Programação com eventos

- Diferente da programação convencional
- O programa não está sob controle 100% do tempo
  - Programa entrega controle ao sistema
  - Em Tk: método(função) mainloop
- Interação gera eventos. Ex:
  - Acionamento de um menu ou de um botão
  - Mouse arrastado sobre uma janela
  - Uma caixa de texto teve seu valor alterado
- O tratamento de um evento é feito por uma rotina "*Callback*"

A opção *command*

- Muitos componentes do Tk suportam a opção command que indica uma função a ser invocada sempre que o widget é acionado
- Tipicamente, a função (ou método) usado obtém valores de outros widgets para realizar alguma operação

Exemplo

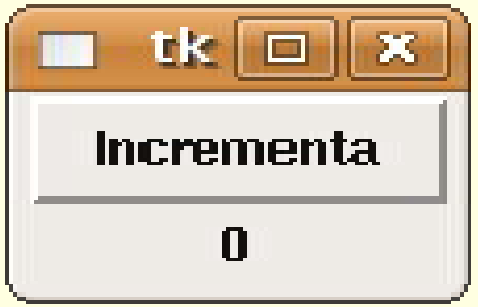
```
from Tkinter import *

def inc():
    n=int(rotulo.configure("text")[4])+1
    rotulo.configure(text=str(n))

b = Button(text="Incrementa",command=inc)
b.pack()
rotulo = Label(text="0”)
rotulo.pack()
mainloop()
```

Exemplo

```
from Tkinter import *

def inc():
    n=int(rotulo.configure("text")[4])+1
    rotulo.configure(text=str(n))

b = Button(text="Incrementa",command=inc)
b.pack()
rotulo = Label(text="0”)
rotulo.pack()
mainloop()
```

Exemplo

```
from Tkinter import *

def inc():
    n=int(rotulo.configure("text")[4])+1
    rotulo.configure(text=str(n))

b = Button(text="Incrementa",command=inc)
b.pack()
rotulo = Label(text="0”)
rotulo.pack()
mainloop()
```



Eventos e *Bind*

- Widgets que não dispõem da opção command também podem receber eventos e responder a eles
- O método bind permite especificar um padrão de eventos ao qual o widget será sensível e uma rotina callback para tratá-lo

  bind(*padrão*,*rotina*)
  - *padrão* é uma string que descreve quais eventos a rotina irá tratar
  - *rotina* é uma função ou método com exatamente um parâmetro: o evento que deve ser tratado

Exemplo

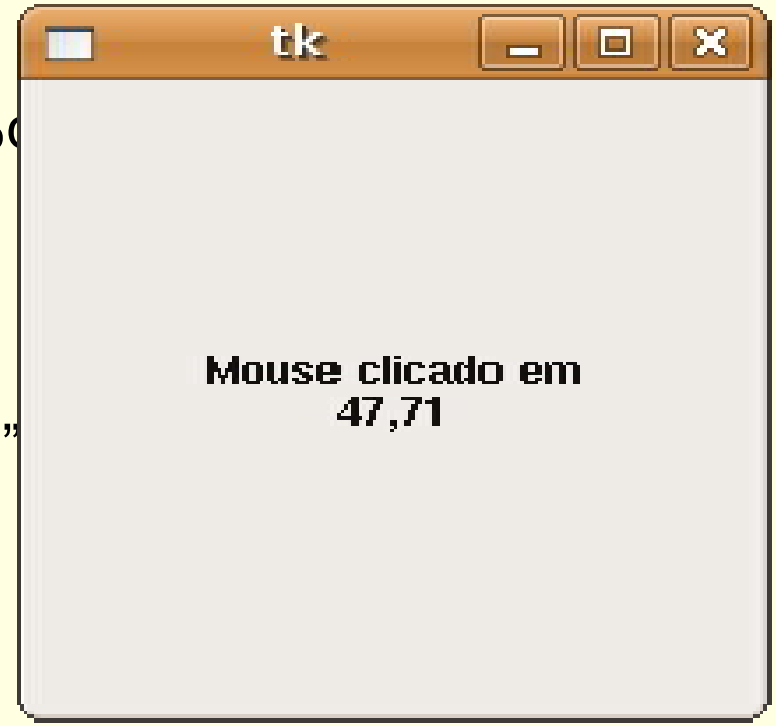
```
from Tkinter import *

def clica (e):
    txt = "Mouse clicado em\n%d,%d"%(e.x,e.y)
    r.configure(text=txt)

r = Label()
r.pack(expand=True, fill="both”)
r.master.geometry("200x200”)
r.bind("<Button-1>", clica)
mainloop()
```

Exemplo

```
from Tkinter import *

def clica (e):
    txt = "Mouse clicado em\n%(
    r.configure(text=txt)

r = Label()
r.pack(expand=True, fill="both”
r.master.geometry("200x200”)
r.bind("<Button-1>", clica)
mainloop()
```

Exemplo

```
from Tkinter import *

def clica (e):
    txt = "Mouse clicado em\n%(
    r.configure(text=txt)

r = Label()
r.pack(expand=True, fill="both”
r.master.geometry("200x200”)
r.bind("<Button-1>", clica)
mainloop()
```



