

Kalkulačka

Velký projekt předmětu BPC-PP2

Autor: Václav Zahrádka

17. ledna 2021

Blok 1

Řádek 2: slouží k importování modulu tkinter, kde jsou příkazy pro GUI

pojmenuji ho jako tk -> v dalších příkazech se tak odvolávám na tento modul

Řádek 4: vytvořím prázdný string calc_expr

```
1 # Blok 1:  
2 import tkinter as tk  
3  
4 calc_expr = ""
```

Blok 2

Řádek 7: definuji funkci `btn_pressed` s parametrem `num`

Řádek 8: příkaz `global` slouží k nalezení proměnné `calc_expr` mimo oblast definování funkce (řádky 7 až 10)

Řádek 9: po stisku daného tlačítka tato funkce rozšíří string `calc_expr` o symbol na tlačítku (parametr `num`)

Řádek 10: proměnnou `disp_expr` změním na string proměnné `calc_expr`

Pozn.: tato funkce se provede vždy po stisku tlačítka na kalkulačce (kromě tlačítek DEL, AC a „rovná se“)

```
6 # Blok 2:
7 def btn_pressed(num):
8     global calc_expr
9     calc_expr = calc_expr + str(num)
10    disp_expr.set(calc_expr)
```

Blok 3

Řádek 13: definuji funkci `btn_eq_pressed`

Řádek 14 + 19: příkaz `try` provede skript pod ním, pokud je to možné, jinak bez provedení skočí na řádek 19 a provede se skript pod `except`

Řádek 15: vyhledá proměnnou `calc_expr` mimo tento blok

Řádek 16: do proměnné `total` vložím string, který obsahuje výsledek výpočtu kalkulačky (funkce `eval` spočítá matematicky to, co je ve stringu `calc_expr`, neboli obsah widgetu `Entry`)

Řádek 17: proměnnou `disp_expr` změním na string proměnné `total`

```
12 #Blok 3:
13 def btn_eq_pressed():
14     try:
15         global calc_expr
16         total = str(eval(calc_expr))
17         disp_expr.set(total)
18         calc_expr = disp_expr.get()
19     except:
20         disp_expr.set(" ERROR ")
21         calc_expr = ""
22         display.configure(fg='red')
```

Blok 3

Řádek 18: do proměnné `calc_expr` vložím string proměnné `disp_expr` (výsledek výpočtu jako 1 číslo) – je to důležité, pokud bychom chtěli dále počítat s výsledkem

Řádek 20: proměnnou `disp_expr` změním na string „ERROR“ oznamující chybu při výpočtu

Řádek 21: proměnnou `calc_expr` změním na prázdný string jako na počátku (abych mohl začít počítat od začátku i bez smazání displaye)

Řádek 22: tímto příkazem změním barvu nápisu ERROR na červenou

Pozn.: funkce `btn_eq_pressed` se provede po stisku tlačítka „rovná se“
obsah proměnné `disp_expr` vidíme na displayi

```
12 #Blok 3:
13 def btn_eq_pressed():
14     try:
15         global calc_expr
16         total = str(eval(calc_expr))
17         disp_expr.set(total)
18         calc_expr = disp_expr.get()
19     except:
20         disp_expr.set(" ERROR ")
21         calc_expr = ""
22         display.configure(fg='red')
```

Blok 4

Řádek 25: definuji funkci `btn_AC_pressed` (ta se provede vždy po stisku tlačítka AC)

Řádek 26: příkaz vyhledá proměnnou `calc_expr` mimo tento blok

Řádek 27: proměnnou `calc_expr` změním na prázdný string, čímž smažu paměť kalkulačky

Řádek 28: proměnnou `disp_expr` také změním na prázdnou

Řádek 29: tím změním barvu symbolů v displayi na černou (vhodné při ERRORU)

Pozn.: tato funkce vše smaže, a vrátí tak stav kalkulačky na samý počátek

```
24 # Blok 4:
25 def btn_AC_pressed():
26     global calc_expr
27     calc_expr = ""
28     disp_expr.set("")
29     display.configure(fg='black')
```

Blok 5

Řádek 32: definuji funkci btn_DEL_pressed (ta se spustí vždy po stisknutí tlačítka DEL)

Řádek 33: opět vyhledám proměnnou calc_expr mimo tento blok

Řádek 34: z proměnné calc_expr vymažu poslední symbol

Řádek 35: proměnnou disp_expr změním na string proměnné calc_expr

Pozn.: tato funkce smaže poslední slovo, číslici nebo symbol na displayi

```
31 # Blok 5:
32 def btn_DEL_pressed():
33     global calc_expr
34     calc_expr = calc_expr[0:-1]
35     disp_expr.set(calc_expr)
```

Blok 6

Řádek 38: tento příkaz vytvoří okno GUI, které pojmenuji win

Řádek 39: tímto změním pozadí okna na určitý odstín šedé

Řádek 40: tímto nastavím nadpis okna win, který bude vidět nahoře

Řádek 41: tento příkaz zakáže změnu velikosti okna

Řádek 42: tady nastavím styl písma

```
37 # Blok 6:
38 win = tk.Tk()
39 win.configure(background="LightSlateGray")
40 win.title('Simple Calculator for "blind" people')
41 win.resizable(width=False, height=False)
42 win.option_add('*Font', 'Verdana 20 bold')
```


Blok 7

- Řádek 45:** vytvořím stringovou proměnnou `disp_expr` (jedná se o specifický typ stringu, proto musím pro uložení do ní použít metodu `set()` a pro zkopírování do jiného stringu metodu `get()`)
- Řádek 46:** vytvořím widget `Entry (display)` v okně `win`, pojmenuji ho `display`; nastavím, aby zobrazoval obsah proměnné `disp_expr`; zarovnáám ho doprava; zvolím barvu pozadí `khaki`
- Řádek 47:** `display` tímto roztáhnu vodorovně na 4 sloupce
- Řádek 48:** proměnnou `disp_expr` změním na string „enter your expression“, což je první text zobrazený na `display` po spuštění

```
44 # Blok 7:
45 disp_expr = tk.StringVar()
46 display = tk.Entry(win, textvariable=disp_expr, justify="right", bg="khaki")
47 display.grid(columnspan=4, sticky="WE")
48 disp_expr.set('enter your expression')
```

Blok 8

Řádek 51 + 52: vytvořím šedé tlačítko btn_1 (nápis 1), které po stisknutí napíše číslo 1 do displaye

Řádek 53: tímto umístím tlačítko do řádku 4 a sloupce 0

- podobně vytvořím ostatní tlačítka pomocí tk.Button() a zvolím patřičné umístění pomocí metody .grid()
- parametrem fg zvolím barvu popředí (nápisu) a parametrem bg barvu pozadí (tlačítka)
- do parametru command přiřadím funkci, která se provede po stisknutí tlačítka

Řádek 134: tento příkaz spustí GUI – funguje jako nekonečná smyčka

```
50 # Blok 8:
51 btn_1 = tk.Button(win, text=' 1 ', fg='white', bg='gray',
52                   command = lambda: btn_pressed(1), height=1, width=7)
53 btn_1.grid(row=4, column=0)
    .
    .
    .
130 btn_Clear_ent = tk.Button(win, text=' DEL ', fg='white', bg='Orange',
131                           command = lambda: btn_DEL_pressed(), height=1, width=7)
132 btn_Clear_ent.grid(row=1, column='1')
133
134 win.mainloop()
```

KONEC

Děkuji za pozornost