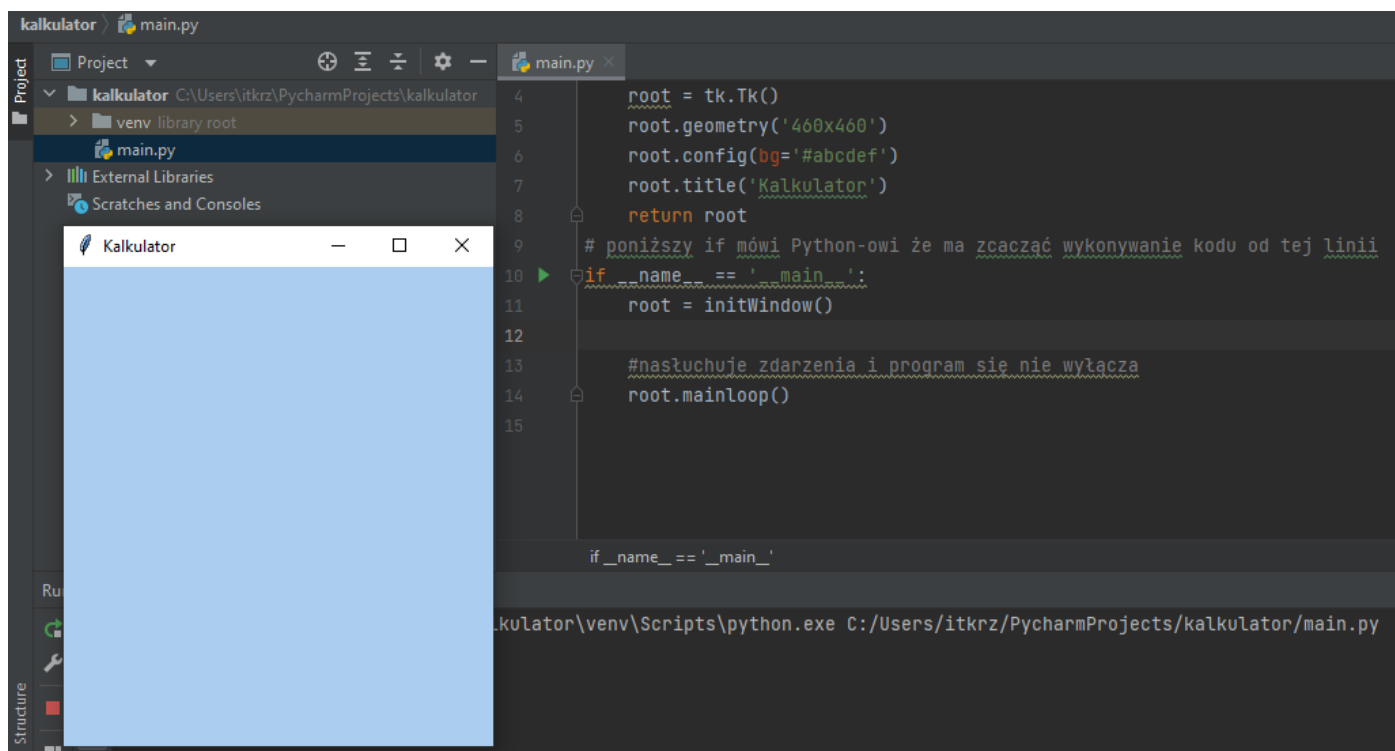


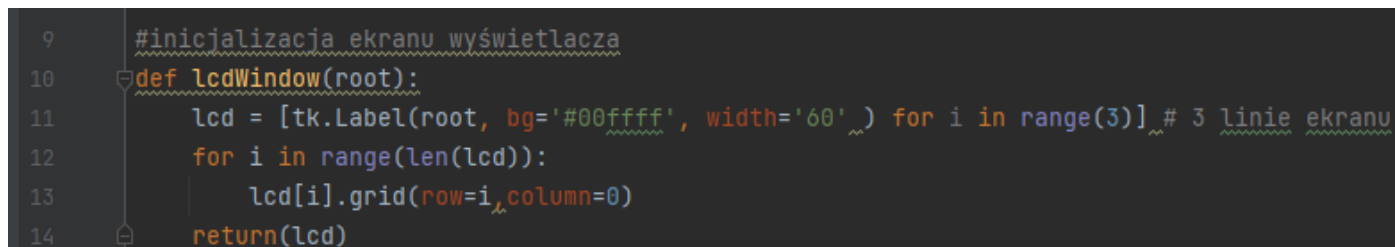
Python – pierwsza praktyczna apka - kalkulator

