

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАТЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Програмування

Лабораторна робота №8

«Розробка програми з графічним інтерфейсом на основі бібліотеки tkinter»

Виконав:
студент групи ІО-32
Крадожон М. Р.
Номер у списку групи: 16
Перевірив:
Пономаренко А. М.

Київ – 2023

Лабораторна робота №8

Тема: «Розробка програми з графічним інтерфейсом на основі бібліотеки tkinter».

Мета: ознайомитися з організацією графічного інтерфейсу на основі бібліотеки tkinter. Графічний інтерфейс (GUI) та його елементи. Модуль tkinter.

Загальне завдання:

1. Вивчити матеріал лекцій 24, 25, 26 та 27.
2. Виконати індивідуальне завдання лабораторної роботи, вибране відповідно до варіанту.

$$D = (A \cap (B \cup A)) \cap (\overline{C \cup B}) \cap \bar{C}$$

Короткі теоретичні основи:

Tkinter - це пакет для Python, призначений для роботи з бібліотекою Tk. Бібліотека Tk містить компоненти графічного інтерфейсу користувача (graphical user interface - GUI), написані на мові програмування Tcl.

Під графічним інтерфейсом користувача (GUI) маються на увазі всі ті вікна, кнопки, текстові поля для введення, скролери, списки, радіокнопки, прапорці та ін., Які ви бачите на екрані, відкриваючи ту чи іншу програму. Через них ви взаємодієте з програмою і керуєте нею. Всі ці елементи інтерфейсу разом будемо називати віджетами (widgets).

В даний час майже всі програми, які створюються для кінцевого користувача, мають GUI. Рідкісні програми, які передбачають взаємодію з людиною, залишаються консольними.

Завдання:

Програма повинна складатися з 3-х вікон.

Вміст вікна №1

1. Головне меню, яке повинно включати меню виклику вікна №2, вікна №3.
2. Віджети виводу П.І.Б студента, номера групи, номера у списку та віджет виводу результатів обчислення варіанту відповідно до програми, що задана у завданні – пункт загального порядку виконання лабораторної роботи).
3. Віджети для задавання кількості елементів множин A, B і C.
4. Віджети для формування випадковим чином множин A, B і C з заданою кількістю елементів.
5. Віджети, що дають можливість ручного вводу множин A, B і C.
6. Віджет для задавання діапазону цілих чисел, з яких будемо вибирати елементи

множин.

Вміст вікна №2

1. Віджети для відображення елементів множин A, B і C.
2. Віджети запуску покрокового виконання виразу у відповідності з варіантом. Одним кроком вважати виконання однієї логічної операції.
3. Віджети відображення множин-операндів та множини-результату кожної логічної операції.
4. Віджет відображення множини D та віджет для виконання команди збереження даного результату у файлі.
5. Віджет, який містить зображення виразу 1 у відповідності з варіантом

Вміст вікна №3

1. Віджет для відображення елементів множин D.
2. Віджети для запуску випадкового генерування множини F з такими даними:
 - кількість елементів множини F має дорівнювати кількості елементів множини D;
 - мінімальний елемент множини F має дорівнювати мінімальному елементу множини D;
 - максимальний елемент множини F має дорівнювати максимальному елементу множини D.
3. Віджет для відображення елементів множин F.
4. Віджет відображення множини X, та віджет для виконання команди збереження даного результату у файлі.
5. Віджет, який містить зображення виразу 2 у відповідності з варіантом

Роздруківка коду:

```
from tkinter import *
import random
from tkinter import filedialog as fd

def generate_array():
    outputA.delete(0, END)
    outputB.delete(0, END)
    outputC.delete(0, END)
    size_A = int(entr_A.get())
    size_B = int(entr_B.get())
    size_C = int(entr_C.get())
    start_rand = int(range_array1.get())
    end_rand = int(range_array2.get())
    arrayA = {random.randint(start_rand, end_rand) for i in range(size_A)}
    while (len(arrayA) < size_A):
        arrayA.add(random.randint(start_rand, end_rand))
    arrayB = {random.randint(start_rand, end_rand) for i in range(size_B)}
    while (len(arrayB) < size_B):
        arrayB.add(random.randint(start_rand, end_rand))
    arrayC = {random.randint(start_rand, end_rand) for i in range(size_C)}
    while (len(arrayC) < size_C):
        arrayC.add(random.randint(start_rand, end_rand))
    outputA.insert(0, arrayA)
    outputB.insert(0, arrayB)
```

```

outputC.insert(0, arrayC)
print(str(arrayA))
print(arrayB)
print(arrayC)

