Text

Description automatically generated **Ministerul Educației și Cercetării al Republicii Moldova**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**RAPORT**

Lucrare de laborator nr. 3,4,5

la cursul „Grafica pe Calculatorul”

**Tema3:**„ Tipuri de date în Python.Liste”

**Tema4:**„ Operatori logici și de comparație. Expresii’’

**Tema5:** ,,Funcții’’

A efectuat : **st. gr.TI-214 Buza Cătălin**

A verificat: **asis. univ. Toma Olga**

**Chișinău 2022**

**Cuprins**

INTRODUCERII………………………………………………………………………………..2-3

SARCINI..........................................................................................................................................4

REZOLVAREA SARCINIILOR.......................................................................................................5

REZULTATUL OBȚINUT…………………………………………………………………………6

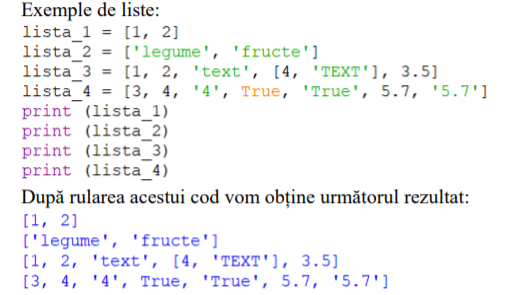
CONCLUZII…………………………………………………………………………………………7

BIBLIOGRAFIE……………………………………………………………………………………8

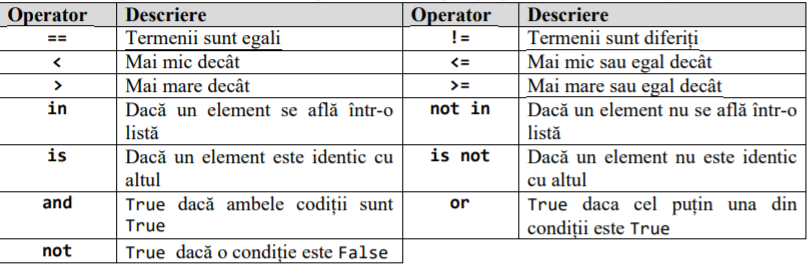
# **Introducere**

O listă în Python este o structură de date mutabilă care poate conține mai multe elemente separate prin virgulă. Listele reprezintă un mod de organizare a datelor oricât de mare și care se scriu între paranteze pătrate, fiecare element (articol) din listă are de obicei un indice, în Python acest indice începe de la 0.

Pentru cei familiari cu limbaje precum C, C++, Java, C# etc, listele pot fi privite drept vectori (array) în acele limbaje, diferența fiind că, un vector poate păstra doar elemente care aparțin unui singur tip de date, pe când listele pot păstra elemente de tipuri diferite.



Pentru a putea folosi o condiție în codul nostru avem nevoie de o comandă condiționată. Acestea folosesc condiții pentru a verifica dacă ceva este True (Adevărat) sau False (Fals). În funcție de rezultat, se vor urma căi diferite în program.

În Python putem folosi următorii operatori de comparație pentru a crea o condiție

**Operatori de comparație:** == != < > <= >=

**Operatori logici:** and or is is not not

**Expresii condiționate. Instrucțiunile if elif else**

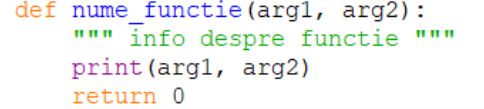
**IF** – o instrucțiune condiționată care testează dacă o condiție este True. În acest caz computerul va urma instrucțiunea care urmează după if.

**ELSE** – o instrucțiune condiționată care îi transmite computerului ce cale să urmeze atunci când condiția if nu este îndeplinită.

**ELIF** – este prescurtarea de la else-if. Astfel instrucțiunea condiționată elif vine întotdeauna după un if și intră în acțiune dacă condițiile lui if nu au fost îndeplinite. Are propria sa condiție și rulează doar dacă aceasta a fost îndeplinită. Ne vom folosi de instrucțiunea elif atunci când avem mai multe condiții de verificat.

O funcție ne permite gruparea mai multor linii de cod cu scopul de a reduce duplicarea codului.

Putem defini o funcție astfel:



În acest cod def este cuvântul cheie care indică unei noi funcții, nume\_functie este

numele funcției ales de noi (cu aceleași restricții și recomandări ca denumirea unei variabile), iar între

paranteze vor fi precizați parametrii funcției (dacă nu există, lăsăm parantezele goale).

