

Лабораторна робота № 7

**Використання неупорядкованих колекцій. Опрацювання множин.
Асоціативні колекції. Опрацювання словників.**

Мета роботи: ознайомлення з функціоналом неіндексованих колекцій та асоціативних списків та їх застосуванням.

Короткі теоретичні відомості

Множина — це сукупність унікальних даних, де елементи набору не можуть дублюватися. Створити множину у Python можна просто записавши її елементи у фігурних дужках {}, розділяючи їх комою. Ця сукупність може містити будь-яку кількість елементів, і вони можуть бути різних типів (ціле число, дійсне число, кортеж чи рядок тексту). Проте множина не може містити елементів, що самі є множинами, списками чи словниками. Наприклад:

```
# create a set of integer type
student_id = {112, 114, 116, 118, 115}
print('Student ID:', student_id)

# create a set of string type
vowel_letters = {'a', 'e', 'i', 'o', 'u'}
print('Vowel Letters:', vowel_letters)

# create a set of mixed data types
mixed_set = {'Hello', 101, -2, 'Bye'}
print('Set of mixed data types:', mixed_set)
```

Коли елементи множини виводяться на екран функцією **print**, порядок слідування їх елементів може бути, взагалі кажучи, довільним, оскільки множини не мають заданого порядку їх елементів.

Створити порожню множину за допомогою порожніх фігурних дужок не можна, бо порожні фігурні дужки {} у Python створюють іншу порожню колекцію — словник. Щоб створити множину без елементів, можна скористатися функцією set() без аргументів. Наприклад:

```
# create an empty set
empty_set = set()

# check data type of empty_set
print('Data type of empty_set:', type(empty_set))
```

Множини містять набір лише унікальних елементів, спроба включити до множини елемент, який вона вже містить приведе до того, що множина не

зміниться. Додавання повторюваних елементів при створенні множини приведе до тієї ж ситуації, наприклад:

```
numbers = {2, 4, 6, 6, 2, 8}

print(numbers)    # буде виведено {8, 2, 4, 6}
```

Тут ми бачимо, що в наборі немає повторюваних елементів, оскільки набір не може містити дублікатів.

Набори змінні. Однак, оскільки вони не впорядковані, індексація не має сенсу. Ми не можемо отримати доступ або змінити елемент набору за допомогою індексування або зрізу.

Для додавання елемента до множини у Python використовується метод `add()`. Наприклад,

```
numbers = {21, 34, 54, 12}

print('Initial Set:', numbers)

# using add() method

numbers.add(32)

print('Updated Set:', numbers)
```

Метод `update()` використовується для доповнення множини елементами інших типів колекцій (списків, кортежів, наборів тощо). Наприклад,

```
companies = {'Lacoste', 'Ralph Lauren'}

tech_companies = ['apple', 'google', 'apple']

companies.update(tech_companies)

print(companies)

# Output: {'google', 'apple', 'Lacoste', 'Ralph Lauren'}
```

Методом **`discard()`** можна видалити вказаний елемент із набору, наприклад:

```
languages = {'Swift', 'Java', 'Python'}

print('Initial Set:', languages)

# remove 'Java' from a set

removedValue = languages.discard('Java')

print('Set after remove():', languages)
```

Для опрацювання множин в Python реалізовано вбудовані функції, такі як **`all()`**, **`any()`**, **`enumerate()`**, **`len()`**, **`max()`**, **`min()`**, **`sorted()`**, **`sum()`**, тощо.

Python Set надає вбудовані методи для виконання математичних операцій з множинами.

