**Лабораторна робота Списки.**

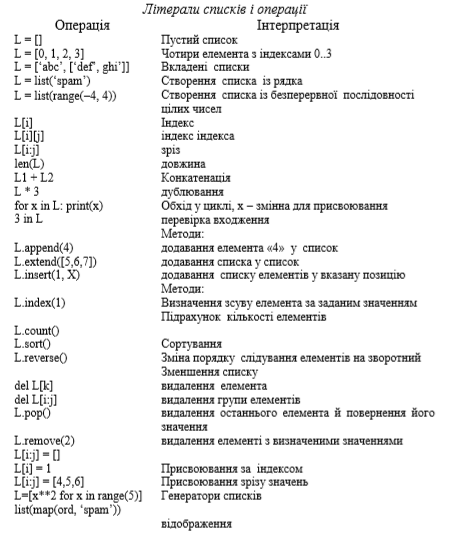
**Список -** колекція інших об'єктів, змінюваний об'єкт, вони можуть бути вкладеними, збільшуватися і зменшуватися, містити об'єкти будь-яких типів. Завдяки спискам можна створювати і обробляти в своїх сценаріях структури даних будь-якого ступеня складності.

Нижче наводяться основні властивості списків, списки в мові Python - це:

**1. Впорядковані колекції об'єктів довільних типів.**

**2. Доступ до елементів за зсувом.** Можна використовувати операцію індексування для отримання окремих об'єктів зі списку за їхнім зсувом.

**3. Змінна довжина, гетерогенність і довільна кількість рівнів вкладеності.** Списки можуть збільшуватися і зменшуватися безпосередньо (їх довжина може змінюватися), вони можуть містити не тільки односимвольні рядки, а й будь-які інші об'єкти (списки гетерогенні). Списки можуть містити інші складні об'єкти, вони підтримують можливість створення довільної кількості рівнів вкладеності, тому є можливість створювати зі списків списки списків. **4. Відносяться до категорії змінних об'єктів.**

****

**Список** – впорядкована послідовність певних значень, які можуть повторюватися. Для створення списку необхідно записати його елементи через кому у квадратних дужках

Методи списків.

Об'єкти списків в Python підтримують специфічні методи, багато з яких змінюють сам список безпосередньо:

>>> L.append('please’) # Виклик метода додавання елемента у кінець списка >>> L

['eat’, 'more', 'SPAM!’, 'please']

>>> L.sort() # Сортування елементів списка ('S'<'e')

>>> L

['SPAM!’, 'eat', 'more’, 'please']

* - - - -

>>> L = [‘abc’, ‘ABD’, ‘aBe’]

>>> L.sort() # Сортування з урахуванням регістру символів

>>> L

[‘ABD’, ‘aBe’, ‘abc’]

>>> L = [‘abc’, ‘ABD’, ‘aBe’]

>>> L.sort(key=str.lower) # Приведення символів до нижнього регістру

>>> L

[‘abc’, ‘ABD’, ‘aBe’]

>>>

>>> L = [‘abc’, ‘ABD’, ‘aBe’]

>>> L.sort(key=str.lower, reverse=True) # Змінює напрямок сортування

>>> L

[‘aBe’, ‘ABD’, ‘abc’]

* - - - -

методи списків дозволяють видаляти елементи з певними значеннями (remove), вставляти елементи у визначену позицію (insert), визначяти зсув елемента за заданим значенням (index) тощо:

>>> L = [‘spam’, ‘eggs’, ‘ham’]

>>> L.index(‗eggs‘)

# індекс об‘єкта 1

>>> L.insert(1, ‘toast’)

# Вставка у потрібну позицію

>>> L

[‘spam’, ‘toast’, ‘eggs’, ‘ham’]

>>> L.remove(‘eggs’) # видалення елемента із визначеним значенням

>>> L

[‘spam’, ‘toast’, ‘ham’]

>>> L.pop(1) # видалення елемента у вказаній позиції ‘toast’