Campos do objeto evento

- x,y : posição do mouse com relação ao canto superior esquerdo do widget
- x_root, y_root: posição do mouse com relação ao canto superior esquerdo da tela
- char: caractere digitado (eventos de teclado)
- keysym: representação simbólica da tecla
- keycode: representação numérica da tecla
- num: número do botão – 1/2/3 = Esquerdo/Meio/Direito – (eventos de mouse)
- widget: o objeto que gerou o evento
- width,height: largura e altura do widget (evento Configure)

Padrões de evento (mouse)

- <Button-*i*> para *i* = 1,2,3: botão *i* do mouse pressionado sobre o widget
- <Motion> : mouse arrastado sobre o widget
- <B*i*-Motion> : mouse arrastado sobre o widget com o botão *i* pressionado
- <ButtonRelease-*i*> : botão *i* do mouse solto sobre o widget
- <Double-Button-*i*>: botão *i* do mouse clicado duas vezes em seguida
- <Enter>: O mouse entrou na área do widget
- <Leave>: O mouse saiu da área do widget

Padrões de evento (teclado)

- *caracter* : O *caracter* foi digitado sobre o widget
- <Key>: Algum caracter foi digitado sobre o widget
- <Return>: Tecla *enter* foi digitada
- <Tab>, <F1>, <Up>...: A tecla correspondente foi digitada
- <Shift-Tab>, <Alt-F1>, <Ctrl-Up>...: Tecla com modificador
- Para os eventos serem gerados, é preciso que o *foco* de teclado esteja sobre o widget
  - Depende do sistema de janelas
  - O foco para um widget pode ser forçado usando o método focus

Exemplo

```
from Tkinter import *
def clica (e):
    txt = "Mouse clicado em\n%d,%d"%(e.x,e.y)
    r.configure(text=txt)
    r.focus()
def tecla(e):
    txt="Keysym=%s\nKeycode=%s\nChar=%s"\
        %(e.keysym,e.keycode,e.char)
    r.configure(text=txt)
r = Label()
r.pack(expand=True, fill="both”)
r.master.geometry("200x200”)
r.bind("<Button-1>", clica)
r.bind("<Key>", tecla)
mainloop()
```

Exemplo

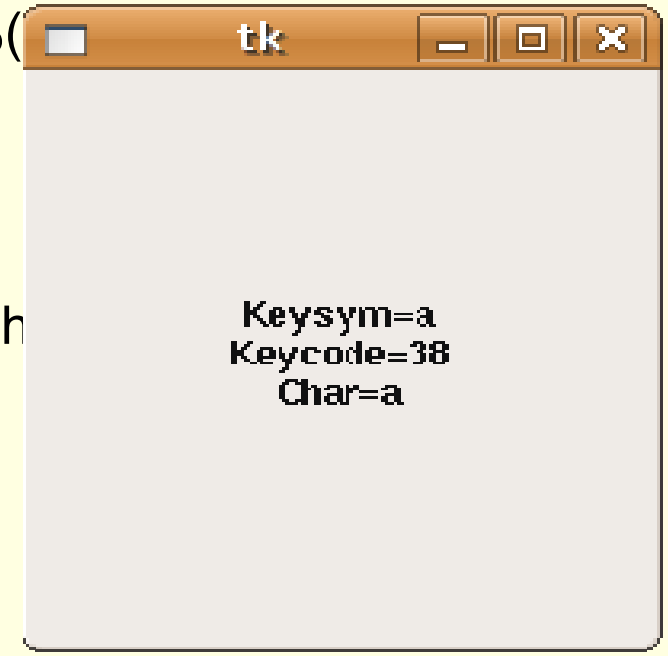
```
from Tkinter import *
def clica (e):
    txt = "Mouse clicado em\n%d,%
    r.configure(text=txt)
    r.focus()
def tecla(e):
    txt="Keysym=%s\nKeycode=%s
        %(e.keysym,e.keycode,e.ch
    r.configure(text=txt)
r = Label()
r.pack(expand=True, fill="both")
r.master.geometry("200x200")
r.bind("<Button-1>", clica)
r.bind("<Key>", tecla)
mainloop()
```

Exemplo

```
from Tkinter import *
def clica (e):
    txt = "Mouse clicado em\n%d,%d"
    r.configure(text=txt)
    r.focus()
def tecla(e):
    txt="Keysym=%s\nKeycode=%s\r
        %(e.keysym,e.keycode,e.char
    r.configure(text=txt)
r = Label()
r.pack(expand=True, fill="both”)
r.master.geometry("200x200”)
r.bind("<Button-1>", clica)
r.bind("<Key>", tecla)
mainloop()
```

Exemplo

```
from Tkinter import *
def clica (e):
    txt = "Mouse clicado em\n%d,%d"%(
    r.configure(text=txt)
    r.focus()
def tecla(e):
    txt="Keysym=%s\nKeycode=%s\nCh
        %(e.keysym,e.keycode,e.char)
    r.configure(text=txt)
r = Label()
r.pack(expand=True, fill="both")
r.master.geometry("200x200")
r.bind("<Button-1>", clica)
r.bind("<Key>", tecla)
mainloop()
```



Exemplo

```
from Tkinter import *
def clica (e):
    txt = "Mouse clicado em\n%d,%d"%(
    r.configure(text=txt)
    r.focus()
def tecla(e):
    txt="Keysym=%s\nKeycode=%s\nCh
        %(e.keysym,e.keycode,e.char)
    r.configure(text=txt)
r = Label()
r.pack(expand=True, fill="both”)
r.master.geometry("200x200”)
r.bind("<Button-1>", clica)
r.bind("<Key>", tecla)
mainloop()
```