Upravujeme vzhľad aplikácií

Táto kapitola neobsahuje úlohy a otázky, pretože je skôr doplňujúcou kapitolou. Informácie v nej obsiahnuté vám pomôžu upraviť vzhľad aplikácií, ktoré ste doposiaľ vytvárali. Naučíme sa vytvárať ovládacie prvky, ktoré použijeme pri vytváraní väčších komplexnejších programov, napríklad pri tvorbe seminárnej práce.
V našich aplikáciách sme doposiaľ bežne používali **canvas**, **button**, **entry**. Nazývali sme ich grafickými súčiastkami. Sú súčasťou grafickej knižnice **tkinter** a v Pythone sa nazývajú **widget** (ovládacie prvky).
Predstavíme si niektoré ďalšie ovládacie prvky, aby sme mohli vytvárať komplexnejšie programy, ktoré sú dobre ovládateľné používateľom.

Metódy zobrazenia widgetov a ich umiestnenie

Keď sme vytvárali nejaký **widget**, zobrazovali sme ho metódou **pack()**. Ešte si na porozumenie zobrazovania prvkov ukážeme na príklade, čo sa stane, keď pre rôzne prvky zavoláme metódu **pack()** v inom poradí. A tiež uvidíme, že aplikácia môže mať viaceré grafické plátna.
Prvý program poukladá do okna aplikácie v poradí **canvas1**, **canvas2**, **label1**, **entry1** a **button1**. Ak poradie volania metód **pack()** zameníme, ako vidíme v ukážke druhého programu, v hornej časti okna budú najprv **label1**, **entry1** a **button1** a až potom bude **canvas2** a **canvas1**.

```
import tkinter, random
def nakresli():
      x = random.randrange(400)
      y = random.randrange(200)
      canvas1.create_text(x, y, text=entry1.get())
      canvas2.create_text(x, y, text=entry1.get()[::-1])

canvas1 = tkinter.Canvas(width=400, height=200, bg='yellow')
canvas2 = tkinter.Canvas(width=400, height=200, bg='green')
label1 = tkinter.Label(text='Napíš meno:')
entry1 = tkinter.Entry()
button1 = tkinter.Button(text='nakresli', command=nakresli)

canvas1.pack()
canvas2.pack()
label1.pack()
entry1.pack()
button1.pack()
```

```
import tkinter, random

def nakresli():
      x = random.randrange(400)
      y = random.randrange(200)
      canvas1.create_text(x, y, text=entry1.get())
      canvas2.create_text(x, y, text=entry1.get()[::-1])

canvas1 = tkinter.Canvas(width=400, height=200, bg='yellow')
canvas2 = tkinter.Canvas(width=400, height=200, bg='green')
label1 = tkinter.Label(text='Napíš meno:')
entry1 = tkinter.Entry()
button1 = tkinter.Button(text='nakresli', command=nakresli)

label1.pack()
entry1.pack()
button1.pack()
canvas2.pack()
canvas1.pack()
```

V programe sme použili widget **label**, ktorý slúži na vypísanie nejakého textu, napríklad k entry.

V metóde **pack()** môžeme použiť aj parameter **side** s hodnotou **'left'**, **'right'**.

```
canvas1.pack()
canvas2.pack()
label1.pack(side='left')
```

```
entry1.pack(side='left')
button1.pack(side='left')
```

Takéto nastavenie poukladá canvasy pod seba a potom dá vedľa seba vľavo ostatné prvky.

Alebo si môžeme vytvoriť aj takéto usporiadanie:

```
canvas1.pack(side='right')
label1.pack(side='left')
entry1.pack(side='left')
button1.pack(side='left')
```

Viacero widgetov môžeme zoskupiť do jedného (mysleného) rámika - **frame**. A, samozrejme, widgetom môžeme nastavovať aj ďalšie vlastnosti (**width**, **bg**, **fg**, ...).

```
import tkinter
canvas = tkinter.Canvas(width=400, height=200, bg='white')
canvas.pack()
frame1 = tkinter.Frame()
frame1.pack(side='left')

button1.pack()
button2 = tkinter.Button(frame1, text='tlačidlo 2', width=20)
button2.pack()

frame2 = tkinter.Frame()
frame2.pack(side='left')
button3 = tkinter.Button(frame2, text='3', width=10, bg='green')
button3.pack()
button4 = tkinter.Button(frame2, text='4', width=10, fg='red')
button4.pack()
```

Na zobrazenie widgetu sa používajú aj metódy **grid()** a **place()**. Pri každom prvku môžeme na jeho zobrazenie použiť len jednu z metód. V metóde **place** nastavujeme súradnice umiestenia widgetu v rámci okna aplikácie. V metóde **grid()** sa widgety umiestňuju do fiktívnej tabuľky pomocou nastavení stĺpca a riadku (viac nájdete na internete).

```
import tkinter
canvas = tkinter.Canvas(width=400, height=200, bg='white')
canvas.pack()

frame1 = tkinter.Frame()
frame1.pack(side='left')
button1 = tkinter.Button(frame1, text='tlačidlo 1', width=20)
button1.pack()
button2 = tkinter.Button(frame1, text='tlačidlo 2', width=20)
button2.pack()

frame2 = tkinter.Frame()
frame2.place(x=300, y=10)
button3 = tkinter.Button(frame2, text='3', width=10, bg='green')
button3.pack()
button4 = tkinter.Button(frame2, text='4', width=10, fg='red')
button4.pack()
```

