainloop()
174...
175... ##################################################
```

Quais as alternativas corretas em relação ao código das linhas 047 à 052 para imprimir os dados apresentados na figura em uma janela Messagebox ?

Escolha uma ou mais:

a.
```
047...         Texto = ""
048...         for row in range(0, row, 1):
049...             for cell in range(0, col, 1):
050...                 Texto += "%s, " % cell.get()
051...             Texto += "\n"
052...         messagebox.showinfo("Information", Texto)
```

b.
```
047...         Texto = ""
048...         for row in self.Matriz:
049...             for cell in self.Matriz[0]:
050...                 Texto = "%s, " % cell.get()
051...             Texto = "\n"
052...         messagebox.showinfo("Information", Texto)
```

c. ✓
```
047...         Texto = ""
048...         for rw in self.Matriz:
049...             for cl in rw:
050...                 Texto += "%s, " % cl.get()
051...             Texto += "\n"
052...         messagebox.showinfo("Information", Texto)
```

d.
```
047...         Texto = ""
048...         for i in range(0, self.Matriz, 1):
049...             for j in range(0, self.Matriz[0], 1):
050...                 Texto += "%s, " % Matriz[i][j]
051...             Texto = "\n"
052...         messagebox.showinfo("Information", Texto)
```

e.

```
047...          Nome = ""
048...          for rw in self.Matriz:
049...              for cl in rw:
050...                  Nome += "%s, " % cl.get()
051...              Nome += "\n"
052...          messagebox.showinfo("Information", Nome)
```

Sua resposta está correta.

As respostas corretas são:

```
047...          Texto = ""
048...          for rw in self.Matriz:
049...              for cl in rw:
050...                  Texto += "%s, " % cl.get()
051...              Texto += "\n"
052...          messagebox.showinfo("Information", Texto)
```

,

```
047...          Nome = ""
048...          for rw in self.Matriz:
049...              for cl in rw:
050...                  Nome += "%s, " % cl.get()
051...              Nome += "\n"
052...          messagebox.showinfo("Information", Nome)
```

Questão 4

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 122 à 124 para alocar os Labels amarelos que indicam os nomes das colunas da tabela ?

Escolha uma ou mais:

a.
```
122...        Lb_nome = Label(self, text="Nome")
123...        Lb_idade = Label(self, text="Idade")
124...        Lb_ingresso = Label(self, text="Ingresso")
```

b.
```
122...        Lb_nome = Label(self.inter, text="Idade")
123...        Lb_idade = Label(self.inter, text="Nome")
124...        Lb_ingresso = Label(self.inter, text="Ingresso")
```

c. ✓
```
122...        Lb_nome = Label(self.inter, text="Nome")
123...        Lb_idade = Label(self.inter, text="Idade")
124...        Lb_ingresso = Label(self.inter, text="Ingresso")
```

d.
```
122...        lb_ingresso = Label(self, text="Ingresso")
123...        lb_idade = Label(self, text="Idade")
124...        lb_nome = Label(self, text="Nome")
```

e. ✓
```
122...        Lb_ingresso = Label(self.inter, text="Ingresso")
123...        Lb_idade = Label(self.inter, text="Idade")
124...        Lb_nome = Label(self.inter, text="Nome")
```

Sua resposta está correta.

As respostas corretas são:

```
122...        Lb_nome = Label(self.inter, text="Nome")
123...        Lb_idade = Label(self.inter, text="Idade")
124...        Lb_ingresso = Label(self.inter, text="Ingresso")
```

,

```
122...        Lb_ingresso = Label(self.inter, text="Ingresso")
123...        Lb_idade = Label(self.inter, text="Idade")
124...        Lb_nome = Label(self.inter, text="Nome")
```

Questão 5

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 136 à 146 para preencher os campos da tabela com os dados fornecidos no arquivo Dados.py ?

Escolha uma ou mais:

a.
```
136...         nlin = len(Dados)
137...         ncol = len(Dados[0])
138...         self.Matriz = []
139...         for i in range(nlin):
140...             cols = []
141...             for j in range(ncol):
142...                 Et1 = Entry(self.inter, relief=RIDGE)
143...                 Et1.grid(row=i, column=j, sticky=NSEW, padx=2, pady=2)
144...                 Et1.insert(END, '%s' % Dados[i][j])
145...                 cols.append(Et1)
146...             self.Matriz.append(cols)
```

b.
```
136...         nlin = len(Dados[0])
137...         ncol = len(Dados[1])
138...         self.Matriz = []
139...         for i in range(nlin):
140...             cols = []
141...             for j in range(ncol):
142...                 Et1 = Entry(self.inter, relief=RIDGE)
143...                 Et1.grid(row=i, column=j, sticky=NSEW, padx=2, pady=2)
144...                 Et1.insert(END, '%s' % Dados[i+1][j+1])
145...                 cols.append(Et1)
146...             self.Matriz.append(cols)
```

c.
```
136...         n = len(Dados)
137...         m = len(Dados[0])
138...         self.Matriz = []
139...         for i in range(n):
140...             cols = []
141...             for j in range(m):
142...                 Et1 = Entry(self.inter, relief=RIDGE)
143...                 Et1.grid(row=i, column=j, sticky=NSEW, padx=3, pady=3)
144...                 Et1.insert(END, '%s' % Dados[i][j])
145...                 cols.append(Et1)
146...             self.Matriz.append(cols)
```

d.
```
136...         nlin = len(date)
137...         ncol = len(date[0])
138...         self.Matrix = []
139...         for i in range(nlin):
140...             cols = []
141...             for j in range(ncol):
142...                 Et1 = Entry(self.inter, relief=RIDGE)
143...                 Et1.grid(row=i, column=j, sticky=NSEW, padx=2, pady=2)
144...                 Et1.insert(END, '%s' % date[i][j])
145...                 cols.append(Et1)
146...             self.Matrix.append(columns)
```

e.