Comentariul delimitat de ghilimele triple (""") și precizat imediat sub definiția funcției poartă numele de și este folosit pentru a oferi detalii despre funcția în care se află (detaliile includ informații despre rolul funcției, tipurile de date și descrierea argumentelor și tipul de date returnat). Prezența unui *docstring* nu este obligatorie, însă aceasta poate face codul mult mai clar și poate ușura menținerea acestuia. Mai mult, prezenta unui pentru fiecare funcție este recomandată și de standardul PEP8.

***Sarciniile:***

***Ex3\_4***

Dacă avem lista lista\_mea = ['Vera', ['Ion', 'Vlad', 'Petru'], 'Ana', 'Ala']. Scrieți codul care după rulare să ne dea următorul răspuns:

1. Petru;
2. [’Ion’,’Vlad’,’Petru’];
3. Ala

***Ex4\_4***

Să presupunem că avem următorul dicționar my\_dict = {'key 1': 1, 'key 2': 7, 'key 3': 9}, trebuie să creăm un cod care la răspuns să ne dea o listă cu valorile cheilor din dicționar care sunt mai mari decât 5.

***Ex5\_4***

Trebuie să realizăm o funcție care să redea după rulare factorialul numărului 4, iar numărul 4 să fie introdus de utilizator (4x3x2x1=24). După rularea codului răspunsul trebuie să arate în felul următor:

1

***Rezolvarea sarcinii***

#Lucrare de laborator nr.3 la grafica calculatorului

print("Ex 3\_4")

lista\_mea=['Vera',['Ion','Vlad','Petru'],'Ana','Ala']

print("Trebuie de scris codul ca dupa rulare sa obtinem:a)Petru;b)['Ion','Vlad','Petru'];c)Ala")

print("\na)")

print(lista\_mea[1][2])

print("\nb)")

print(lista\_mea[1])

print("\nc)")

print(lista\_mea[3])

#Lucrare de laborator nr.4

print("Ex 4\_4")

my\_dict={'key 1':1, 'key 2':7, 'key 3':9}

lista=[]

for key,val in my\_dict.items():

    if val>5:

        lista=lista+[val]

print(lista)

#Lucrare de laborator nr.5 Buza Catalin TI-214

def factorial (n):

    facto=1

    for i in range(1,n+1):

        facto\*=i

    return facto

numar=int(input("Doriti sa determinati factorialul numarului:"))

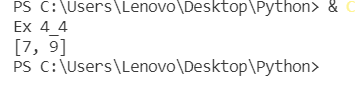
factorialul\_numarului=factorial(numar)

print(f'Factorialul lui {numar} este {factorialul\_numarului}')

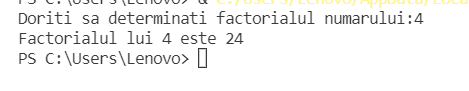
***Rezultatul laborator 3***



***Rezultat laborator 4***



***Rezultat laborator 5***

******

***Concluzii:***

În timpul efectuării lucrări de laborator **nr.3** la ”Grafica pe Calculator” am învățat despre liste în limbajul Python. Am aflat că listele sunt structuri de date mutabile care pot conține mai multe elemente separate prin virgulă.Liste pot păstra elemente de diferite tipuri de date spre deosebire de vectorii.Am învățat că selectarea unui element dintr-o listă poate fi realizată cu operatorul [] iar pentru selectarea mai multor elemente putem folosi notația slice.

În timpul efectuării lucrări de laborator **nr.4** la ”Grafica pe Calculator” am învățat despre operatorii condiționali,logici și instrucțiunile if elif și else în limbajul Python. M-am familiarizat cu modul de lucru al acestor instrucțiuni și momentul când trebuie să folosesc operatorii condiționali sau cei logici.

În timpul efectuării lucrări de laborator **nr.5** la ,,Grafica pe Calculator’’ am învățat cum să-mi creez propiile funcții în limbajul Python.Am învățat cum să folosesc expresia lambda.Am învățat care sunt variabilele globale și locale și cum să le deosebesc.

***Bibliografie***

1)Tema2\_NoteCurs.pdf ;

2)Tema1\_NoteCurs.pdf;

3)Tema3\_NoteCurs.pdf;

4)Tema4\_NoteCurs.pdf;

5)Tema5\_NoteCurs.pdf;