```

```

def manual_input():
    outputA.delete(0, END)
    outputB.delete(0,END)
    outputC.delete(0, END)
    arrayA_man = entr_A1.get()
    arrayB_man = entr_B1.get()
    arrayC_man = entr_C1.get()
    arrayA_man = {int(i) for i in arrayA_man.split(' ')}
    arrayB_man = {int(i) for i in arrayB_man.split(' ')}
    arrayC_man = {int(i) for i in arrayC_man.split(' ')}
    outputA.insert(0, arrayA_man)
    outputB.insert(0, arrayB_man)
    outputC.insert(0, arrayC_man)
    print(arrayA_man)
    print(arrayB_man)
    print(arrayC_man)

```

```

def change():
    if r_var.get() == 0:
        generate_array()
    elif r_var.get() == 1:
        manual_input()

```

```

def generate_arrayD():
    outputD.delete(0, END)
    arrayA = outputA.get()
    arrayB = outputB.get()
    arrayC = outputC.get()
    arrayA = {int(i) for i in arrayA.replace("{", "").replace("}", "").split(' ')}
    arrayB = {int(i) for i in arrayB.replace("{", "").replace("}", "").split(' ')}
    arrayC = {int(i) for i in arrayC.replace("{", "").replace("}", "").split(' ')}
    arrayD = (arrayA and (arrayB or arrayA)) and (arrayC or arrayB) and arrayC
    outputD.insert(0, arrayD)
    print(arrayD)

```

```

def extractD():
    file_name = fd.asksaveasfilename(filetypes=(("TXT files", "*.txt"),
                                                ("HTML files", "*.html;*.htm"),
                                                ("All files", "*..*")))

    f = open(file_name, 'w')
    s = outputD.get()
    f.write(s)
    f.close()

```

```

def generate_arrayF():
    outputF.delete(0, END)
    copyD = outputD.get()
    copyD2 = {int(i) for i in copyD.replace("{", "").replace("}", "").split(' ')}
    rangeF = len(copyD2)
    minD = min(copyD2)
    maxD = max(copyD2)
    arrayF = {random.randint(minD, maxD) for i in range(rangeF)}
    while (len(arrayF) < len(copyD2)):
        arrayF.add(random.randint(minD,maxD))
    print(arrayF)