```
136...          nlin = len(Dados)
137...          self.Matriz = []
138...          for i in range(nlin):
139...              cols = []
140...              for j in range(nlin):
141...                  Et1 = Entry(self.inter, relief=RIDGE)
142...                  Et1.insert(END, '%s' % Dados[j][i])
143...                  cols.append(Et1)
144...              self.Matriz.append(cols)
145...
146...
```

Sua resposta está correta.

As respostas corretas são:

```
136...          nlin = len(Dados)
137...          ncol = len(Dados[0])
138...          self.Matriz = []
139...          for i in range(nlin):
140...              cols = []
141...              for j in range(ncol):
142...                  Et1 = Entry(self.inter, relief=RIDGE)
143...                  Et1.grid(row=i, column=j, sticky=NSEW, padx=2, pady=2)
144...                  Et1.insert(END, '%s' % Dados[i][j])
145...                  cols.append(Et1)
146...              self.Matriz.append(cols)
```

,

```
136...          n = len(Dados)
137...          m = len(Dados[0])
138...          self.Matriz = []
139...          for i in range(n):
140...              cols = []
141...              for j in range(m):
142...                  Et1 = Entry(self.inter, relief=RIDGE)
143...                  Et1.grid(row=i, column=j, sticky=NSEW, padx=3, pady=3)
144...                  Et1.insert(END, '%s' % Dados[i][j])
145...                  cols.append(Et1)
146...              self.Matriz.append(cols)
```

Questão 6

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Porque o label da linha 121 foi o único alocado utilizando a palavra reservada self ?

a. A palavra reservada self deve ser sempre utilizada na alocação do primeiro objeto do tipo Label.

b. A palavra reservada self transforma esse objeto em uma variável local.

c. Esse objeto necessita estar visível em toda a classe pois é acessado pelo método **_configure_interior**

d. Esse objeto necessita estar visível em toda a classe pois é acessado pelo método **calculo_percentual** ✓

e. Não há necessidade de alocar esse objeto com a palavra reservada self, assim como os demais Lables.

Sua resposta está correta.

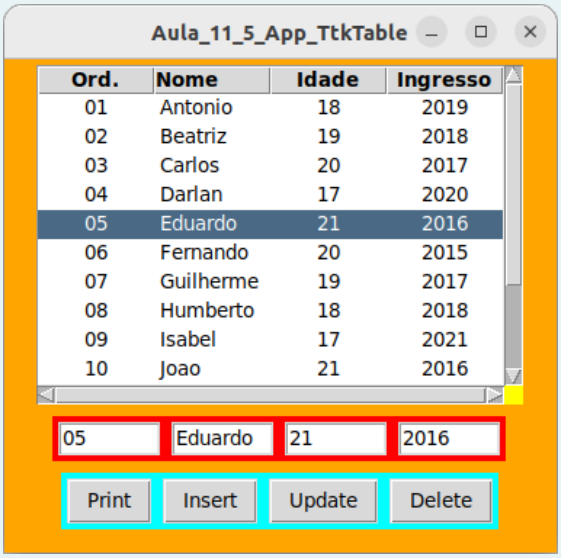
A resposta correta é: Esse objeto necessita estar visível em toda a classe pois é acessado pelo método **calculo_percentual**

Questão 7

Correto

Atingiu 1,00 de 1,00

Dado o código abaixo:



Programa: Aula_11_5_App_TkTable

Dados.py

```
001... ######################################################
002...
003... Dados = [
004...             ['01', 'Antonio', '18', '2019'],
005...             ['02', 'Beatriz', '19', '2018'],
006...             ['03', 'Carlos', '20', '2017'],
007...             ['04', 'Darlan', '17', '2020'],
008...             ['05', 'Eduardo', '21', '2016'],
009...             ['06', 'Fernando', '20', '2015'],
010...             ['07', 'Guilherme', '19', '2017'],
011...             ['08', 'Humberto', '18', '2018'],
012...             ['09', 'Isabel', '17', '2021'],
013...             ['10', 'Joao', '21', '2016'],
014...             ['11', 'Karen', '23', '2015'],
015...             ['12', 'Luciana', '19', '2018'],
016...             ['13', 'Maria', '18', '2020'],
017...             ['14', 'Nicole', '19', '2019']
018...         ]
019...
020... ######################################################
```

Janela.py

```
021... from tkinter import *
022... from tkinter import ttk
023... from tkinter import messagebox
024... from Dados import Dados
025...
026... import sys
027...
028... ################################################
029...
030... class Janela(Tk):
031...     Tab = None
032...
033...     def __init__(self, Str="Janela", px=0, py=0, dx=640, dy=480, cor="ligthgray"):
034...         super().__init__()
035...         super().title(Str)
036...         super().geometry("%dx%d+%d+%d" % (dx, dy, px, py))
037...         super().configure(bg=cor)
038...
039...         self.inicialize(dx, dy)
040...
041...     def action_exit(self):
042...         self.destroy()
043...         sys.exit(0)
044...
045...     def action_print(self):
046...         ########################################
047...         ##                                    ##
048...         ##                                    ##
049...         ##                                    ##
050...         ##         Complete o código          ##
051...         ##             do evento              ##
052...         ##                                    ##
053...         ##                                    ##
054...         ##                                    ##
055...         ########################################
056...
057...     def string_to_int(self, str):
058...         try:
059...             val = int(str)
060...         except ValueError:
061...             val = -1
062...         return(val)
063...
064...     def exist_item(self, val):
065...         ########################################
066...         ##                                    ##
067...         ##         Complete o código          ##
068...         ##             do método              ##
069...         ##                                    ##
070...         ########################################
071...
072...     def action_insert(self):
073...         ########################################
074...         ##                                    ##
075...         ##                                    ##
076...         ##                                    ##
077...         ##                                    ##
078...         ##         Complete o código          ##
079...         ##             do evento              ##
080...         ##                                    ##
081...         ##                                    ##
082...         ##                                    ##
083...         ##                                    ##
084...         ##                                    ##
```