Otwieramy nowy projekt w PyCharm o nazwie kalkulator

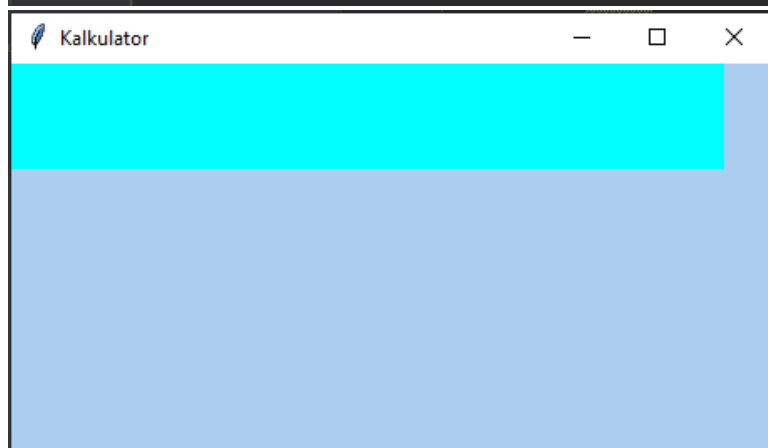
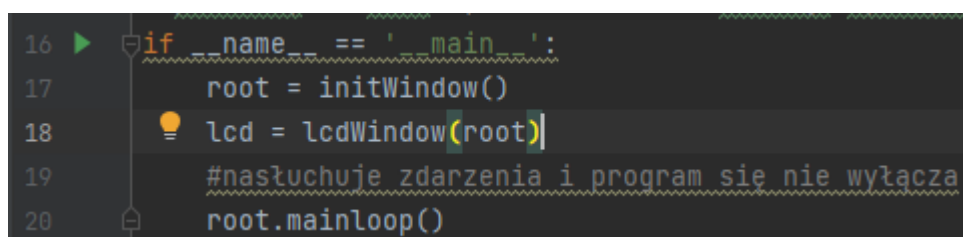
Kod startowy:



Dodajemy funkcję tworzącą ekran wyświetlacza:



Oraz wywołujemy tworzenie ekranu w pętli:



Dodamy marginesy elementom wyświetlacza (przy width w linii 11 należy usunąć apostrofy):

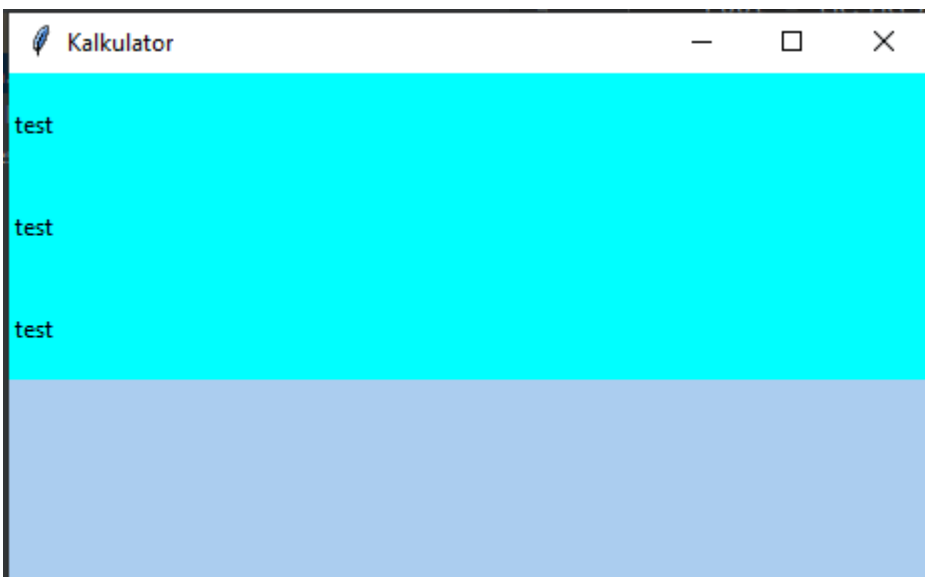
```
10 def lcdWindow(root):
11     lcd = [tk.Label(root, bg='#00ffff', width=60) for i in range(3)] # 3 linie ekranu
12     for i in range(len(lcd)):
13         lcd[i].grid(row=i, column=0, ipady=15, ipadx=1)
14     return lcd
```

Wstawimy do tworzonych linii tekst kontrolny (linia 12):

```
12 lcd = [tk.Label(root, bg='#00FFFF', text='test', width=65) for i in range(3)]
```

