

СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

**122 «Комп'ютерні науки»
КН-18**

2019 / 2020 навчальний рік

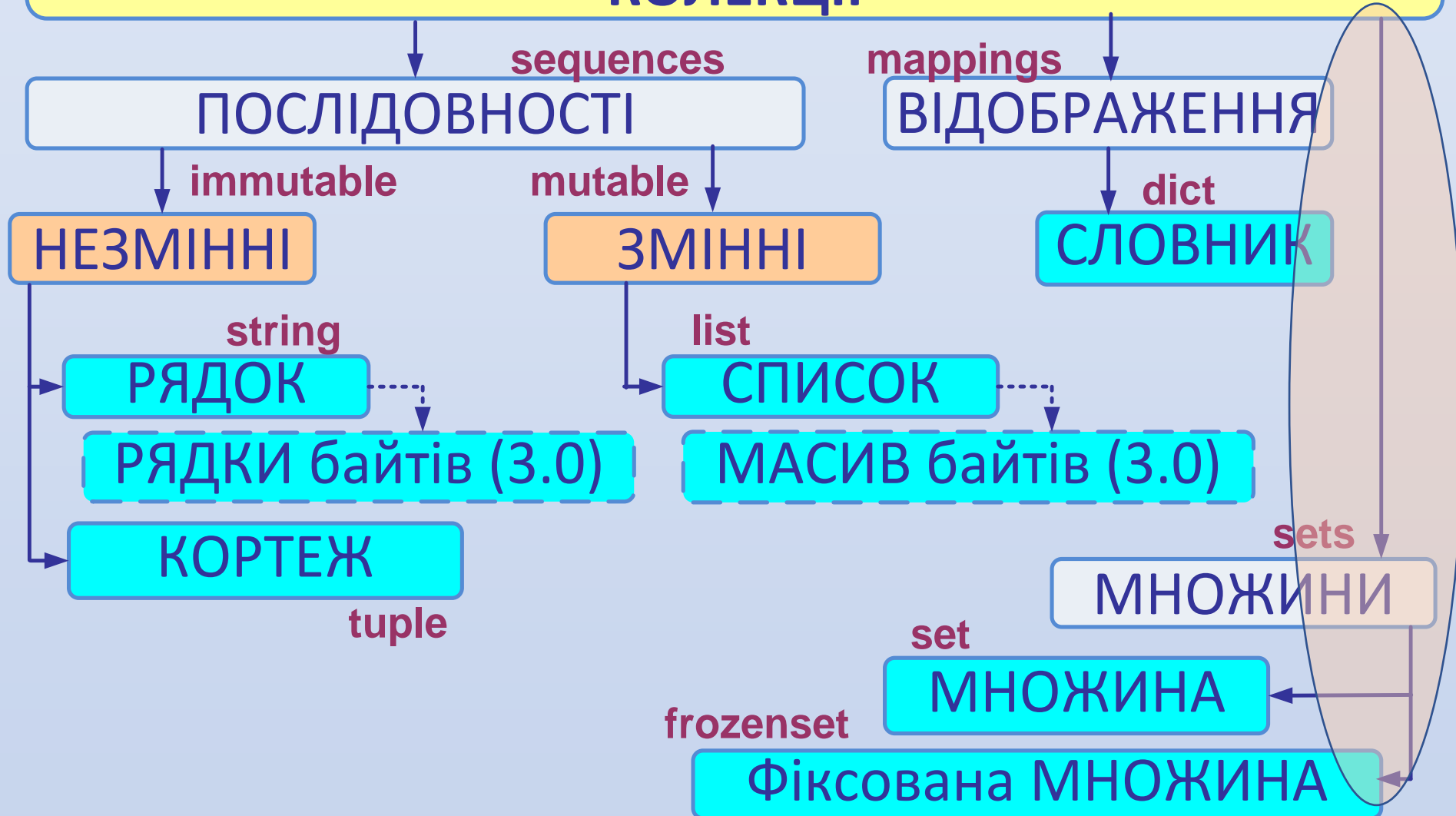
PYTHON #4

1. Множина (set)
2. Файли в Python
3. Огляд інструкцій Python
4. Структура програми

https://github.com/eabshkvprof/2020_Mod_Prog_Tech

МНОЖИНА

КОЛЕКЦІЇ



МНОЖИНА (> 3.0)

Множина → НЕупорядкована змінна - set (незмінна - frozenset)) колекція унікальних об'єктів довільного типу

Тип	Змінність	Індексованість	Унікальність	Створення
set	+	-	+	{el1, el2, } set()
frozenset	-	-	+	frozenset()