```
085...         ######################################
086...
087...     def action_update(self):
088...         ######################################
089...         ##                                  ##
090...         ##                                  ##
091...         ##                                  ##
092...         ##                                  ##
093...         ##         Complete o código        ##
094...         ##             do evento            ##
095...         ##                                  ##
096...         ##                                  ##
097...         ##                                  ##
098...         ##                                  ##
099...         ######################################
100...
101...     def action_delete(self):
102...         ######################################
103...         ##                                  ##
104...         ##                                  ##
105...         ##         Complete o código        ##
106...         ##             do evento            ##
107...         ##                                  ##
108...         ##                                  ##
109...         ##                                  ##
110...         ######################################
111...
112...     def event_selected(self, event):
113...         ######################################
114...         ##                                  ##
115...         ##                                  ##
116...         ##                                  ##
117...         ##         Complete o código        ##
118...         ##             do evento            ##
119...         ##                                  ##
120...         ##                                  ##
121...         ##                                  ##
122...         ##                                  ##
123...         ######################################
124...
125...     def inicialize(self, dx, dy):
126...         Frm1 = Frame(self)
127...         Frm1.configure(bg='yellow')
128...
129...         Frm2 = Frame(self)
130...         Frm2.configure(bg='red')
131...
132...         Frm3 = Frame(self)
133...         Frm3.configure(bg='cyan')
134...
135...         nlin = len(Dados)
136...         ncol = len(Dados[0])
137...
138...         # scrollbar
139...         Sbx = Scrollbar(Frm1, orient='horizontal')
140...         Sby = Scrollbar(Frm1, orient='vertical')
141...
142...         self.Tab = ttk.Treeview(Frm1, yscrollcommand=Sby.set, xscrollcommand=Sbx.set,
143...                                 height=10) ## 10 linhas
144...         self.Tab.bind('<<TreeviewSelect>>', self.event_selected)
145...
146...         Sby.config(command=self.Tab.yview)
147...         Sbx.config(command=self.Tab.xview)
148...
```

```
149...         self.Tab['columns'] = ('Ord_id', 'Nome_id', 'Idade_id', 'Ingresso_id')
150...
151...         # format our column
152...         self.Tab.column("#0", width=0, stretch=NO)
153...         self.Tab.column("Ord_id", anchor=CENTER, width=80)
154...         self.Tab.column("Nome_id", anchor=W, width=80)
155...         self.Tab.column("Idade_id", anchor=CENTER, width=80)
156...         self.Tab.column("Ingresso_id", anchor=CENTER, width=80)
157...
158...         # Create Headings
159...         self.Tab.heading("#0", text="", anchor=CENTER)
160...         self.Tab.heading("Ord_id", text="Ord.", anchor=CENTER)
161...         self.Tab.heading("Nome_id", text="Nome", anchor=W)
162...         self.Tab.heading("Idade_id", text="Idade", anchor=CENTER)
163...         self.Tab.heading("Ingresso_id", text="Ingresso", anchor=CENTER)
164...
165...         ########################################
166...         ##    Preencha os campos da tabela    ##
167...         ########################################
168...
169...         self.Et_ord = Entry(Frm2, width=8)
170...         self.Et_nome = Entry(Frm2, width=8)
171...         self.Et_idade = Entry(Frm2, width=8)
172...         self.Et_ingresso = Entry(Frm2, width=8)
173...
174...         Bt_print = Button(Frm3, text='Print', command=self.action_print)
175...         Bt_insert = Button(Frm3, text='Insert', command=self.action_insert)
176...         Bt_update = Button(Frm3, text='Update', command=self.action_update)
177...         Bt_delete = Button(Frm3, text='Delete', command=self.action_delete)
178...
179...         self.Tab.grid(row=0, column=0, padx=0, pady=0)
180...         Sbx.grid(row=1, column=0, padx=0, pady=0, sticky=EW)
181...         Sby.grid(row=0, column=1, padx=0, pady=0, sticky=NS)
182...
183...         self.Et_ord.grid(row=0, column=0, sticky=E, padx=4, pady=4)
184...         self.Et_nome.grid(row=0, column=1, sticky=E, padx=4, pady=4)
185...         self.Et_idade.grid(row=0, column=2, sticky=E, padx=4, pady=4)
186...         self.Et_ingresso.grid(row=0, column=3, sticky=E, padx=4, pady=4)
187...
188...         Bt_print.grid(row=0, column=0, sticky=E, padx=4, pady=4)
189...         Bt_insert.grid(row=0, column=1, sticky=W, padx=4, pady=4)
190...         Bt_update.grid(row=0, column=2, sticky=W, padx=4, pady=4)
191...         Bt_delete.grid(row=0, column=3, sticky=W, padx=4, pady=4)
192...
193...         Frm1.grid(row=0, column=0, padx=4, pady=4, sticky=NSEW)
194...         Frm2.grid(row=1, column=0, padx=4, pady=4)
195...         Frm3.grid(row=2, column=0, padx=4, pady=4)
196...
197...         self.grid_anchor("n")
198...
199...         self.protocol("WM_DELETE_WINDOW", self.action_exit)
200...
201... ################################################
```

Aula_11_5_App_TtkTable.py

```
202... from Janela import Janela
203...
204... ######################################################
205...
206... Jan1=Janela("Aula_11_5_App_TtkTable", 400, 200, 380, 340, "orange")
207... Jan1.mainloop()
208...
209... ######################################################
```

Quais as alternativas corretas em relação ao código das linhas 046 à 055 para imprimir os dados apresentados na figura em uma janela Messagebox ?

Escolha uma ou mais:

a.
```
046...          Texto=""
047...          for col in self.Tab[0]:
048...              Texto += "%s, " % col
049...          Texto += "\n"
050...          for v in self.Tab.get_children(0):
051...              row = self.Tab.item(v)[0]
052...              for col in row:
053...                  Texto += "%s, " % col
054...              Texto += "\n"
055...          messagebox.showinfo("Information", Texto)
```

b. ✓
```
046...          Texto=""
047...          for col in self.Tab["columns"]:
048...              Texto += "%s, " % col
049...          Texto += "\n"
050...          for v in self.Tab.get_children():
051...              row = self.Tab.item(v)["values"]
052...              for col in row:
053...                  Texto += "%s, " % col
054...              Texto += "\n"
055...          messagebox.showinfo("Information", Texto)
```

c.
```
046...          Texto=""
047...          for col in self.Tab["columns"]:
048...              Texto = "%s, " % col
049...          Texto = "\n"
050...          for v in self.Tab.get_children():
051...              row = self.Tab.item(v)["values"]
052...              for col in row:
053...                  Texto = "%s, " % col
054...              Texto = "\n"
055...          messagebox.showinfo("Information", Texto)
```

d. ✓
```
046...          Aux=""
047...          for col in self.Tab["columns"]:
048...              Aux += "%s, " % col
049...          Aux += "\n"
050...          for v in self.Tab.get_children():
051...              row = self.Tab.item(v)["values"]
052...              for col in row:
053...                  Aux += "%s, " % col
054...              Aux += "\n"
055...          messagebox.showinfo("Information", Aux)
```

e.
```
046...          Texto=""
047...          for col in self.Tab["columns"]:
048...              Texto += "%s, " % col
049...          Texto += "\n"
050...          for i in range(0, len(self.Tab), 1):
051...              row = self.Tab.item(v)["values"]
052...              for col in row:
053...                  Texto += "%d, " % col
054...              Texto += "\n"
055...          messagebox.showinfo("Information", Texto)
```

Sua resposta está correta.

As respostas corretas são:

