

Лабораторна Робота 4

Колекції даних: Кортежі, словники.

Мета роботи – вивчити та засвоїти базові навички використання базових структур даних Python. Дізнатися про принципи взаємодії з кортежами та словниками, а також про основні вбудовані методи для їх обробки.

1. Кортеж

1.1 Базові поняття про кортежі

Список так само може бути незмінним (immutable), як і рядок, в цьому випадку він називається кортеж (tuple). Кортеж використовує менше пам'яті, ніж список. Кортеж замість квадратних дужок використовує круглі (хоча можна і зовсім без дужок). Кортеж не допускає змін, в нього не можна додати новий елемент, хоча він може містити об'єкти, які можна змінити:

```
[4]: t = 1,[2,3]
      print(t)
      t[1] = 2
```

```
(1, [2, 3])
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-4-aeb14024b19c> in <module>
      1 t = 1,[2,3]
      2 print(t)
----> 3 t[1] = 2

TypeError: 'tuple' object does not support item assignment
```

```
[3]: t[1].append(4)
      print(t)
```

```
(1, [2, 3, 4, 4])
```

Функція `tuple()` бере в якості аргументу рядок або список і перетворює його в кортеж:

```
[5]: t=tuple('abc')
      print(t)
      ('a', 'b', 'c')
```

До можливих операцій над кортежами відносяться всі операції над списками, що не змінюють список (додавання, множення на число, методи `index()` і `count()` і деякі інші операції). Можна також по-різному змінювати елементи місцями і так далі.

2. Словник

2.1. Базові поняття про словники

Після списків словник є самим гнучким вбудованим типом. Якщо список - це впорядкована колекція, то словник - неупорядкована. Основні особливості словників:

1. Доступ здійснюється по ключу, а не за індексом. За аналогією зі списком, в словнику можна отримати доступ до елементів в циклі по ключам.
2. Значення словника зберігаються в невідсортованому порядку, більш того, ключі можуть зберігатися не в тому порядку, в якому вони додаються.
3. За аналогією зі списками, словник може зберігати вкладені словники. Словник може зберігати в якості значень об'єкти будь-якого типу (heterogeneous). Ключ в словнику - immutable тип, може бути рядком, цілим числом, float або кортежем, що складається із зазначених типів.
4. Словники реалізовані як хеш-таблиці з швидким доступом.
5. Словники, так само як і списки, зберігають посилання на об'єкти, а не самі об'єкти.

Словник (dictionary) - це асоціативний масив або хеш. Це нерегульована безліч пар ключ: значення з вимогою унікальності ключів. Пара фігурних дужок `{}` створює порожній словник. На відміну від послідовностей, доступ до елементів словника проводиться по ключу, а не за індексом, ключ може бути будь-якого типу, ключ не допускає змін.

Основні операції над словником - збереження з заданим ключем і витяг з нього значення. Також можна видалити пару key: value за допомогою інструкції del.

Метод keys () для словника повертає список всіх використовуваних ключів в довільному порядку; для сортування списку потрібно застосувати метод sort (). Для визначення наявності певного ключа є метод has_key (), який у версії 3.0 встигне застаріти - замість нього є оператор in. Додавання нового об'єкта в словник не вимагає попередніх перевірок: якщо раніше ключу вже відповідало деяке значення, воно буде перезаписано.

```
[10]: tel = {'Lola' : 380950505095, 'BuBochKA' : 380503223323}
      tel['Lapusya'] = 380990767676
      print(tel)
      print(tel['Lapusya'])
      del tel['Lola']
      print(tel)
      print(tel.keys())

{'Lola': 380950505095, 'BuBochKA': 380503223323, 'Lapusya': 380990767676}
380990767676
{'BuBochKA': 380503223323, 'Lapusya': 380990767676}
dict_keys(['BuBochKA', 'Lapusya'])
```

Створити словник можна декількома способами:

Звичайний вираз - це зручно, якщо словник статичний:

```
[12]: T = {'name': 'Alexandro', 'age': 35}
      print(T)

{'name': 'Alexandro', 'age': 35}
```