Об'єднання двох множин **A** і **B** включає всі елементи множини **A** і **B**. Для об'єднання множин використовується оператор `|` або метод `union()`. Наприклад:

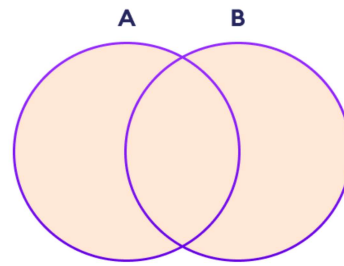
```
A = {1, 3, 5}
```

```
B = {0, 2, 4}
```

```
print('Union using |:', A | B)
```

```
print('Union using union():',
```

```
A.union(B))
```



Перетин двох множин **A** та **B** містить спільні елементи лише множин **A** і **B**. У Python використовується оператор `&` або метод `intersection()`. Наприклад:

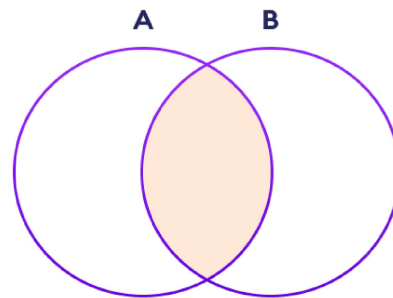
```
A = {1, 3, 5}
```

```
B = {1, 2, 3}
```

```
print('Intersection using &:', A & B)
```

```
print('Intersection using intersection():',
```

```
A.intersection(B))
```



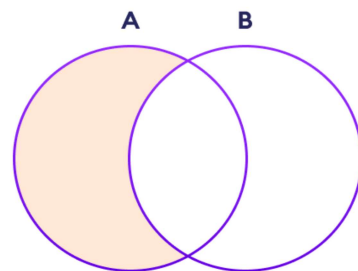
Різниця множин **A** та **B** містить елементи множини **A**, яких немає в множині **B**. Для обчислення різниці множин в Python використовується оператор `-` або метод `difference()`. Наприклад:

```
A = {2, 3, 5}
```

```
B = {1, 2, 6}
```

```
print('Difference using -:', A - B)
```

```
print('Difference using difference():', A.difference(B))
```



Симетрична різниця між двома множинами **A** та **B** складається з усіх елементів **A** та **B** без спільних елементів. Для позначення симетричної різниці використовується оператор `^`, або метод `symmetric_difference()`.

Наприклад,

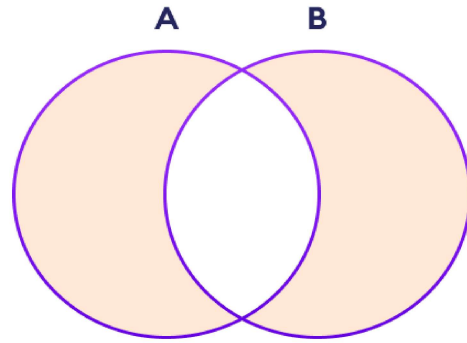
```
A = {2, 3, 5}
```

```
B = {1, 2, 6}
```

```
print('using ^:', A ^ B)
```

```
print('using symmetric_difference():',
```

```
      A.symmetric_difference(B))
```



Для перевірки рівності двох множин можна використовувати оператор `==`.

Для різних практичних задач зручно мати доступ до елементів набору не за його номером, а за деякою назвою. Для таких цілей в Python використовують **словники**. Словник (*dict*) – це структура даних, призначена для зберігання об'єктів з доступом за деяким ключем, взагалі кажучи, довільного типу. Значення ключа в межах словника мають бути унікальними. Ключ повинен мати незмінюваний тип даних: ціле або дійсне число, рядок, кортеж. Словники в Python є реалізацією структури даних, яку інколи ще називають **асоціативним масивом**.

Словник записується, як перелік пар ключ : значення, розділених комою та взятих в фігурні дужки: `{'A1':2, 'A2':3}`. Словник є змінюваним типом даних: в нього можна додавати нові елементи з довільними ключами і вилучати вже існуючі елементи.