```
        Texto=""
        for col in self.Tab["columns"]:
            Texto += "%s, " % col
        Texto += "\n"
        for v in self.Tab.get_children():
            row = self.Tab.item(v)["values"]
            for col in row:
                Texto += "%s, " % col
            Texto += "\n"
        messagebox.showinfo("Information", Texto)
```

,

```
        Aux=""
        for col in self.Tab["columns"]:
            Aux += "%s, " % col
        Aux += "\n"
        for v in self.Tab.get_children():
            row = self.Tab.item(v)["values"]
            for col in row:
                Aux += "%s, " % col
            Aux += "\n"
        messagebox.showinfo("Information", Aux)
```

Questão 8

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 065 à 070 para detectar se um determinado item pertence à tabela de dados ?

a.
```
065...         for i in range(0, self.Tab.get_children(), 1):
066...             row = self.Tab.item(v)["values"]
067...             i = row
068...             if (i == val):
069...                 return(True)
070...         return(False)
```

b.
```
065...         for v in self.Tab.get_children():
066...             rw = self.Tab.item(v)
067...             i = row[0]
068...             if (i == val):
069...                 return(True)
070...         return(False)
```

c. ✓
```
065...         for v in self.Tab.get_children():
066...             row = self.Tab.item(v)["values"]
067...             i = row[0]
068...             if (i == val):
069...                 return(True)
070...         return(False)
```

d.
```
065...         for i in range(0, self.Tab, 1):
066...             row = self.Tab.item(i)["values"]
067...             i = row[0]
068...             if (i = val):
069...                 return(True)
070...         return(False)
```

e. ✓
```
065...         for v in self.Tab.get_children():
066...             rw = self.Tab.item(v)["values"]
067...             i = rw[0]
068...             if (i == val):
069...                 return(True)
070...         return(False)
```

Sua resposta está correta.

As respostas corretas são:

```
065...         for v in self.Tab.get_children():
066...             row = self.Tab.item(v)["values"]
067...             i = row[0]
068...             if (i == val):
069...                 return(True)
070...         return(False)
```

,

```
065...         for v in self.Tab.get_children():
066...             rw = self.Tab.item(v)["values"]
067...             i = rw[0]
068...             if (i == val):
069...                 return(True)
070...         return(False)
```

Questão 9

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 073 à 085 para inserir um novo item pertence na tabela de dados ?

a.

```
073...        ord = self.Et_ord.get()
074...        i = self.string_to_int(ord)
075...        if (self.exist_item(i)==False):
076...            if (i > 0):
077...                nome = self.Et_nome.get()
078...                idade = self.Et_idade.get()
079...                ingresso = self.Et_ingresso.get()
080...                self.Tab.insert(parent='', index='end', iid=i-1, text='',
081...                                values=(ord, nome, idade, ingresso))
082...            else:
083...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
084...        else:
085...            messagebox.showerror("Error", "Registro '%s' já existente" % ord)
```

b.

```
073...        ord = self.Et_ord.get()
074...        i = ord - 1
075...        if (self.exist_item(i)==False):
076...            if (i < 0):
077...                nome = self.Et_nome.get()
078...                idade = self.Et_idade.get()
079...                ingresso = self.Et_ingresso.get()
080...                self.Tab.insert(parent='', index='end', iid=i-1, text='',
081...                                values=(ord, nome, idade, ingresso))
082...            else:
083...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
084...        else:
085...            messagebox.showerror("Error", "Registro '%s' já existente" % ord)
```

c.

```
073...        i = self.Et_ord.get()
074...        ord = self.string_to_float(i)
075...        if (self.exist_item(i)==False):
076...            if (i > 0):
077...                nome = self.Et_nome.get()
078...                idade = self.Et_idade.get()
079...                ingresso = self.Et_ingresso.get()
080...                self.Tab.insert(parent='', index='end', iid=i-1, text='',
081...                                values=(ord, nome, idade, ingresso))
082...            else:
083...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
084...        else:
085...            messagebox.showerror("Error", "Registro '%s' já existente" % ord)
```

d.

```
073...        ord = self.Et_ord.get()
074...        i = self.string_to_int(ord)
075...        if (self.exist_item(j)==False):
076...            if (j < 0):
077...                nome = self.Et_nome.get()
078...                idade = self.Et_idade.get()
079...                ingresso = self.Et_ingresso.get()
080...                self.Tab.insert(parent='', index='end', iid=j-1, text='',
081...                                values=(j, nome, idade, ingresso))
082...            else:
083...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
084...        else:
085...            messagebox.showerror("Error", "Registro '%s' já existente" % ord)
```

e.

