CURS 4

Liste (clasa list)

- Listă = o secv. mutabilă de elem. eterogene indexate de la 0

ex: l = [ ]
l = [1, 2, 3, 4]
l = [1, "grau", 3.14, True]
l = [1, "grau", [1, 2, 3], True]

$\underbrace{[1,2,3]}_{l[3]} \rightarrow l[3][3]$

- List comprehension:

L = [x for x im range(5)] = [0, 1, 2, 3, 4]

L = [k**2 for k im range(10) if k%2==0] = [0, 4, 16, 36, 64]

- list=imput ("listă: ") 5 -14 10 100 -2

șiruri = listă.split() ["5", "-17", "10", "100", "-2"]

limie = [imt (șir) for șir im șiruri] [5, -17, 10, 100, -2]

listă = [imT (șir) for șir im imput ("listă: ").split()]

- Suma cifrelor unui nr:

print ("Suma cfr= ", sum([imt (c) for c im imput ("nr= ")]))

1725

[1, 7, 2, 5]

OPERATORI

① Concatenare „+":

L=[1,2] + [3,4,5] => L=[1,2,3,4,5]

② Op „*":

L=[1,2]*3 => L=[1,2,1,2,1,2]

③ Operatori relaționali: ==, !=, in, not in, <, <=, >=, >

L1 = [1, 2, 8, 7, 3]
L2 = [1, 2, 5, 3]
| => L1 < L2 = False

8<5 Fale (se oprește la primul el. dif, acc. poz)

L3 = [1, 20, "test", 3, 4]
L5 = [1, 10, 7, "da", 3, 7]
| => L3 > L5 = False

↳ se oprește aici, dacă compara "test" > 7 → eroare

Funcții Predefinite

① len (listă)

len([1, 2, [4, 5]]) = 3

② list (secv)

ex: L = list("test") => L = ["t", "e", "s", "t"]

L = list(123) => eroare

③ min (list) / max (list)

ex: min ([1, 2, 3, 5, [6, 7, 8]]) → eroare

④ sum (listă)

ex: sum ([1; 3,14 ; True]) = 5,14 (True e considerat 1)

Metode din clasa list

① count (valoare)

ex: L = [1, 2, [1, 2, 3]]
k = L.count(2)
| → k = 1

② append (valoare)

```
ex: L= [1,2]
    L.append (3) => L= [1,2,3]
    L.append ( 4,5 ) => eroare
    L.append ([4,5]) => L= [1,2,3,[4,5]]
```

③ extend (secvență)

```
ex: L=[1,2]
    L.extend (3) => eroare
    L.extend ([3,4,5]) => L= [1,2,3,4,5]
```

CREAREA UNEI LISTE

- L= [int (x) for x in input("Elem: "). split ()]
- L= []

-
  ```
  for x in input (" Elem: "). split () :
          L. append (int (x))
  ```

-
  ```
  for x in input (" Elem: "). split ():
          L += int (x)      // sau L= L+ [int (x)]
  ```

④ index (valoare)

```
try:
    print (" Apare pe poz: ... ", L. index(x))
except:
    print (" Nu există ")
```

⑤ remove (valoare)

⑥ pop (imdex)

```
L = [1,2,3,4]
x. pop(
```

⑦ imsert (poziție , valoare)

```
L= [1, 2, 3, 4]
L.imsert (2, 100)      =>  L=[1,2,100,3,4]
L.imsert (1000, 100)   =>  L = [1,2,3, 4, 100]
```

⑧ clear()

```
L. clear ()    sau    L=[]
```

⑨ sort () (sortează crescător)

```
L= [1,2,5,9, 8, 7, 4,6,3]
L.sort()   =>  L= [1,2,3,... 9]
```

COPIEREA UNEI LISTE

1) L2 = L1. copy()   shallow copy

```
L1 = [1,2,[3,4,5]]
L2 = L1. copy()
```

2) import copy   (foarte lentă metoda asta)

```
L2 = copy. deepcopy (L1)
```