

Лабораторна робота №7

Створення GUI на Python за допомогою бібліотеки Tkinter.

Мета роботи - вивчити основні прийоми програмування програми з графічним інтерфейсом користувача.

Завдання:

Розробити програму (скрипт) мовою Пітон, з використанням відповідних пакетів, що використовує GUI інтерфейс.

Програма має побудувати графічне інтерфейсне вікно:

- для введення у графічному вікні значень інтервалу змінюючи значення змінної x $[a, b]$
- для введення двох функціональних залежностей $y_1=f_1(x)$ та $y_2=f_2(x)$ у вигляді текстового рядка формул

Програма на підставі введеної інформації має виконати:

- перевірку на допустимість вихідних даних та у разі помилки видати відповідне попередження
- побудувати та вивести вікно, що містить графіки заданих функцій. Поле легенди має містити текст формул двох функціональних залежностей.

Графічне інтерфейсне вікно повинно містити щонайменше дві «кнопки» - БУДІВНИТИ_ГРАФІКИ та ЗАВЕРШИТИ_РОБОТУ

Примітка:

Ви можете скористатися наступним скриптом, наприклад, для програмування функціональної частини своєї програми. Рекомендується розібратися в алгоритмі його роботи та обов'язково перевірити його працездатність.

Зверніть увагу на `z[i]=eval(fu)`

```
import sys
import numpy as np
import matplotlib.pyplot as plt
import warnings
def user(xxx):
    '''
    повертає numpy масив значень функції користувача
    від аргументів, що передаються як numpy вхідний масив xxx
    функція користувача запитується з консолі
    у разі помилки - видає повідомлення про помилку та
    запитує іншу функцію користувача
    '''
    #print("введіть f(x)=", end='')
    Flag=True
    while True:
        print("введіть f(x)=", end='')
        fu=input() #"введіть f(x)="
        z=np.empty(xxx.size,dtype=float)
        for i in range(0,len(xxx)):
            x=xxx[i]
            with warnings.catch_warnings():
                warnings.filterwarnings('error')
                try:
                    z[i]=eval(fu)
                    #print(i,z[i])
                except Exception as e:
                    #print('e=',e)
```

```

        print("помилка", e, \
              "\nпрозрахунку значень введеної " \
              "функції f(x)=", fu, "'')
        Flag=False
        break

    if (Flag):
        break
    Flag=True
    #print(z)
    return z
# Варіант використання для побудови графіка,
# введених користувачем функцій
import matplotlib as mpl
import numpy as np

while True:
    try:
        a, b = map(int, input('вкажіть значення a та b - інтервалу аргументу x').split())
    except ValueError:
        print('***\nЦе не числа. повторіть введення значень')
        continue
    break
# print(a, b); exit(0)
x = np.linspace(a, b, 1000)
plt.plot(x, user(x), label='користувача1')
plt.plot(x, user(x), label='користувача2')

```

```
plt.grid(True) #Cітка
plt.xlabel('x label')
plt.ylabel('y label')
plt.title("Simple Plot")
plt.legend()
plt.savefig('pic_5.pdf', fmt='pdg')

plt.show()
```