```
073...        ord = self.Et_ord.get()
074...        i = self.string_to_int(ord)
075...        if (self.exist_item(i)==False):
076...            if (i > 0):
077...                ingresso = self.Et_ingresso.get()
078...                idade = self.Et_idade.get()
079...                nome = self.Et_nome.get()
080...                self.Tab.insert(parent='', index='end', iid=i-1, text='',
081...                                values=(ord, nome, idade, ingresso))
082...            else:
083...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
084...        else:
085...            messagebox.showerror("Error", "Registro '%s' já existente" % ord)
```

Sua resposta está correta.

As respostas corretas são:

```
073...        ord = self.Et_ord.get()
074...        i = self.string_to_int(ord)
075...        if (self.exist_item(i)==False):
076...            if (i > 0):
077...                nome = self.Et_nome.get()
078...                idade = self.Et_idade.get()
079...                ingresso = self.Et_ingresso.get()
080...                self.Tab.insert(parent='', index='end', iid=i-1, text='',
081...                                values=(ord, nome, idade, ingresso))
082...            else:
083...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
084...        else:
085...            messagebox.showerror("Error", "Registro '%s' já existente" % ord)
```

,

```
073...        ord = self.Et_ord.get()
074...        i = self.string_to_int(ord)
075...        if (self.exist_item(i)==False):
076...            if (i > 0):
077...                ingresso = self.Et_ingresso.get()
078...                idade = self.Et_idade.get()
079...                nome = self.Et_nome.get()
080...                self.Tab.insert(parent='', index='end', iid=i-1, text='',
081...                                values=(ord, nome, idade, ingresso))
082...            else:
083...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
084...        else:
085...            messagebox.showerror("Error", "Registro '%s' já existente" % ord)
```

Questão 10

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 088 à 099 para atualizar um item pertence à tabela de dados ?

Escolha uma ou mais:

a.

```
088...        ord = self.Et_ord.get()
089...        i = self.string_to_float(ord)
090...        if (self.exist_item(i) == True):
091...            if (i > 0):
092...                nome = self.Et_nome.get()
093...                idade = self.Et_idade.get()
094...                ingresso = self.Et_ingresso.get()
095...                self.Tab.item(i-1, text='', values=(ordem, nome, idade, ingress
096...            else:
097...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
098...        else:
099...            messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

b.

```
088...        ord = self.Et_ord.get()
089...        i = self.string_to_int(ord)
090...        if (self.exist_item(i) == True):
091...            if (i > 0):
092...                nome = self.Et_nome.get()
093...                idade = self.Et_idade.get()
094...                ingresso = self.Et_ingresso.get()
095...                self.Tab.item(i-1, text='', values=(ord, nome, idade, ingresso)
096...            else:
097...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
098...        else:
099...            messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

c.

```
088...        ord = self.Et_ord.get()
089...        i = string_to_int(ord)
090...        if (exist_item(i) == True):
091...            if (i > 0):
092...                nome = Et_nome.get()
093...                idade = Et_idade.get()
094...                ingresso = Et_ingresso.get()
095...                Tab.item(i-1, text='', values=(ord, nome, idade, ingresso))
096...            else:
097...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
098...        else:
099...            messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

d.

```
088...        ord = self.Et_ord.get()
089...        j = self.string_to_int(ord)
090...        if (self.exist_item(j) == True):
091...            if (j > 0):
092...                ingresso = self.Et_ingresso.get()
093...                nome = self.Et_nome.get()
094...                idade = self.Et_idade.get()
095...                self.Tab.item(j-1, text='', values=(ord, nome, idade, ingresso)
096...            else:
097...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
098...        else:
099...            messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

e.

```
088...         ord = self.Et_ord.get(0)
089...         i = self.string_to_int(ord)
090...         if (self.exist_item(i) = True):
091...             if (i > 0):
092...                 nome = self.Et_nome.get(0)
093...                 idade = self.Et_idade.get(0)
094...                 ingresso = self.Et_ingresso.get(0)
095...                 self.Tab.item(i-1, text='', values=(ord, nome, idade, ingresso)
096...             else:
097...                 messagebox.showerror("Error", "Registro '%s' inválido" % ord)
098...         else:
099...             messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

Sua resposta está correta.

As respostas corretas são:

```
088...         ord = self.Et_ord.get()
089...         i = self.string_to_int(ord)
090...         if (self.exist_item(i) == True):
091...             if (i > 0):
092...                 nome = self.Et_nome.get()
093...                 idade = self.Et_idade.get()
094...                 ingresso = self.Et_ingresso.get()
095...                 self.Tab.item(i-1, text='', values=(ord, nome, idade, ingresso))
096...             else:
097...                 messagebox.showerror("Error", "Registro '%s' inválido" % ord)
098...         else:
099...             messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

,

```
088...         ord = self.Et_ord.get()
089...         j = self.string_to_int(ord)
090...         if (self.exist_item(j) == True):
091...             if (j > 0):
092...                 ingresso = self.Et_ingresso.get()
093...                 nome = self.Et_nome.get()
094...                 idade = self.Et_idade.get()
095...                 self.Tab.item(j-1, text='', values=(ord, nome, idade, ingresso))
096...             else:
097...                 messagebox.showerror("Error", "Registro '%s' inválido" % ord)
098...         else:
099...             messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

Questão 11

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 102 à 110 para apagar um item pertence à tabela de dados ?

Escolha uma ou mais:

a.
```
102...        ord = self.Et_ord.get()
103...        i = self.string_to_int(ord)
104...        if (self.exist_item(i) == True):
105...            if (i > 0):
106...                self.Tab.delete(i-1)
107...            else:
108...                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
109...        else:
110...            messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

b.
```
102...        ord = self.Et_ord.get()
103...        i = self.string_to_int(od)
104...        if (self.exist_item(i) == True):
105...            if (i > 0):
106...                self.Tab.delete(i-1)
107...            else:
108...                messagebox.showerror("Error", "Registro '%s' inválido" % od)
109...        else:
110...            messagebox.showerror("Error", "Registro '%s' inexistente" % od)
```

c.
```
102...        ord = self.Et_ord.get()
103...        i = self.string_to_int(ord)
104...        if (self.exist_item(i) == True):
105...            if (i > 0):
106...                self.Tab.insert(i-1)
107...            else:
108...                messagebox.show("Error", "Registro '%s' inválido" % ord)
109...        else:
110...            messagebox.show("Error", "Registro '%s' inexistente" % ord)
```

d.
```
102...        ord = self.Et_ord.get()
103...        i = self.string_to_int(ord)
104...        if (self.exist_item(i) == True):
105...            if (i < 0):
106...                self.Tab.delete(i-1)
107...            else:
108...                messagebox.error("Error", "Registro '%s' inválido" % ord)
109...        else:
110...            messagebox.error("Error", "Registro '%s' inexistente" % ord)
```

e.
```
102...        str = self.Et_ord.get()
103...        i = self.string_to_int(str)
104...        if (self.exist_item(i) == True):
105...            if (i > 0):
106...                self.Tab.delete(i-1)
107...            else:
108...                messagebox.showerror("Error", "Registro '%s' inválido" % str)
109...        else:
110...            messagebox.showerror("Error", "Registro '%s' inexistente" % str)
```

Sua resposta está correta.

As respostas corretas são:

```
        ord = self.Et_ord.get()
        i = self.string_to_int(ord)
        if (self.exist_item(i) == True):
            if (i > 0):
                self.Tab.delete(i-1)
            else:
                messagebox.showerror("Error", "Registro '%s' inválido" % ord)
        else:
            messagebox.showerror("Error", "Registro '%s' inexistente" % ord)
```

,

```
        str = self.Et_ord.get()
        i = self.string_to_int(str)
        if (self.exist_item(i) == True):
            if (i > 0):
                self.Tab.delete(i-1)
            else:
                messagebox.showerror("Error", "Registro '%s' inválido" % str)
        else:
            messagebox.showerror("Error", "Registro '%s' inexistente" % str)
```

Questão 12

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 113 à 123 para selecionar uma linha pertence à tabela de dados e inseri-la nas caixas de entrada do rodapé ?

Escolha uma ou mais:

a.

```
113...         for sel_item in self.Tab.selection():
114...             row = self.Tab.item(sel)["values"]
115...             self.Et_ord.delete(0, END)
116...             self.Et_ord.insert(END, "%02d" % row[0])
117...             self.Et_nome.delete(0, END)
118...             self.Et_nome.insert(END, "%s" % row[1])
119...             self.Et_idade.delete(0, END)
120...             self.Et_idade.insert(END, "%s" % row[1])
121...             self.Et_ingresso.delete(0, END)
122...             self.Et_ingresso.insert(END, "%s" % row[2])
123...             break
```

b.

```
113...         for sel_item in self.Tab.selection():
114...             row = self.Tab.item(sel_item)["ord"]
115...             self.Et_ord.delete(0, END)
116...             self.Et_ord.insert(END, "%02d" % row[3])
117...             self.Et_nome.delete(0, END)
118...             self.Et_nome.insert(END, "%s" % row[2])
119...             self.Et_idade.delete(0, END)
120...             self.Et_idade.insert(END, "%s" % row[1])
121...             self.Et_ingresso.delete(0, END)
122...             self.Et_ingresso.insert(END, "%s" % row[0])
123...             break
```

c.

```
113...         for i in range(0, self.Tab, 1):
114...             row = self.Tab.item(sel)["values"]
115...             self.Et_ord.delete(0, END)
116...             self.Et_ord.insert(END, "%02d" % row[0])
117...             self.Et_nome.delete(0, END)
118...             self.Et_nome.insert(END, "%s" % row[1])
119...             self.Et_idade.delete(0, END)
120...             self.Et_idade.insert(END, "%s" % row[2])
121...             self.Et_ingresso.delete(0, END)
122...             self.Et_ingresso.insert(END, "%s" % row[3])
123...             continue
```

d.

```
113...         for sel_item in self.Tab.selection():
114...             row = self.Tab.item(sel_item)["values"]
115...             self.Et_ord.delete(0, END)
116...             self.Et_ord.insert(END, "%02d" % row[0])
117...             self.Et_nome.delete(0, END)
118...             self.Et_nome.insert(END, "%s" % row[1])
119...             self.Et_idade.delete(0, END)
120...             self.Et_idade.insert(END, "%s" % row[2])
121...             self.Et_ingresso.delete(0, END)
122...             self.Et_ingresso.insert(END, "%s" % row[3])
123...             break
```

e.

```
113...          for sel_item in self.Tab.selection():
114...              rw = self.Tab.item(sel_item)["values"]
115...              self.Et_ord.delete(0, END)
116...              self.Et_ord.insert(END, "%02d" % rw[0])
117...              self.Et_nome.delete(0, END)
118...              self.Et_nome.insert(END, "%s" % rw[1])
119...              self.Et_idade.delete(0, END)
120...              self.Et_idade.insert(END, "%s" % rw[2])
121...              self.Et_ingresso.delete(0, END)
122...              self.Et_ingresso.insert(END, "%s" % rw[3])
123...              break
```

Sua resposta está correta.

As respostas corretas são:

```
113...          for sel_item in self.Tab.selection():
114...              row = self.Tab.item(sel_item)["values"]
115...              self.Et_ord.delete(0, END)
116...              self.Et_ord.insert(END, "%02d" % row[0])
117...              self.Et_nome.delete(0, END)
118...              self.Et_nome.insert(END, "%s" % row[1])
119...              self.Et_idade.delete(0, END)
120...              self.Et_idade.insert(END, "%s" % row[2])
121...              self.Et_ingresso.delete(0, END)
122...              self.Et_ingresso.insert(END, "%s" % row[3])
123...              break
```

,

```
113...          for sel_item in self.Tab.selection():
114...              rw = self.Tab.item(sel_item)["values"]
115...              self.Et_ord.delete(0, END)
116...              self.Et_ord.insert(END, "%02d" % rw[0])
117...              self.Et_nome.delete(0, END)
118...              self.Et_nome.insert(END, "%s" % rw[1])
119...              self.Et_idade.delete(0, END)
120...              self.Et_idade.insert(END, "%s" % rw[2])
121...              self.Et_ingresso.delete(0, END)
122...              self.Et_ingresso.insert(END, "%s" % rw[3])
123...              break
```

Questão 13

Correto

Atingiu 1,00 de 1,00

Considere o código fornecido anteriormente:
Quais as alternativas corretas em relação ao código das linhas 165 à 167 para preencher os campos da tabela com os dados fornecidos no arquivo Dados.py ?

Escolha uma ou mais:

a.
```
165...        for index in range(0, nlin, 1):
166...            self.Tab.insert(parent='', index='end', iid=i, text='',
167...                            values=(Dados[i][0], Dados[i][1], Dados[i][2], Dados[i
```

b.
```
165...        for i in range(0, nlin, 1):
166...            self.Tab.insert(parent='', index='end', iid=i, text='',
167...                            values=(Dados[i][0], Dados[i][1], Dados[i][2], Dados[i
```

c.
```
165...        for j in range(0, nlin, 1):
166...            self.Tab.insert(parent='', index='end', iid=j, text='',
167...                            values=(Dados[j][0], Dados[j][1], Dados[j][2], Dados[j
```

d.
```
165...        for i in range(0, ncol, 1):
166...            self.Tab.insert(parent='', index='end', iid=i, text='',
167...                            values=(Dados[j][0], Dados[j][1], Dados[j][2], Dados[j
```

e.
```
165...        for index in range(0, ncol, 2):
166...            self.Tab.insert(parent='', index='end', iid=i, text='',
167...                            values=(Dados[i][3], Dados[i][2], Dados[i][2], Dados[i
```

Sua resposta está correta.

As respostas corretas são:

```
165...        for i in range(0, nlin, 1):
166...            self.Tab.insert(parent='', index='end', iid=i, text='',
167...                            values=(Dados[i][0], Dados[i][1], Dados[i][2], Dados[i][3]))
```

,

```
165...        for j in range(0, nlin, 1):
166...            self.Tab.insert(parent='', index='end', iid=j, text='',
167...                            values=(Dados[j][0], Dados[j][1], Dados[j][2], Dados[j][3]))
```

Questão 14

Correto

Atingiu 2,00 de 2,00

Assista o vídeo disponível no link abaixo:

Título: 018 - Python tkinter - APLICAÇÃO PARA DESENHO DE INTERFACES
Autor: João Ribeiro
Link: (Acessar aqui)

Agora responda a seguinte questão:
Quais os principais passos descritos no vídeo para utilizar o aplicativo Pygubu para gerar uma interface em Python Tkinter?
Mantenha a mesma ordem apresentada no vídeo!
Ative a legenda no rodapé do video player.

Passo 01:	**Localize na internet a página do fabricante do Pygubu**	✓
Passo 02:	**Digite: "pip install pygubu" para instalar o software**	✓
Passo 03:	**Chame o aplicativo: "pygubu-designer" para iniciar o software instalado**	✓
Passo 04:	**Crie um container Toplevel**	✓
Passo 05:	**Acesse o menu Preview para visualizar a interface que está sendo construída**	✓
Passo 06:	**Adicione um Label ao container Toplevel**	✓
Passo 07:	**Altere a geometria do container Toplevel para 500x300**	✓
Passo 08:	**Adicione um Button ao container Toplevel**	✓
Passo 09:	**Salve o aplicativo gerado com a extensão .ui**	✓
Passo 10:	**Crie um aplicativo em Python capaz de ler e apresentar a interface .ui**	✓

Sua resposta está correta.

A resposta correta é:
Assista o vídeo disponível no link abaixo:

Título: 018 - Python tkinter - APLICAÇÃO PARA DESENHO DE INTERFACES
Autor: João Ribeiro
Link: (Acessar aqui)

Agora responda a seguinte questão:
Quais os principais passos descritos no vídeo para utilizar o aplicativo Pygubu para gerar uma interface em Python Tkinter?
Mantenha a mesma ordem apresentada no vídeo!
Ative a legenda no rodapé do video player.

Passo 01:	**[Localize na internet a página do fabricante do Pygubu]**
Passo 02:	**[Digite: "pip install pygubu" para instalar o software]**
Passo 03:	**[Chame o aplicativo: "pygubu-designer" para iniciar o software instalado]**
Passo 04:	**[Crie um container Toplevel]**
Passo 05:	**[Acesse o menu Preview para visualizar a interface que está sendo construída]**
Passo 06:	**[Adicione um Label ao container Toplevel]**
Passo 07:	**[Altere a geometria do container Toplevel para 500x300]**
Passo 08:	**[Adicione um Button ao container Toplevel]**
Passo 09:	**[Salve o aplicativo gerado com a extensão .ui]**
Passo 10:	**[Crie um aplicativo em Python capaz de ler e apresentar a interface .ui]**

Questão 15

Correto

Atingiu 1,00 de 1,00

Consulte o link abaixo:

Título: pygubu-designer 0.32
Site: PyPI - Python Package Index
Link: (Acessar aqui)

Agora responda quais as alternativas estão corretas em relação ao Pygubu

a. É um software para auxiliar no desenvolvimento de aplicativos Android
b. É um software para auxiliar no desenvolvimento de interface gráfica com a biblioteca Tkinter ✓
c. É um software para auxiliar no desenvolvimento de aplicativos em PyQt5
d. É um software para auxiliar no diagnóstico de problemas no computador
e. É um software para auxiliar no desenvolvimento de interface gráfica com biblioteca matplotlib

Sua resposta está correta.

A resposta correta é: É um software para auxiliar no desenvolvimento de interface gráfica com a biblioteca Tkinter

Questão 16

Correto

Atingiu 1,00 de 1,00

Consulte o link abaixo:

Título: pygubu-designer 0.32
Site: PyPI - Python Package Index
Link: (Acessar aqui)

Agora responda quais as alternativas estão corretas em relação ao processo de desenvolvimento utilizando o Pygubu

a. Quando você cria uma interface, a especificação da mesma é salva em um arquivo Python, com extensão ".py"
b. Quando você cria uma interface, a especificação da mesma é salva em um arquivo xml, com extensão ".xml"
c. Quando você cria uma interface, a especificação da mesma é salva em um arquivo xml, com extensão ".ui" ✓
d. Quando você cria uma interface, a especificação da mesma é salva em um arquivo txt, com extensão ".txt"
e. Quando você cria uma interface, a especificação da mesma roda direto sobre o interpretador do Python

Sua resposta está correta.

A resposta correta é: Quando você cria uma interface, a especificação da mesma é salva em um arquivo xml, com extensão ".ui"

Questão 17

Correto

Atingiu 2,00 de 2,00

Considere que foi montada uma interface com o Pygubu e gravada com o nome: 'HelloWorld_App.ui'. Essa interface é representada pela Figura Q-04:

Figura Q_04: Aplicativo HelloWorld_App

Em seguida foi montado um programa para ler e apresentar essa interface.

Código do programa: HelloWorld_App.py