Для створення порожнього словника можна скористатися однією з наступних команд:

```
>>> a={}
```

```
>>> b=dict()
```

Словник, що містить вказані елементи створюють інструкціями вигляду:

```
>>> a={'A1':2, 'A2':3}
```

```
>>> b=dict(id1=4, id2=8)
```

```
>>> print(b)
```

```
{'id1': 4, 'id2': 8}
```

```
>>> d={x: x**2 for x in (2, 4, 6)}
```

```
>>> print(d)
{2: 4, 4: 16, 6: 36}
```

Доступ до значень елементів словника можна отримати за його ключем, наприклад:

```
>>> e={'A1':2, 'A2':3}
>>> print(e['A2'])
3
```

Якщо при цьому вказати неіснуючий у словнику ключ, – отримаємо виняток **KeyError**:

```
>>> print(e['A3'])
Traceback (most recent call last):  File "<pyshell#99>",
line 1, in <module>      print(e['A3'])
KeyError: 'A3'
```

Для перевірки наявності значення ключа в словнику можна скористатися оператором **in**:

`<ключ> in <словника>`

Результат оператора буде **in** логічним значенням True/False. Тим же оператором можна перевірити наявність в словнику елемента з вказаним значенням, шукаючи його в наборі значень словника, що є результатом методу **values()**.

Перегляд усіх елементів словника можна виконати за допомогою циклу:

```
d={'A1': 1, 'A2': 3, 'A3': 5, 'A4': 7}
for k in d:
    print(k, d[k])
```

Словник є змінюваним типом даних, за необхідності можна змінювати значення його елементів командою вигляду:

`<ім'я_словника>[<ключ>] = <нове_значення>`

Якщо ж елемента з заданим ключем в словнику нема, то його буде додано до словника. Для видалення елемента зі словника можна скористатися функцією **del**, наприклад:

```
>>> e={'A1':2, 'A2':3}
>>> del e['A1']
>>> print(e)
{'A2': 3}
```

Для опрацювання словників можна користуватися такими методами:

`dict.fromkeys(iterable [, value])` – для створення нового словника з ключами, що є елементами об'єкта **`iterable`** і однаковим для всіх значенням **`value`**.

`dict.get(key[, default])` – для отримання значення елемента за ключем **`key`**. У випадку відсутності елемента з ключем **`key`** повертається значення **`default`**.

`dict.keys()` – для отримання колекції ключів словника.

`dict.values()` – для отримання колекції значень елементів словника.

`dict.items()` – для отримання колекції елементів словника у вигляді кортежів (ключ, значення).

`dict.pop(key[, default])` – для вилучення зі словника **`dict`** елемента з ключем **`key`**. Метод повертає значення елемента, як результат виконання

`dict.clear()` – для вилучення усіх елементів (очищення) словника.

Завдання. 1.

Скласти програму, яка утворює три множини символів А, В, та С із фрагментів тексту, введених користувачем. Запрограмувати обчислення заданих у варіанті виразів з множинами виконати перевірки рівності отриманих множин. У звіт подати:

- умову задачі варіанта;
- складений Python-код;
- результати обчислень для деякого набору вхідних даних.

Варіанти завдань

В.1. $A \setminus B \setminus C = (A \cup B) \setminus (B \cup C);$

В.2. $(A \Delta B) \setminus C = (A \setminus C) \Delta (B \setminus C);$

В.3. $A \cap (B \Delta C) = (A \setminus B) \Delta (A \setminus C);$

В.4. $(A \Delta C) \cup (A \cap B) = A \Delta (C \setminus (A \cap B));$

В.5. $(A \cup B) \Delta (C \setminus A) = (A \cup C) \Delta (B \setminus A);$

В.6. $(A \setminus B) \cup (B \Delta C) = (A \cup B \cup C) \setminus (B \cap C);$

В.7. $(A \cap B) \Delta (A \cap C) = (A \setminus B) \Delta (A \setminus C);$

В.8. $(A \cup B) \Delta C = ((A \Delta C) \setminus B) \cup (B \setminus C);$

В.9. $(A \cup B) \Delta (B \setminus C) = (A \setminus B) \cup (B \cap C);$

B.10. $A \cap (B \Delta C) = (A \cap B) \Delta (A \cap C);$

B.11. $A \setminus (B \Delta C) = (A \setminus B \setminus C) \cup (A \cap B \cap C);$