МНОЖИНА. Створення

Створення множини

	Дія
<code>St={el1 , el2 , , }</code>	Множина елементів
<code>St=set('gsgjsdh')</code>	Множина елементів
<code>St=set([<i>iterable</i>])</code>	Множина елементів (ітератор)
<code>St=frozenset([<i>iterable</i>])</code>	Фіксована множина елементів (ітератор)

СЛОВНИКИ. Базові операції/функції

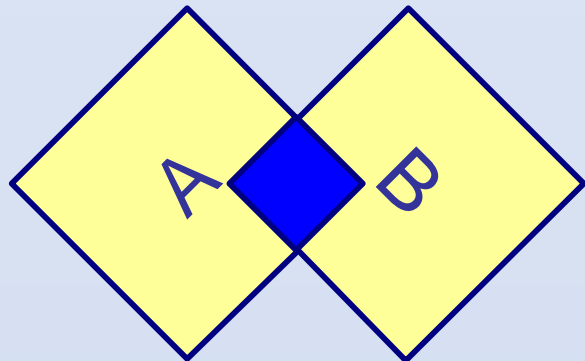
Функція	Дія
print(St)	<i>Друкування об'єктів множини</i>
len(St)	<i>Кількість об'єктів в множині</i>
el in St	<i>Перевірка на входження об'єкту до множини</i>
for el in St	<i>Виклик усіх елементів множини</i>

МНОЖИНА. Методи

Метод	Дія
<code>St.copy()</code> <code>Fst.copy()</code>	<i>Вертає копію множини</i>
<code>St.clear()</code>	<i>Видалення всіх елементів множини</i>
<code>St.add(el)</code>	<i>Додає <code>el</code> до множини</i>
<code>St.discard(el)</code>	<i>Видаляє <code>el</code>, якщо він є</i>
<code>St.remove(el)</code>	<i>Видаляє <code>el</code>, якщо відсутній <code>KeyError</code></i>
<code>St.pop()</code>	<i>Видаляє перший елемент та вертає його</i>

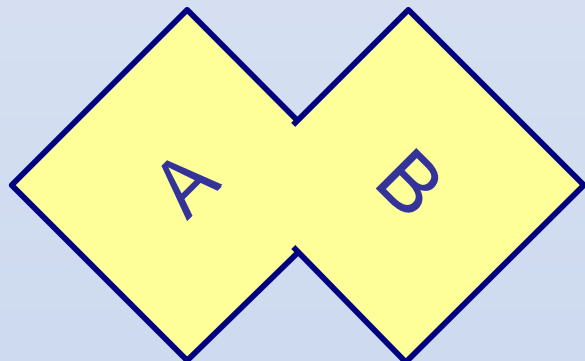
Метод	Дія
<code>set_a.isdisjoint(set_b)</code>	<i>Set_A неперерізний з Set_B ???</i>
<code>set_a.issubset(set_b)</code>	<i>Set_A підмножена Set_B ???</i>
<code>set_a.issuperset(set_b)</code>	<i>Set_A надмножена Set_B ???</i>

МНОЖИНИ. Методи



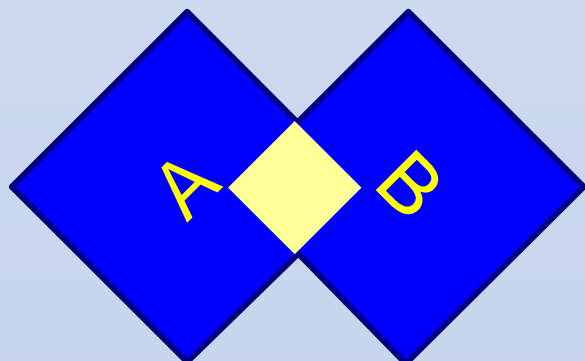
Метод

```
set_c=set_a.intersection(set_b)  
set_c=set_a & set_b
```



Метод

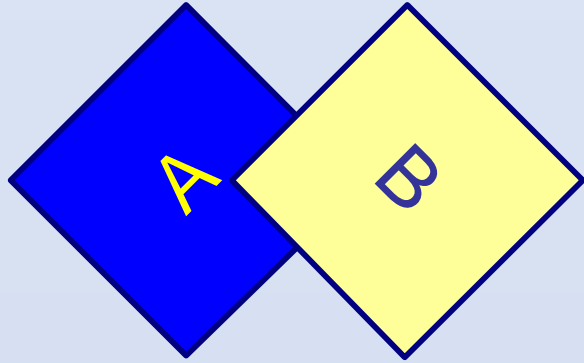
```
set_c=set_a.union(set_b)  
set_c=set_a | set_b
```