```
from [ tkinter ] ✓ import *
import [ pygubu ] ✓

class [ HelloWorld_App ] ✓ :

    def __init__(self):

        #1: Create a builder
        self.builder = [ pygubu.Builder() ] ✓

        #2: Load an ui file
        self.builder.add_from_file([ 'HelloWorld_App.ui' ] ✓)

        #3: Create the mainwindow
        self.Jan1 = [ self.builder ] ✓ .get_object('JanelaPrincipal')

    def run(self):
        self.Jan1.[ mainloop() ] ✓

if __name__ == '__main__':
    app = [ HelloWorld_App() ] ✓
    app.run()
```

Arraste sobre o código as respostas correspondentes para construir esse aplicativo.

pygubu.compiler

Sua resposta está correta.

A resposta correta é:
Considere que foi montada uma interface com o Pygubu e gravada com o nome: 'HelloWorld_App.ui'. Essa interface é representada pela Figura Q-04:

Figura Q_04: Aplicativo HelloWorld_App

Em seguida foi montado um programa para ler e apresentar essa interface.

Código do programa: HelloWorld_App.py

```
from [tkinter] import *
import [pygubu]

class [HelloWorld_App]:

    def __init__(self):

        #1: Create a builder
        self.builder = [pygubu.Builder()]

        #2: Load an ui file
        self.builder.add_from_file(['HelloWorld_App.ui'])

        #3: Create the mainwindow
        self.Jan1 = [self.builder].get_object('JanelaPrincipal')

    def run(self):
        self.Jan1.[mainloop()]

if __name__ == '__main__':
    app = [HelloWorld_App()]
    app.run()
```

Arraste sobre o código as respostas correspondentes para construir esse aplicativo.

Questão 18

Correto

Atingiu 2,00 de 2,00

Considere que foi montada uma interface com o Pygubu e gravada com o nome: 'Aluno_App.ui'. Essa interface é representada pela Figura Q-05:

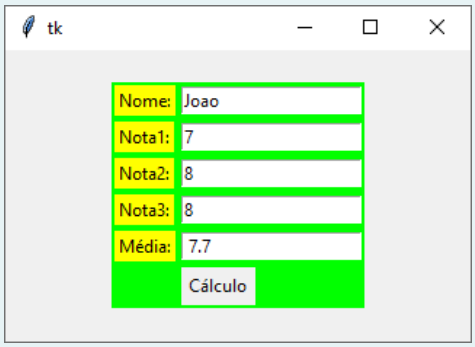


Figura Q_05: Aplicativo Aluno_App

Em seguida foi montado um programa para ler e apresentar essa interface.

Código do programa: Aluno_App.py

```
from tkinter import *
import pygubu

class Aluno_App:

    def __init__(self):

        #1: Create a builder
        self.builder = pygubu.Builder()

        #2: Load an ui file
        self.builder.add_from_file('Aluno_App.ui')

        #3: obtém os objetos da UI
        self.Jan1 = self.builder.get_object('JanelaPrincipal' )
        self.    Et_Nt1     = self.builder.  get_object  ('Et_Nt1')
        self.Et_Nt2  = self.builder.get_object('Et_Nt2')
        self.Et_Nt3 = self.builder.get_object(   'Et_Nt3'   )
        self.Et_Media =  self.builder  .get_object('Et_Media')

        self.    builder     .connect_callbacks(self)

    def run(self):
        self.Jan1.  mainloop()

    def action_calcular(self):
        print('Entrei no evento')
        n1=float(self.Et_Nt1.     get()     )
        n2=float(self.  Et_Nt2.get()  )
        n3=     float     (self.Et_Nt3.get())
        total=  (n1+n2+n3)  /3
        self.Et_Media.delete(0, END)
        self.Et_Media.     insert     (END, "%4.1f" % total)

if __name__ == '    __main__    ':
    app = Aluno_App()
    app.     run()
```

Arraste sobre o código as respostas correspondentes para construir esse aplicativo.

delete

Sua resposta está correta.

A resposta correta é:
Considere que foi montada uma interface com o Pygubu e gravada com o nome: 'Aluno_App.ui'. Essa interface é representada pela Figura Q-05:

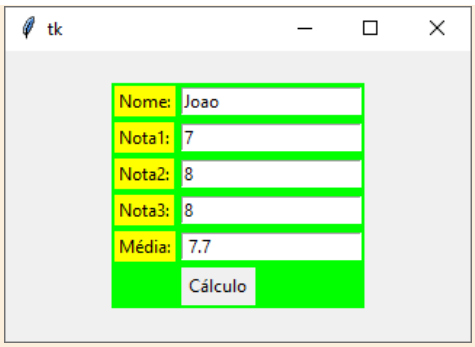


Figura Q_05: Aplicativo Aluno_App

Em seguida foi montado um programa para ler e apresentar essa interface.

Código do programa: Aluno_App.py

```
from tkinter import *
import pygubu

class Aluno_App:

    def __init__(self):

        #[1: Create a builder]
        self.builder = pygubu.Builder()

        #[2: Load an ui file]
        self.builder.add_from_file('Aluno_App.ui')

        #3: obtém os objetos da UI
        self.Jan1 = self.builder.get_object(['JanelaPrincipal'])
        self.[Et_Nt1] = self.builder.[get_object]('Et_Nt1')
        [self.Et_Nt2] = self.builder.get_object('Et_Nt2')
        self.Et_Nt3 = self.builder.get_object(['Et_Nt3'])
        self.Et_Media = [self.builder].get_object('Et_Media')

        self.[builder].connect_callbacks(self)

    def run(self):
        self.Jan1.[mainloop()]

    def action_calcular(self):
        print('Entrei no evento')
        n1=float(self.Et_Nt1.[get()])
        n2=float(self.[Et_Nt2.get()])
        n3=[float](self.Et_Nt3.get())
        total=[(n1+n2+n3)]/3
        self.Et_Media.delete(0, END)
        self.Et_Media.[insert](END, "%4.1f" % total)

if __name__ == '[__main__]':
    app = Aluno_App()
    app.[run()]
```

Arraste sobre o código as respostas correspondentes para construir esse aplicativo.