```

```

arrayF.remove(min(arrayF))
arrayF.remove(max(arrayF))
arrayF.add(minD)
arrayF.add(maxD)
print(arrayF)
outputF.insert(0, arrayF)

root = Tk()
root.title('Генератор множин')
w = root.winfo_screenwidth()
h = root.winfo_screenheight()
w = w // 2
h = h // 2
w = w - 675
h = h - 400
root.geometry('1300x750+{}+{}'.format(w, h))
root.configure(bg='#F8F8FF')

Label(root, width=15, height=2, bg='cadetblue2', text='П. І. Б: ', relief="solid").grid(row=0, column=4, pady=5)
Label(root, width=20, height=2, bg='cadetblue2', text='Крадожон Максим', relief="groove").grid(row=0, column=5, pady=5,
padx=(0, 150))
Label(root, width=15, height=2, bg='cadetblue2', text='Номер групи:', relief="solid").grid(row=1, column=4, pady=5)
Label(root, width=20, height=2, bg='cadetblue2', text='ІО-32', relief="groove").grid(row=1, column=5, pady=5, padx=(0,
150))
Label(root, width=15, height=2, bg='cadetblue2', text='Номер у списку:', relief="solid").grid(row=2, column=4, pady=5)
Label(root, width=20, height=2, bg='cadetblue2', text='16', relief="groove").grid(row=2, column=5, pady=5, padx=(0,
150))

Label(root, width=20, height=2, bg='cadetblue2', text='Множина А:', relief="solid").grid(row=1, column=2, pady=5)
entr_A = Entry(root, width=30, relief="groove")
entr_A.grid(row=1, column=3, pady=5, ipady=2, padx=(2, 150))
Label(root, width=20, height=2, bg='cadetblue2', text='Множина В:', relief="solid").grid(row=2, column=2, pady=5)
entr_B = Entry(root, width=30, relief="groove")
entr_B.grid(row=2, column=3, pady=5, ipady=2, padx=(2, 150))
Label(root, width=20, height=2, bg='cadetblue2', text='Множина С:', relief="solid").grid(row=3, column=2, pady=5)
entr_C = Entry(root, width=30, relief="groove")
entr_C.grid(row=3, column=3, pady=5, ipady=2, padx=(2, 150))
Label(root, width=20, height=2, bg='cadetblue2', text='Діапазон чисел:', relief="solid").grid(row=4, column=2, pady=5)
range_array1 = Entry(root, width=30, relief="groove")
range_array1.grid(row=4, column=3, pady=5, ipady=2, padx=(2, 150))
range_array2 = Entry(root, width=30, relief="groove")
range_array2.grid(row=5, column=3, pady=5, ipady=2, padx=(2, 150))

Label(root, width=20, height=2, bg='cadetblue2', text='Множина А:', relief="solid").grid(row=1, column=0, pady=5)
entr_A1 = Entry(root, width=30, relief="groove")
entr_A1.grid(row=1, column=1, pady=5, ipady=2, padx=(2, 150))
Label(root, width=20, height=2, bg='cadetblue2', text='Множина В:', relief="solid").grid(row=2, column=0, pady=5)
entr_B1 = Entry(root, width=30, relief="groove")
entr_B1.grid(row=2, column=1, pady=2, ipady=2, padx=(2, 150))
Label(root, width=20, height=2, bg='cadetblue2', text='Множина С:', relief="solid").grid(row=3, column=0, pady=5)
entr_C1 = Entry(root, width=30, relief="groove")
entr_C1.grid(row=3, column=1, pady=5, ipady=2, padx=(2, 150))

r_var = BooleanVar()
r_var.set(0)
r1 = Radiobutton(text='Генерація випадкових множин', variable=r_var, value=0, relief="ridge", width=40, height=2,
bg='cadetblue2').grid(row=0, column=2, pady=5, columnspan=2)
r2 = Radiobutton(text='Ввести множини', variable=r_var, value=1, relief="ridge", width=40, height=2,
bg='cadetblue2').grid(row=0, column=0, pady=5, columnspan=2)