B.12. $(A \setminus B) \Delta (A \cap B) = (A \setminus B \setminus C) \cup (A \cap B \cap C);$

B.13. $(A \cap B) \Delta C = ((A \Delta C) \cap B) \cup (B \cap C);$

B.14. $(A \Delta B) \setminus C = (A \cup C) \Delta (B \cup C);$

B.15. $((A \Delta C) \setminus B) \cup (A \cap B) = (A \Delta (C \setminus B)) \cup (A \cap B);$

B.16. $A \cup (C \setminus B) = ((A \cup C) \setminus B) \cup (A \cap B);$

Приклад виконання завдання 1

Варіант № 16. Умова задачі: Скласти програму, яка утворює три множини символів A, B, та C із фрагментів тексту, введених користувачем. Запрограмувати обчислення заданих у варіанті виразів з множинами виконати перевірки рівності отриманих множин.

Орієнтовний код програми:

```
# Множина розділових знаків, що вилючаються з тексту
P = {' ', '.', ',', '!', '?', '-'}

str1 = input('Введіть фрагмент тексту для множини A\n')
# Утворюємо з введеної стрічки множину символів,
# виключивши з неї пробіли та розділові знаки
A = set(str1).difference(P)

str2 = input('Введіть фрагмент тексту для множини B\n')
B = set(str2).difference(P)

str3 = input('Введіть фрагмент тексту для множини C\n')
C = set(str3).difference(P)

# Обчислюємо вираз з множинами зліва від =
LeftSet = A.union(C.difference(B))

# Обчислюємо вираз з множинами справа від =
RightSet = ((A.union(C)).difference(B)).union(
    A.intersection(B))

print('A \u22C3 (C \\\\ B) = ', LeftSet)
print('((A \u22C3 C) \\\\ B) \u22C3 (A \u22C2 B) = ',
```

```

    RightSet)

If LeftSet == RightSet:
    print(LeftSet, ' = ', RightSet)

```

Завдання2.

Скласти програму розв'язання задачі відповідно до варіанта. Програма повинна утворювати словники та інші вказані структури даних відповідно до умови задачі, заповнювати їх даними та здійснювати обчислення, задані у варіанті завдання.

1. Відомості про студента складаються з його унікального номера студентського квитка, імені, прізвища, року навчання та середнього бала на основі оцінок, отриманих на останній сесії. Дано словник, який містить відомості про студентів деякого факультету. Скласти програму, яка дозволяє

- визначити чи навчаються на факультеті студенти з однаковими прізвищами;
- визначити середню успішність студентів на кожному курсі;

2. В Інститут філології кожен зі студентів вивчає принаймні декілька іноземних мов. Дані про вивчення мов студентами зберігаються у словнику, у якому у якості значення міститься список мов, які вивчає студент. Заповнити словник даними,

- скласти і вивести список усіх мов, що вивчаються у інституті філології;
- визначити мови, які вивчають всі студенти інституту.

3. Словник містить дані про вакансії, подані ІТ-компаніями в систему пошуку персоналу, ключем є назва компанії та пропонованої позиції, а значеннями список мов програмування та технологій, якими повинен володіти претендент на посаду. Скласти програму для заповнення словника даними та

- пошуку та формування списку вакансій, за обраною технологією;
- підрахунку для усіх технологій сумарних кількостей вакансій, що подані в систему.

4. Дано словник, що у якості ключів містить назви пір року (зима, весна, літо, осінь), а у ролі значень – кортеж з назв місяців року, що належать до цієї пори року. Здано таблицю, що задається словником з ключами номер місяця у році, і значеннями – показниками середньомісячних опадів у деякому регіоні. Скласти програму для заповнення словників даними та знаходження

- пори року на яку припадає найбільша кількість опадів;
- пори року з найбільшими середньомісячними коливаннями кількості опадів.