Dociągniemy tekst do lewej strony (west):

```
12 lcd = [tk.Label(root, bg='#00FFFF', text='test', anchor='w', borderwidth=2, width=65) for i in range(3)]
13 for i in range(len(lcd)):
14     lcd[i].grid(row=i, column=0, ipady=15, ipadx=1)
15 return lcd
```



Dodajemy pole wprowadzania danych:

```
16 # pole wprowadzania danych
17 def dataPlace(root, lcd):
18     dane = tk.Entry(root).grid(row=len(lcd), column=0)
19     return dane
20
21 # poniższy if mówi Python-owi że ma zacząć wykonywanie
22 if __name__ == '__main__':
23     root = initWindow()
24     lcd = lcdWindow(root)
25     dane = dataPlace(root, lcd)
```



Modyfikujemy wygląd utworzonego pola:

```
18 dane = tk.Entry(root, borderwidth=0).grid(row=len(lcd), column=0, ipadx=166, ipady=10)
```

Dodajemy dodatkowe info o niewłaściwym wprowadzaniu danych (linia 19):

```
19 info = tk.Label(root, text='test', anchor='w', borderwidth=0).grid(row=len(lcd)+1, column=0, ipadx=216, ipady=10)
```

Dodajemy siatkę z przyciskami kalkulatora:

Najpierw dodajemy tablicę z przyciskami:

```
main.py x
1 import tkinter as tk
2 # tablica z symbolami od góry w dół o szer 6szt. #\u21BC - cofnij #\u221A - pierwiastek
3 symbole = ['7', '8', '9', '/', '\u21BA', 'C',
4            '4', '5', '6', '*', '(', ')',
5            '1', '2', '3', '-', 'x^2', '\u221A',
6            '0', '.', '%', '+']
```

Modyfikujemy pole na przyciski jako grid o wymiarach 6 kolumn. Przy okazji zmieniamy rozmiar pól na tekst i wyświetlacz jako 6 kolumn połączonych (columnspan).

Columnspan:

```
15 def lcdWindow(root):
16     lcd = [tk.Label(root, bg='#00FFFF', text='test', anchor='w', borderwidth=2, width=65) for i in range(3)] # 3 linie ekr
17     for i in range(len(lcd)):
18         lcd[i].grid(row=i, columnspan=6, ipady=15, ipadx=1)
19     return lcd
20 # pole wprowadzania danych
21 def dataPlace(root, lcd):
22     dane = tk.Entry(root, borderwidth=0).grid(row=len(lcd), columnspan=6, ipadx=166, ipady=10)
23     info = tk.Label(root, text='test', anchor='w', borderwidth=0).grid(row=len(lcd) + 1, columnspan=6, ipadx=216, ipady=10)
24     return dane
```

Tworzenie przycisków:

```
26 def klawisze(root, lcd):
27     przyciski = [tk.Button(root, text=symbol) for symbol in symbole]
28     j = len(lcd) + 2
29     for i in range(len(przyciski)):
30         if i % 6 == 0:
31             j += 1
32         przyciski[i].grid(row=j, column=i % 6, ipady=2, ipadx=20)
33     return przyciski
34 # poniższy if mówi Pythonowi że ma zacząć wykonywanie kodu od tej linii
35 if __name__ == '__main__':
36     root = initWindow()
37     lcd = lcdWindow(root)
38     dane = dataPlace(root, lcd)
39     przyciski = klawisze(root, lcd)
```

Przyciski mają różną szerokość. Można je dopasować wprowadzając do nich margines:

```
29     for i in range(len(przyciski)):
30         if i % 6 == 0:
31             j += 1
32         margin = 20 if len(symbole[i]) == 1 else 12 # regulacja szerokości przycisków
33         przyciski[i].grid(row=j, column=i % 6, ipady=5, ipadx=margin)
34     return przyciski
```

Po ustawieniu przycisków można usunąć im ramkę oraz zmienić kolor tła co spowoduje polepszenie wrażenia wzrokowego.

```
26 def klawisze(root, lcd):
27     przyciski = [tk.Button(root, text=symbol, bg='#abcdef', borderwidth=0) for symbol in symbole]
28     j = len(lcd) + 2
```

Można już usunąć testowy tekst:

```
15 def lcdWindow(root):
16     lcd = [tk.Label(root, bg='#00FFFF', text='test', anchor=
17         for i in range(len(lcd)):
18             lcd[i].grid(row=i, columnspan=6, ipady=15, ipadx=1)
19     return lcd
20     # pole wprowadzania danych
21 def dataPlace(root, lcd):
22     dane = tk.Entry(root, borderwidth=0).grid(row=len(lcd),
23     info = tk.Label(root, text='test', anchor='w', borderwid
24     return dane
```