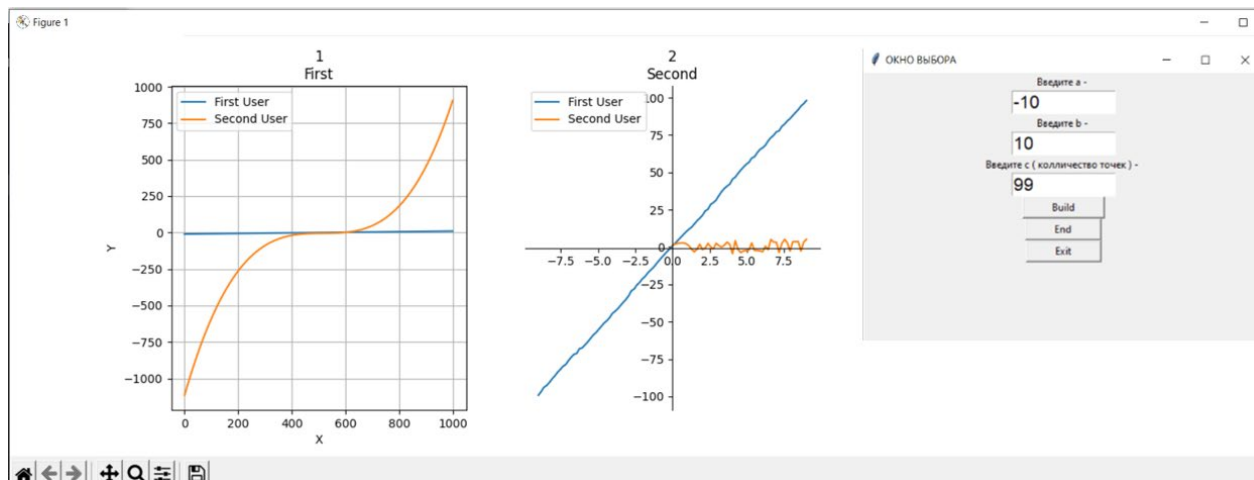
Успіхів у роботі.

приклад

Програма на підставі введеної інформації має виконати:

- перевірку на допустимість вихідних даних та у разі помилки видати відповідне попередження
- побудувати та вивести вікно, що містить графіки заданих функцій. Поле легенди має містити текст формул двох функціональних залежностей.

Графічне інтерфейсне вікно повинно містити щонайменше дві «кнопки» - БУДІВНИТИ_ГРАФІКИ та ЗАВЕРШИТИ_РОБОТУ



```
import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
from tkinter import *
```

```
while True:
    try:
        a=-10
        b=10
```

```

except ValueError:
    print('***\nЦе не числа. повторіть введення значень')
    continue
break
def start():
    import matplotlib as mpl
import numpy as np
import matplotlib.pyplot as plt
з tkinter import *
root = Tk()
root.title('ВІКНО ВИБОРУ')
root.geometry("480x320")

def start():
    fig = plt.figure()
    fig.set_figheight(5)
    fig.set_figwidth(70)
    plt.subplot(1,3,1)
    x = np.linspace(a, b, 1000)
    y= x*x*xx*x+x-5
    plt.plot(x, label='First User')
    plt.plot(y, label='Second User')
    plt.grid(True) #Сітка
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.title("1\nFirst")
    plt.legend()
    x = np.linspace(-9,9,999)

```

```
y = x-np.sin(x+np.cos(x*x*x+15))+x*10
leg1='Graph'
plt.subplot(1,3,2)
plt.title("2\nSecond")
plt.plot (x, y,
label="First User")
plt.grid()
ax = plt.gca()
ax.spines['left'].set_position('center')
ax.spines['bottom'].set_position('center')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.plot()
plt.legend(loc='upper left')
```

```
x = np.linspace(-9,9,999)
y =np.cos(x*x*x)*np.log(1+9*x)+np.cos(x*x)
leg1='Graph'
plt.subplot(1,3,2)
plt.title("2\nSecond")
plt.plot (x, y,
label="Second User")
plt.grid()
ax = plt.gca()
ax.spines['left'].set_position('center')
ax.spines['bottom'].set_position('center')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
```

```
plt.plot()
plt.legend(loc='upper left')
plt.show()
def delete():
    plt.close()
Label(root,text='Введіть a - ').pack()
a=Entry(root,width=10,font='Arial 16')
a.pack()
Label(root,text='Введіть b - ').pack()
b=Entry(root,width=10,font='Arial 16')
b.pack()
Label(root,text='Введіть step C - ').pack()
ccc=Entry(root,width=10,font='Arial 16')
ccc.pack()
butt1 = Button (root, text="Build", command=start)
butt1.pack()
butt2 = Button (root, text = "End", command = delete)
butt2.pack()
butt3 = Button (root, text = "Exit", command = root.destroy)
butt3.pack()
root.mainloop()
```