Listbox

Listbox je **widget**, ktorý ukáže v ponuke viaceré možnosti a my môžeme (štandardne) z nich jednu označiť (v inom nastavenom režime sa dá vyberať aj viacero položiek).
Nasledujúci program vytvorí naplnený **listbox** zoznamom farieb. Po dvojkliku na farbu sa **canvas** prefarbí vybranou farbou. V programe je aj **entry** a dve tlačidlá. Do **entry** zadávame farbu a pridávame ju do listboxu tlačidlom. Iným tlačidlom z listboxu vymažeme označenú položku (farbu).

```
import tkinter
canvas = tkinter.Canvas(width=400, height=200, bg='white')
canvas.pack()
```

```
def prefarbi(event):
      oznacene = listbox1.curselection()
      canvas['bg'] = listbox1.get(oznacene)

def pridaj():
      listbox1.insert('end', entry1.get())

def vymaz():
      oznacene = listbox1.curselection()
      if len(oznacene) == 1:
            listbox1.delete(oznacene)

listbox1 = tkinter.Listbox()
listbox1.pack()

farby = ['red', 'green', 'blue', 'orange', 'yellow', 'white']

for prvok in farby:
      listbox1.insert('end', prvok)

listbox1.bind('<Double-Button-1>', prefarbi)

label1 = tkinter.Label(text='Napíš názov farby:')
label1.pack()
entry1 = tkinter.Entry()
entry1.pack()
button1 = tkinter.Button(text='Pridaj farbu', command=pridaj)
button1.pack()
button2 = tkinter.Button(text='Vymaž označenú farbu', command=vymaz)
button2.pack()
```

Vytvorenie listboxu a niektoré jeho metódy:
Listbox má riadky indexované od čísla 0,

- **`listbox1 = tkinter.Listbox()`** - vytvorenie listboxu. Môžeme nastaviť napríklad **`height`** – počet riadkov listboxu.
- **`listbox1.insert('end', 'hodnota')`** - vloží na koniec listboxu zadanú hodnotu,
- **`listbox1.insert(index, 'hodnota')`** - vloží pred zadaný index listboxu zadanú hodnotu,
- **`listbox1.delete(index)`** - vymaže hodnotu na zadanom indexe,
- **`listbox1.delete(index1, index2)`** - vymaže hodnoty medzi zadanými indexmi, vrátane indexov,
- **`listbox1.curselection()`** - vráti n-ticu označených riadkov (ich indexy),
- **`listbox1.get(index)`** - vráti hodnotu so zadaným indexom,
- **`listbox1.bind('<Double-Button-1>', funkcia)`** - zviaže udalosť dvojkliku na riadku listboxu so zadanou funkciou,
- **`listbox1.size()`** - vráti počet prvkov listboxu (posledný prvok má index o jedno menší),
- **`listbox1.config(vlastnosť=hodnota)`** - nastaví dodatočne niektorú z vlastností, ak sme ju nenastavili pri vytváraní.

Napríklad: **`listbox1.config(height=5)`**, **`listbox1.config(bg='red')`**.

Posúvač

Na zmenu číselných hodnôt môžeme použiť widget **`scale`** (posúvač). Vytvoríme program, ktorý nakreslí **`oval`** a dvoma posúvačmi budeme interaktívne meniť jeho polomer na osi x a polomer na osi y.

```
import tkinter
canvas = tkinter.Canvas(width=440, height=200, bg='white')
canvas.pack()

rx, ry = 100, 50
x, y = 200, 100
canvas.create_oval(x-rx, y-ry, x+rx, y+ry, width=5, outline='green', tags='oval')

def zmena1(event):
      global rx
      rx = scale1.get()
      prekresli()

def zmena2(event):
```

```
        global ry
        ry = scale2.get()
        prekresli()

def prekresli():
        canvas.coords('oval',[x-rx, y-ry, x+rx, y+ry])

scale1 = tkinter.Scale(from_=10, to=200, orient='horizontal',
length=400, command=zmena1)
scale1.pack()
scale1.set(rx)
scale2 = tkinter.Scale(from_=10, to=100, orient='vertical', length=200, command=zmena2)
scale2.place(x=400, y=0)
scale2.set(ry)
```

Vytvorenie **scale** a niektoré jeho metódy:

- **scale1 = tkinter.Scale(from_=10, to=200)** - vytvorenie widgetu **scale**. Môže obsahovať ďalšie nastavenia, napríklad: **orient='horizontal'** alebo **orient='vertical'**, **length=400** (dĺžku widgetu), **command=funkcia** (pri zmene zavolá danú funkciu),
- **scale1 = tkinter.Scale(from_=10, to=200, orient='horizontal', command=zmena)**
- **scale1.get()** - zistí aktuálnu hodnotu,
- **scale1.set(hodnota)** - nastaví aktuálnu hodnotu,
- **scale1.config(from_=0)** - zmení zadanú vlastnosť na zadanú hodnotu,
- **scale1.config(showvalue=0)** - vypne zobrazovanie hodnoty (**showvalue=1** - zapne zobrazovanie hodnoty),
- **scale1.config(label='polomer v osi x')** - nastaví popis súčiastky.