Dodajemy przycisk '=':

```
33     przyciski[i].grid(row=j, column=i % 6, ipady=5, ipadx=margin)
34     klawiszRowny = tk.Button(root, text='=', bg='#ffff00', borderwidth=0)
35     klawiszRowny.grid(row=len(lcd)+6, column=4, columnspan=2, ipady=5, ipadx=40)
36     return przyciski
```

Nadszedł czas na oprogramowanie kliknięć przycisków. Tworzymy funkcję:

```
21 def dataPlace(root, lcd):...
25     # Funkcja obsługująca kliknięcia
26 def klikanie(dataPlace, symbol):
27     def f():
28         # tu ma być obsługa przycisków
29     return f
30     # Przyciski kalkulatora
31 def klawisze(root, lcd):
32     przyciski = [tk.Button(root, text=symbol, bg='#abcdef', borderwidth=0) for symbol in symbole]
33     j = len(lcd) + 2
34     for i in range(len(przyciski)):
35         if i % 6 == 0:
36             j += 1
37         margin = 20 if len(symbole[i]) == 1 else 12 # regulacja szerokości przycisków
38         przyciski[i].grid(row=j, column=i % 6, ipady=5, ipadx=margin)
39         przyciski[i].configure(command=klikanie(dataPlace, przyciski[i]['text']))
40     klawiszRowny = tk.Button(root, text='=', bg='#ffff00', borderwidth=0)
```

Funkcja obsługi przycisków (usuwanie – cofnij, usuwanie – wszystko 'C'):

```
28 def klikanie(dane, symbol):
29     def f():
30         if symbol == "\u21BA": #cofnij o jedno pole
31             bufor = dane.get()[:-1]
32             dane.delete(0, tk.END)
33             dane.insert(0, bufor)
34         elif symbol == "C":
35             dane.delete(0, tk.END) #wyczyść
36         else:
37             tekst = symbol if symbol != "x^2" else "^2"
38             dane.insert(tk.END, tekst)
39     return f
```

Pozostała funkcja obliczająca wyniki:

```

41 def oblicz(dane, lcd, info):
42     def czyPoprawnyOstatniZnak(tekst):
43         i = 1
44         while tekst[-i] == ")":
45             i += 1
46         return tekst[-i].isdigit()
47     def czyWielokrotneOperatory(tekst):
48
49         for i in range(len(tekst)):
50             if not tekst[i].isdigit() and not tekst[i + 1].isdigit():
51                 return True
52         return False
53     def zamienZnakPotegi(tekst):
54         for i in range(len(tekst)):
55             if tekst[i] == "^":
56                 tekst = tekst[:i] + "**" + tekst[i + 1:]
57         return tekst
58
59     def f():
60         tekst = dane.get()
61         if not czyPoprawnyOstatniZnak(tekst) or czyWielokrotneOperatory(tekst):
62             info["text"] = "Błędne wyrażenie"
63         else:
64             for i in range(1, len(lcd)):
65                 if lcd[i]["text"]:
66                     lcd[i - 1]["text"] = lcd[i]["text"]
67             if "^" in tekst:
68                 wyrażenie = zamienZnakPotegi(tekst)
69                 lcd[-1]["text"] = tekst + " = " + str(eval(wyrażenie))
70             else:
71                 lcd[-1]["text"] = tekst + " = " + str(eval(tekst))
72     return f

```

Ostatnią rzeczą jest obsługa klawisza '=' którą dodajemy do obsługi klawisza:

```

73 # Przyciski kalkulatora
74 def klawisze(root, lcd, info):
75     przyciski = [tk.Button(root, text=symbol, bg='#abcdef', borderwidth=0) for symbol in symbole]
76     j = len(lcd) + 2
77     for i in range(len(przyciski)):
78         if i % 6 == 0:
79             j += 1
80         margin = 21 if len(symbole[i]) == 1 else 10
81         przyciski[i].grid(row=j, column=i % 6, ipady=5, ipadx=margin)
82         przyciski[i].configure(command=klikanie(dane, przyciski[i]["text"]))
83     klawiszRowny = tk.Button(root, text="=", bg="#ffff00", borderwidth=0, command=oblicz(dane, lcd, info))
84     klawiszRowny.grid(row=len(lcd) + 6, column=4, columnspan=2, ipady=5, ipadx=50)
85     return przyciski

```