```

```

Label(root, width=40, height=2, bg='cadetblue2', text='Результати: ', relief="ridge").grid(row=5, column=0, pady=5,
columnspan=2)
Label(root, width=20, height=2, bg='cadetblue2', text='Множина A:', relief="solid").grid(row=7, column=0, pady=5)
outputA = Entry(root, width=30, relief="groove")
outputA.grid(row=7, column=1, pady=5, ipady=2, padx=(2, 150))
Label(root, width=20, height=2, bg='cadetblue2', text='Множина B:', relief="solid").grid(row=8, column=0, pady=5)
outputB = Entry(root, width=30, relief="groove")
outputB.grid(row=8, column=1, pady=5, ipady=2, padx=(2, 150))
Label(root, width=20, height=2, bg='cadetblue2', text='Множина C:', relief="solid").grid(row=9, column=0, pady=5)
outputC = Entry(root, width=30, relief="groove")
outputC.grid(row=9, column=1, pady=5, ipady=2, padx=(2, 150))

b_generate = Button(text="Згенерувати множини", width=40, height=2, bg='cadetblue2', relief="raised",
command=change).grid(row=4, column=0, pady=5, columnspan=2)

Label(root, width=40, height=2, bg='cadetblue2', text='Множина D:', relief="ridge").grid(row=12, column=0, pady=5,
columnspan=2)
imD = PhotoImage(file="arrayD.png")
Label(root, image=imD, relief="ridge").grid(row=13, column=0, pady=5, columnspan=2)
b_arrayD = Button(text="Згенерувати множину D", width=40, height=2, bg='cadetblue2', relief="raised",
command=generate_arrayD).grid(row=14, column=0, pady=5, columnspan=2)

outputD = Entry(root, width=60, relief="solid")
outputD.grid(row=15, column=0, pady=5, ipady=2, columnspan=2)

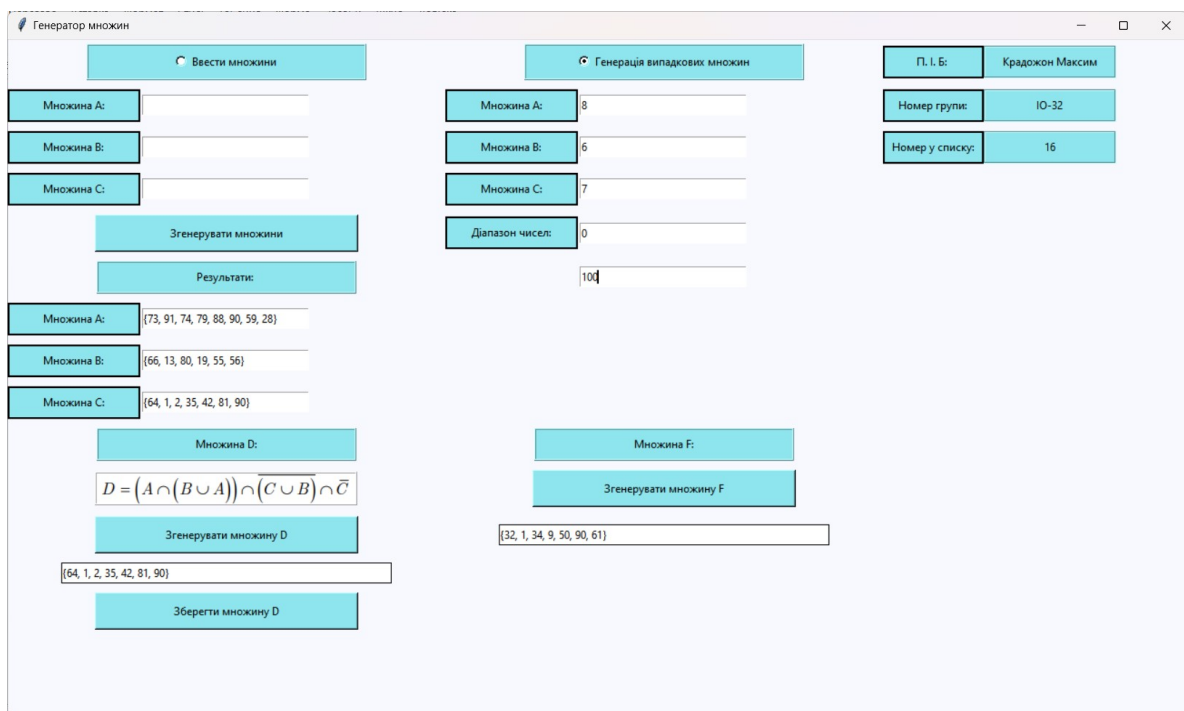
b_saveD = Button(text="Зберегти множину D", width=40, height=2, bg='cadetblue2', relief="raised",
command=extractD).grid(row=16, column=0, pady=5, columnspan=2)

Label(root, width=40, height=2, bg='cadetblue2', text='Множина F:', relief="ridge").grid(row=12, column=2, pady=5,
columnspan=2)
b_arrayF = Button(text="Згенерувати множину F", width=40, height=2, bg='cadetblue2', relief="raised",
command=generate_arrayF).grid(row=13, column=2, pady=5, columnspan=2)
outputF = Entry(root, width=60, relief="solid")
outputF.grid(row=14, column=2, pady=5, ipady=2, columnspan=2)

root.mainloop()

```

Скріншоти:



Висновок: Виконавши цю лабораторну роботу, я зміг здобути відповідні навички в розробці програми з графічним інтерфейсом на основі бібліотеки tkinter в Python. Під час виконання лабораторної роботи проблем не виникало, а складність була в структуруванні коду та приведенні його до більш гарного вигляду.