>>> L

[‘spam’, ‘ham’]

**Приклади**

**int\_list = [1,2,3,5] # список із чотирьох цілих чисел**

**char\_list = ['a', 'c', 'z', 'x'] # список із чотирьох символів**

**empty\_list = [] # empty\_list – пустий список**

**print('Список чисел:', int\_list)**

**print('Список символів:', char\_list)**

**print('Пустий список:', empty\_list)**

**Результат**

**Список чисел: [1, 2, 3, 5]**

**Список символів: ['a', 'c', 'z', 'x']**

**Пустий список: []**

**Значення елемента** можна отримувати за його індексом (номером). Індексація починається з нуля.

**# Створення списку чисел**

**my\_list = [5, 7, 9, 1, 1, 2]**

**# Виведення першого елемент**

**print (my\_list [0]);**

**print (my\_list)**

**# Введення індексу**

**index = int (input ('Введіть номер елемента:'))**

**# Отримання відповідного елемента**

**element=my\_list[index]**

**# Виведення його значення на екран**

**print (element)**

**Результат [5, 7, 9, 1, 1, 2]**

**Введіть номер елемента: 2**

**9**

**3. Якщо використати від‘ємні індекси, то обхід елементів розпочинається з останнього. Індекс останнього елемента списку – «–1», передостаннього – «–2».**

**# Створення списку чисел**

**my\_list = [5, 7, 9, 1, 1, 2]**

**# Отримання передостаннього значення**

**pre\_last = my\_list [-2] # pre\_last == «1»**

**print (pre\_last)**

**# Обчислення суми першого і останнього значень**

**result = my\_list[0] + my\_list[-1];**

**print(result)**

**Результат**

**1**

**7**

**Методи обробки списків:**

**– метод append -** додає значення в список

my\_list = [] **# Створення пустого списку**

my\_list.append(3);

my\_list.append(5)

my\_list.append(my\_list[0]+my\_list[1])

print(my\_list) # Виведення списку на екран

Результат

[3, 5, 8]

**- видалення елемента списку**

my\_list = [5, 1, 5, 7, 8, 1, 0, -23] # Створення списку чисел

print(my\_list) # Виведення списку

# Оператор del видаляє заданий елемент

del my\_list[2]

print(my\_list) # Виведення списку

Результат

[5, 1, 5, 7, 8, 1, 0, -23]

[5, 1, 7, 8, 1, 0, -23]

**Заміна елементу списка**

my\_list = [5, 1, 5, 7, 8, 1, 0, -23] # Створення списку чисел

print(my\_list) # Виведення списку

length = len(my\_list) # Отримання довжини списка

# Введення индексу

index = length

while not -length <= index < length:

index=int(input('Введіть індекс ел-та списку (від %d до……………….. %d):'

% (-length, length - 1)))

# Введення нового значення

value = int(input('Введіть нове значення заданого ел-та: '))

my\_list[index]=value # зміна елемента списку

print(my\_list) # Виведення списку на екран

Результат

[5, 1, 5, 7, 8, 1, 0, -23]

Введіть індекс елемента списку (від -8 до 7): 7

Введіть нове значення заданого елемента: 5

[5, 1, 5, 7, 8, 1, 0, 5]

**Виведення квадратів чисел зі списку**

my\_list = [5, 1, 5, 7, 8, 1, 0, -23] # Створення списку чисел

for x in my\_list:

print('{}^2={}'.format(x, x \*\* 2))

Результат

5 ^ 2 = 25

1 ^ 2 = 1

5 ^ 2 = 25

7 ^ 2 = 49

8 ^ 2 = 64

1 ^ 2 = 1

0 ^ 2 = 0

-23 ^ 2 = 529

**Функція-конструктор, яка створює список із значення іншого типу** empty\_list=list() # Створення пустого списку

print(empty\_list)

# Створення списку з послідовності

range numbers = list(range(10))

print(numbers)