Метод

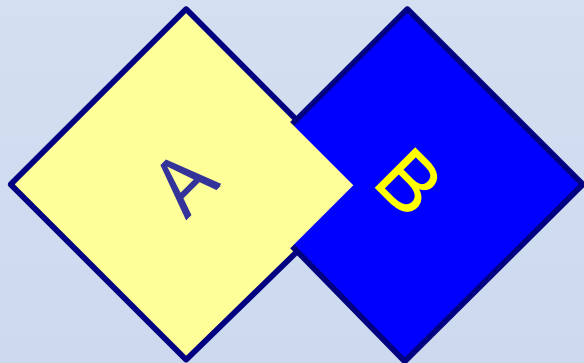
```
set_c=set_a.symmetric_difference(set_b)  
set_c=set_a ^ set_b
```


МНОЖИНИ. Методи



Метод

```
set_c=set_a.difference(set_b)
```



Метод

```
set_c=set_b.difference(set_a)
```

МНОЖИНИ. Методи

Метод (тільки для set)	Дія
set_a.union_update(set_b)	Змінює set_a
set_a.intersection_update (set_b)	
set_a.difference_update (set_b)	
set_a.symmetric_difference_update (set_b)	

ФАЙЛ

Файл – іменована область постійної пам'яті в комп'ютері, якою управляє операційна система.

Python → Файл – вбудований тип об'єкту, що, забезпечує можливість доступу до цих областей пам'яті.

Об'єкт типу «файл» - має тільки методи та атрибути !

Файлові методи (операції)

Функція . Метод	Опис
Open ()	Створює об'єкт типу файл – «відкриває» файл
f.close()	Закриття відкритого файлу
f.Readable() f.Read(), f.readline(), f.readlines()	Перевірка можливості читання Читання
f.Seekable(), f.Seek()	Перевірка можливості зміни позиції
f.Tell()	Вертає поточну позицію
f.Writable(), f.write(), f.writelines()	Перевірка можливості запису Запис до файлу
f.next()	Вертає наступний рядок

https://www.tutorialspoint.com/python/file_methods.htm

https://www.w3schools.com/python/python_ref_file.asp

Файлові атрибути

Атрибут	Опис
f.closed	Індикатор поточного статусу файлу
f.encoding	Спосіб кодування/декодування файлу
f.mode	Режим відкриття файлу
f.name	Ім'я відкритого файлу
f.newlines	Універсальний режим завершення рядку ???

https://www.tutorialspoint.com/python/file_methods.htm

https://www.w3schools.com/python/python_ref_file.asp

Файлові методи (операції)

Вбудована функція	Опис
Open ()	Створює об'єкт типу файл – «відкриває» файл

Open(file, mode, buffering, encoding, errors, newline, closed, opener)

- *file* – path to file.
- *mode* – режим ('r' - читання, 'w' - запис, 'a' - додавання в кінець, 'b' – байтовий режим, 't' - текст, '+' – (r + w)). *За замовчуванням 'rt' – читання тексту.*
- *buffering* – управління буфером;
- *encoding* – режим кодування/декодування;
- *errors* – управління реакцією на помилки;
- *newline* – управління режимом «новий рядок»;
- *closed* – управління дескриптором файлу;
- *opener* – опис обробника помилки.

Файлові методи (операції)

Вбудована функція	Опис
Open ()	Створює об'єкт типу файл – «відкриває» файл

Важливо: *'r', ...* текст – рядки типу **str** – виконується автоматичне кодування / декодування.

'b', – послідовність байтів – деяких змін.

my_file = open ('my_text_file.txt', 'w') – відкриває файл для запису (тексту!)

my_file = open ('my_text_file.txt') – відкриває файл читання (тексту!)

my_file = open ('my_text_file.bin', 'wb') – відкриває файл для запису (послідовності байт)

my_file = open ('my_text_file.bin', 'rb') – відкриває файл читання (послідовності байт)

Файлові методи (операції)

Метод	Опис
f.readable () f.read ()	Зчитування файлу

f.readable() – повертає **True**, коли файл відкрито для читання.

f.read(*size*) - зчитує деяку кількість даних і повертає їх у вигляді рядка (у текстовому режимі) або об'єкта байтів (у двійковому режимі); *size* – максимальна кількість даних, що зчитуються (необов'язковий числовий аргумент).

f.readline() - зчитує з файлу один рядок; символ нового рядка (**\n**) залишається в кінці рядка і опускається лише в останньому рядку файлу.

f.readlines() - зчитує всі рядки до списку.

