ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

Interfaces Gráficas em Python II

Interfaces gráficas

Na aula passada, aprendemos sobre como interfaces gráficas são criadas em Python.

Nesta aula, veremos como adicionar funcionalidade aos widgets criados.

Utilizaremos a abordagem de programação orientada a eventos.

Interfaces gráficas

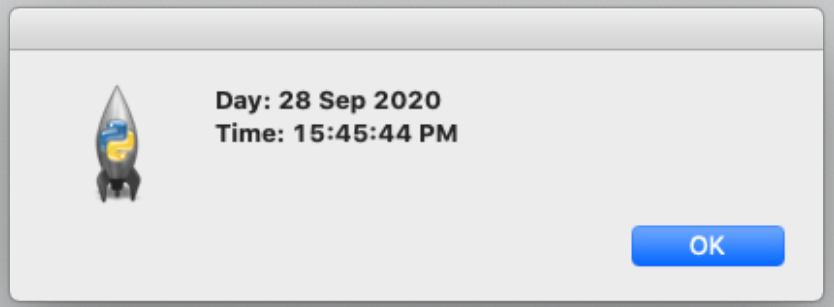
Quando uma interface gráfica é iniciada com a função mainloop(), o interpretador inicia um loop infinito chamado de loop de evento. Seu pseudocódigo é:

```
while True:
    aguarde até um evento ocorrer
    execute a função de tratamento do evento associada
```

Eventos possíveis: clicar, movimentar o mouse, pressionar um tecla, etc.

Interfaces gráficas

Primeiro exemplo: janela com um botão que, quando clicado, exibe o dia e a hora na tela.



Interfaces gráficas

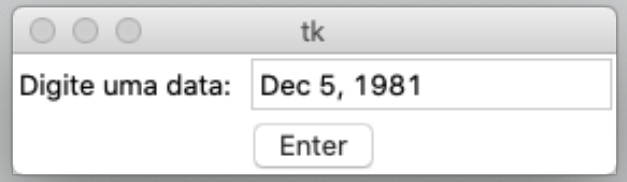
```
from tkinter import Tk, Button, Label
from tkinter.messagebox import showinfo
from time import strftime, localtime

def clicked():
    time = strftime('Day: %d %b %Y\nTime: %H:%M:%S
%p\n', localtime())
    showinfo(message=time)

root = Tk()
button = Button(root, text='Clique', command=clicked)
button.pack()
root.mainloop()
```

Interfaces gráficas

Segundo exemplo: caixa de inserção de texto.



Interfaces gráficas

```
from tkinter import Tk, Button, Label, Entry
from tkinter.messagebox import showinfo
from time import strftime, strptime

def compute():
    global entry
    date = entry.get()
    weekday = strftime('%A', strptime(date, '%b %d, %Y'))
    showinfo(message='{} was a {}'.format(date, weekday))

root = Tk()
label = Label(root, text='Digite uma data: ')
label.grid(row=0, column=0)
entry = Entry(root)
entry.grid(row=0, column=1)
button = Button(root, text='Enter', command=compute)
button.grid(row=1, column=0, columnspan=2)
root.mainloop()
```

Interfaces gráficas

Terceiro exemplo: caixa de texto com diferentes eventos.

>>>

char: u

char: n

char: Return

char: i

mouse left clicked

Interfaces gráficas

Neste exemplo, usamos o widget Text, que funciona como um editor de texto.

Também usamos o método bind() para poder associar diferentes eventos às suas respectivas funções de tratamento.

Para isso, precisamos entender os padrões de evento, que possuem o seguinte formato geral:

<modificador-modificador-tipo-detalhe>

Interfaces gráficas

Exemplos:

- **<Control-Button-1>: Pressionar Ctrl e botão esquerdo do mouse, simultaneamente.**
- **<Button-1><Button-3>: Pressionar botão esquerdo e em seguida o direito do mouse.**
- **<KeyPress-D><Return>: Pressionar D e depois Enter**

Modifier	Description
`Control`	Ctrl key
`Button1`	Left mouse button
`Button3`	Right mouse button
`Shift`	Shift key
Type	
`Button`	Mouse button
`Return`	Enter/Return key
`KeyPress`	Press of a keyboard key
`KeyRelease`	Release of a keyboard key
`Motion`	Mouse motion
Detail	
`<button number>`	1, 2, or 3 for left, middle, and right button, respectively
`<key symbol>`	Key letter symbol

Fonte: Perkovic, 2015

Interfaces gráficas

```
from tkinter import Tk, Text, BOTH

def key_pressed(event):
    print('char: {}'.format(event.keysym))

def mouse_clicked_left(event):
    print('mouse left clicked')

def mouse_clicked_right(event):
    print('mouse right clicked')

root = Tk()
text = Text(root, width=20, height='5')
text.bind('<KeyPress>', key_pressed)
text.bind('<Button-1>', mouse_clicked_left)
text.bind('<Button-2>', mouse_clicked_right)
text.pack(expand=True, fill=BOTH)
root.mainloop()
```

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

Interfaces Gráficas em Python II