# Створення списку символів із рядка

chars = list("a string");

print(chars)

Результат

[]

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

['a', ' ', 's', 't', 'r', 'i', 'n', 'g']

**Числа Фібоначчі**

n=10 # Кількість чисел в послідовності

# Список чисел Фібоначчі (напочатку має дві одиниці)  
 fibs = [1, 1]

# Повторюємо (n–2) рази, тому що два числа вже є в списку

for i in range(n-2):

# Додаємо суму двох останніх чисел

fibs.append(fibs[i]+fibs[i+1])

print(fibs) # Виведення списку на екран

Результат

[1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

**Зріз списка – отримання групи елементів за їхніми індексами**

my\_list = [5, 7, 9, 1, 1, 2] # Створення списку чисел

# Отримання зрізу списка від нульового (першого) елемента

# (включаючи його) до третього (четвертого) (не включаючи)

sub\_list = my\_list[0:3]

print(sub\_list) # Виведення отриманого списку

# Виведення елементів списка від другого до переостаннього

print(my\_list[2:-2])

# Виведення ел-тів списка від 4-ого (5-ого) до 5-ого (6-ого)

print(my\_list[4:5])

Результат

[5, 7, 9]

[9, 1]

[1]

**Зріз з використанням кроку**

my\_list = [5, 7, 9, 1, 1, 2] # Вибір кожного другого ел-та списку (починаючи з першого),

# не включаючи останній елемент

sub\_list = my\_list[0:-1:2]

print(sub\_list) # виведення отриманого списку

# Виведення елементів від 2-ого (3-ього) до передостаннього

# з кроком 2

print(my\_list[2:-2:2])

# Виведення ел-тів списка, крім першого, в зворотному порядку print(my\_list[-1:0:-1])

Результат

[5, 9, 1]

[9]

[2, 1, 1, 9, 7]

**Для перевірки входження елемента в список використовують операцию in**

Перевірка, чи є задане число у списку

my\_list = [5, 7, 9, 1, 1, 2] # Введення значення

value=int(input('Введіть число: '))

if value in my\_list:

print('Число входить у список')

else:

print('Число не входить у список')

Результат

Введіть число:

5

Число входить у список

**Визначення кількості елеметів списка**

my\_list = [1, 5, 1, 3, 7, 8, 124]

print(len(my\_list))

Результат

7

**Нехай задано два списки цілих випадкових чисел від 0 до 5: [а1, …, an] і [b1, …, bn], n=10.** Написати програму їх формування.

Вивести списки на екран.

import random a=[random.randint(0,5)

for i in range(0,10)]

b=[random.randint(0,5)

for j in range(0,10)]

print(a);

print(b)

Результат



**Завдання 1.** Задано текст, в якому є дві і більше однакові літери. Скласти програму, яка визначає і виводить на екран найбільшу кількість однакових символів, розташованих підряд.

**Завдання 2.** Задано множину цілих чисел від «1» до «50». Скласти програму, яка визначає, скільки з них є числами Фібоначчі і скільки чисел, в запису яких перша значуща цифра дорівнює «1» або «2».

**Завдання 3.** Задано множину символів від 'a' до 'z': Ww Xx Aa Ss Tt Uu Vv Ee Ff Gg Hh Ww Xx Yy Zz Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz Скласти програму, яка визначає і виводить на екран елементи цієї множини в алфавітному порядку.

**Завдання 4.** Задано список-матриця цілих випадкових чисел (додатних та від‘ємних) [[а11, …, a1n], …, [аm1, …, amn]]. Написати програму, яка визначить список мінімальних елементів кожного рядка (результат записати в інший список). Вивести на екран початкову матрицю та отриманий список.

**Завдання 5** Потрібно згенерувати список із цілих випадкових чисел та нулів [а1, …, an]. Написати програму визначення елементів, розміщених після першого нульового. Вивести на екран початковий та отриманий списки.