5. Словник містить дані про замовлення інтернет-магазину, ключем є номер замовлення, а значеннями словник з даними про замовлені товари. Дані про товари в окремому замовленні містяться в словнику, де ключем є назва товару, а значеннями кількість замовлених одиниць цього товару. Скласти програму для заповнення словників даними та знаходження

- сумарної кількості замовлених одиниць кожного товару у всіх замовленнях;
- назви товару, що найчастіше зустрічається в замовленнях.

6. Словник містить дані про замовлення служби доставки, ключем є адреса замовника, а значеннями словник з даними про замовлені товари і їх кількість. Ще один словник містить дані про ціну одиниці кожного товару. Скласти програму для заповнення словників даними та

- побудови словника з загальною вартістю замовлення за кожною адресою;
- назви товару, зсумарна вартість якого в усіх замовленнях найбільша.

7. Студенти університету вивчають факультативні курси за вибором, кожен від трьох до семи предметів. Дані про вичення студентами факультативних курсів містяться у словнику ключем в якому є кортеж з прізвища, імені та групи студента а значеннями список предметів, які вивчає студент.

- утворити та вивести словник у зі списками студентів для вивчення кожного факультативного курсу;
- визначити три предмети, які вивчає найбільша кількість студентів.

8. Перевізник подає в систему продажу квитків дані про кількість пасажирських місць в автобусі, список станцій через які проходить маршрут в порядку їх слідування та список вартостей проїзду від початкової станції до кожної іншої в порядку зростання вартості. Вартість проїзду між внутрішніми станціями маршруту розраховується як різниця вартостей проїзду до відповідних станцій від початкової. Скласти програму для моделювання роботи системи з продажу квитків, виористовуючи словник з даними про заповненість місць в автобусі на кожній станції маршруту. Програма повинна

- надавати користувачеві можливість замовити потрібну кількість квитків від вказаної ним станції до потрібної йому станції призначення, повідомляючи при цьому вартість одного квитка та кількість вільних місць;
- контролювати кількість вільних місць і не дозволяти замовлення квитків з перевищенням кількості пасажирських місць автобуса;
- після завершення циклу з замовлення квитків порахувати виручку перевізника, як загальну суму вартості замовлених квитків.

9. Словник містить дані про розробників, зареєстрованих на біржі фрілансу, ключем є електронна адреса фрілансера, а значеннями список мов

програмування та технологій, з використанням яких фрілансер готовий виконувати роботу. Скласти програму для заповнення словника даними та

- пошуку та формування списку розробників, що можуть працювати з вказаною технологією;

- підрахунку для усіх технологій кількостей фрілансерів, що можуть їх використовувати в своїй роботі.

10. Словник містить дані про результати змагань важкоатлетів: прізвища спортсменів зі списком ваги піднятої ними у кожній з декількох спроб. Скласти програму для заповнення словника даними та знаходження трійки переможців

- за сумарною вагою піднятою атлетом в усіх спробах

- за найкращим результатом серед зарахованих спроб.

11. Слова у рядку розділяються одним або декількома пропусками. За введеним з клавіатури фрагментом тексту побудувати словник частот вживання стів у тексті, за яким знайти-

- частоту вживання кожного слова, тобто відношення кількості входження слова до загального числа всіх різних слів;

- слова, що входять до рядка не менше двох разів;

- слова з n букв, що входять до рядка принаймні m разів (m та n з'являється користувачем з клавіатури).

12. Відомості про автомобіль складаються з його марки, моделі, року випуску об'єму двигуна та прізвища власника. Дано словник, який містить відомості про автомобілі за ключем, що задається номерним знаком авто. Скласти програму для знаходження

- прізвищ власників автомобілів даної марки;

- кількостей автомобілів кожної марки.