Файлові методи (операції)

Метод	Опис
f.writable () f.write ()	Зчитування файлу

f.writable() – вертає **True**, коли файл відкрито для запису.

f.write(string) – записує вміст рядка до файлу і повертає кількість записаних символів.

f.writelines(string) – запис наступного елементу списку (рядок) до файлу (файл відкрито з *'wa'*)

!!! Інші типи об'єктів потрібно перетворити - або в рядок (в текстовому режимі), або в байт-об'єкт (у двійковому режимі) - перед їх записуванням.

Файлові методи (операції)

Метод	Опис
f.tell () f.seek ()	Вертає поточну позицію в файлі Змінює поточну позицію в файлі

f.tell () – вертає ціле число, що дає поточне положення об'єкта файлу у файлі, представлене як кількість байтів від початку файлу (у двійковому режимі) і номер поточного символу в текстовому режимі.

f.seek (*offset*, *whence*) – нова позиція обчислюється від додавання *offset* до *whence* (опорної точки).

whence →

= 0 вимірює від початку файлу (за замовчуванням),

= 1 поточне положення файлу,

= 2 кінець файлу.

СТРУКТУРА PYTHON ПРОГРАМИ

Ієрархія програми →

Програма складається з **пакетів та (або) модулів** .

Пакет – Логічно завершена сукупність **модулів** .

Модуль – функціонально завершений фрагмент програми (оформлений як єдиний файл) - складається з **інструкцій** .

Інструкції складаються з **виразів** .

Вирази створюють та обробляють **об'єкти** .

СТРУКТУРА PYTHON ПРОГРАМИ



ІНСТРУКЦІЇ PYTHON

Інструкція	Дія	Приклад
import	Доступ до модуля	Import math
from	Доступ до атрибутів модуля	From sys import stdin
class	Створення об'єкту	Class sub(super): def ...
try/except/finally	Обробка виключень	Try: action () except : print()
raise	Створення виключення	
assert	Перевірки для налаштування	
with/as	Менеджер контексту	With open('file') as myfile: proc(myfile)
del	Видалення посилань	

ІНСТРУКЦІЇ PYTHON

Інструкція	Дія	Приклад
Присвоювання	Створення посилань	A = "gglskfsj"
Виклики	Виклик функції	F = open()
if / elif / else	Вибір	If 'z' in text: print(text)
For / else	Перебір послідовності	for z in mlist: print(z)
while / else	Цикл загального призначення	while x>y: print('Qu')
pass	Пуста інструкція	while True: pass

ІНСТРУКЦІЇ PYTHON

Інструкція	Дія	Приклад
break	Вихід з циклу	While True: if <i>cond</i> : break
continue	Перехід на початок циклу	While True: if <i>cond</i> : continue
def	Створення функцій та методів	Def foo (a): print (a)
return	Повернення результату	Def foo (a): return (a)
yield	Функції-генератори	Def gen (n): for i in n: yield (i)
global	Простір імен	
nonlocal	Простір імен (3.0)	

СТРУКТУРА PYTHON ПРОГРАМИ

**Вкладеність блоків
регулюється
відступами**

БЛОК 0

інструкція :

◀ **ВІДСТУП** ▶

БЛОК 1

інструкція:

◀ **ВІДСТУП** ▶

Блок 2

БЛОК 1

БЛОК 0

Рекомендована ЛІТЕРАТУРА

- **Програмування числових методів мовою Python:** підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2014. – 640 с.
- **Програмування числових методів мовою Python:** навч. посіб. / А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий, Є. В. Глушко ; за ред. А. В. Анісімова. – К. : Видавничо-поліграфічний центр "Київський університет", 2013. – 463 с.
- **Основи програмування Python:** Підручник для студ. спеціальності 122 «Компютерні науки» / А.В.Яковенко; КПІ.- Київ: КПІ, 2018 . – 195 с.
- **Лутц М.** Изучаем Python, 4-е издание. - СПб.: Символ-Плюс. 2011.- 1280 с.: ил.

Контрольні запитання

- Надайте визначення **множини** в мові Python, вкажіть властивості множини, варіанти створення множини. Наведіть приклади.
- Надайте перелік основних операцій із **множиною**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних функцій об'єктів **множина**, вкажіть їх призначення та наведіть відповідні приклади.
- Надайте перелік основних методів об'єктів типу **множина**, вкажіть їх призначення та наведіть відповідні приклади.

The END
Mod 1. Lec 4.