13. Реалізувати шифрування та дешифрування послідовності символів, використовуючи для цього словник побудований на основі символів алфавіту та послідовності чисел Фібоначчі. В програмі передбачити ввід користувачем тексту для шифрування, та послідовності чисел для дешифрування тексту

14. Словник моделює стан гаманця в такий спосіб: значення банкноти (номінал) – це ключ, а кількість банкнот цього номіналу – значення. якщо у гаманці відсутня банкнота певного номіналу, то такий елемент відсутній у гаманці (не допускається у словнику наявність пар типу $\{20:0\}$). Скласти програму для заповнення даних про вміст гаманця та підрахунку суми коштів, що міститься у гаманці. Вважається, що повний перелік номіналів банкнот відомий.

15. В Інститут філології кожен зі студентів вивчає принаймні чотири іноземних мови. Дано словник студентів інституту філології у якому у якості значення міститься список мов, які вивчає студент.

- утворити та вивести словник у якому для кожної мови (ключ), знайти скільки студентів її вивчають (значення);

- визначити мови, які вивчає менше чверті студентів.

16. Дані про успішність студентів зберігаються у вигляді словника, ключем якого є прізвище студента, а значенням – словник з пар (назва предмету : оцінка).

- утворити та вивести на словник, що містить рейтинг студентів у вигляді середньої оцінки з усіх предметів;

- створити список прізвищ студентів, упорядкований відповідно до їх рейтингу у порядку спадання від найвищого до найнижчого.

Приклад виконання завдання 1

Варіант № 16. Умова задачі: Дані про успішність студентів зберігаються у вигляді словника, ключем якого є прізвище студента, а значенням – словник з пар (назва предмету : оцінка).

- утворити та вивести на словник, що містить рейтинг студентів у вигляді середньої оцінки з усіх предметів;

- створити список прізвищ студентів, упорядкований відповідно до їх рейтингу у порядку спадання від найвищого до найнижчого.

Орієнтовний код програми:

```
import random
# Список навчальних предметів
subjects = ['Алгеба', 'Математичний аналіз',
            'Загальна фізика', 'Інформатика']
# Список студентів (можна створювати чи доповнювати
# в ході виконання програми)
students = ['Вовк', 'Заєць', 'Ведмідь', 'Лис', 'Лось']
session = {}
for student in students:
```

```

# створюємо список оцінок
marks = {}
for subject in subjects:
    marks[subject] = random.randint(50,100)
# додаємо успішність студента в загальний словник
session[student] = marks
# друкуємо результати сесії
print('Успішність студентів:')
for student in session.keys():
    print(student, ' : ' , end='')
    for subject in session[student].keys():
        print(subject, ' - ',session[student][subject], ';'
        ', end='')
    print()
# формуємо і друкуємо рейтинг
print('Невпорядкований рейтинг')
rating = {}
for student in session.keys():
    print(student, ' : ' , end='')
    sum = 0
    count = len(session[student].keys())
    for subject in session[student].keys():
        sum += session[student][subject]
    rate = sum / count
    rating[student] = rate
    print(rate)
# формуємо список студентів за рейтингом
ratingList = []
for student in rating.keys():
    id = 0

```

Лабораторний практикум з програмування на Python

```
for surname in ratingList:
    if rating[surname]<rating[student]: break
    id += 1
ratingList.insert(id,student)

# друкуємо рейтинг
print('Рейтинг студентів за результатами сесії')
for student in ratingList:
    print(student, ' : ', rating[student])
```

Запитання для самоконтролю

1. Які колекції даних реалізовано у *Python*?
2. Який тип даних використовується для опрацювання множин значень чи об'єктів?
3. Опишіть відмінності між множинами та списками.
4. Які оператори для дій над множинами об'єктів реалізовано в *Python*?
5. Які функції для опрацювання множин реалізовано в *Python*?
6. Як оголосити множину та проініціалізувати її елементами?
7. Як додати до множини новий елемент?
8. Чи передбачено вилучення елементів з множини? Якщо так, то в який спосіб це можна зробити?