Динамічний варіант створення на льоту:

```
[13]: D = {}
      D['name'] = 'Alexandro'
      D['age'] = 35
      print(D)

{'name': 'Alexandro', 'age': 35}
```

За допомогою функції `dict ()` - ключі при цьому повинні бути рядками. За допомогою цієї функції можна позбавити себе від обов'язкової умови укласти ключ в лапки.

```
[17]: d1 = dict(id=1948, name="Washer", size=3)
      d2 = dict({"id": 1948, "name": "Washer", "size": 3})
      d3 = dict([("id", 1948), ("name", "Washer"), ("size", 3)])
      d4 = dict(zip(("id", "name", "size"), (1948, "Washer", 3)))
      print(d1,d2,d3,d4, sep='\n')
```

За допомогою `fromkeys ()` - створює словник за списком ключів з порожніми значеннями:

```
[18]: D = {}.fromkeys(['name', 'age'],123)
      print(D)

      {'name': 123, 'age': 123}
```

За допомогою конструктора:

```
[21]: d = dict((x, x**2) for x in range(5))
      print(d)

      {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

2.2. Функції та методи словників

- **dict ()** - створення словника;
- **len ()** - повертає число пар;
- **clear ()** - видаляє всі значення зі словника;
- **copy ()** - створює псевдокопію словника;
- **deepcopy ()** - створює повну копію словника;
- **fromkeys ()** - створення словника;
- **get ()** - отримати значення по ключу;
- **has_key ()** - перевірка значення по ключу;
- **items ()** - повертає список значень;
- **iteritems ()** - повертає ітератор;
- **keys ()** - повертає список ключів;
- **iterkeys ()** - повертає ітератор ключів;
- **pop ()** - витягує значення по ключу;
- **popitem ()** - витягує довільне значення;
- **update ()** - змінює словник;
- **values ()** - повертає список значень;
- **itervalues ()** - повертає ітератор на список значень.
- **in** - оператор, перевіряє наявність значення по ключу;
- **del** - оператор, видаляє пару по ключу;
- **dict ()** - конструює словник за допомогою послідовності.

- 2.3. **З**Оскільки словники є мапи (map), а не послідовності, до них не можна застосувати конкатенацію або зрізи. До словників можна застосовувати стандартні оператори порівняння: <, <=, ==, !=, >=, > .

Для того щоб зробити прохід по ключам словника, використовуємо for:

```
[22]: table = {'Python': 'Guido van Rossum',
              'Perl':    'Larry Wall',
              'Tcl':     'John Ousterhout' }
for lang in table:
    print(lang, table[lang])

Python Guido van Rossum
Perl Larry Wall
Tcl John Ousterhout
```

За допомогою словників можна зберігати структуровану інформацію у вигляді записів:

```
[23]: man = {'name': 'Serg',
            'jobs': ['programmer', 'writer'],
            'web': 'www.iakovlev.org',
            'home': {'city': 'Moscow', 'zip': 129000}}
print(man['name'])
print(man['jobs'][1])

Serg
writer
```

3. Варіанти завдання

Використовуючи словники реалізувати довідник з тематикою відповідно до варіанту. Для довідника реалізувати функції:

- Додавання
- Видалення
- Редагування
- Демонстрації всього довідника

А також розрахункову функцію відповідно до варіанту.

Таблиця – Варіанти завдань

№	Завдання
1.	Довідник - « Аеропорт » Поля - [ПІБ] [Рейс] [Клас] [Місце] [Вартість квитка] Вивести білети з вартістю нижче ніж середня вартість квитка
2.	Довідник - « Пошта » Поля - [ID відправлення] [Відправник] [Одержувач] [Адреса] [Вага] Вивести відправлення з вагою більшою за N (N вводити з клавіатури)
3.	Довідник - « Квітковий магазин » Поля - [Назва квітки] [Кількість на складі] [Вартість за шт.] [Дата поставки] [Термін зберігання] Вивести усі квіти залишок яких на складі менший за N (N вводити з клавіатури)
4.	Довідник - « Транспортні компанії » Поля - [Назва] [Кількість Авто] [Вартість 1км перевезення] [Адреса] [Макс. допустима вага] Вивести компанії яких Макс. допустима вага більша за N (N вводити з клавіатури)
5.	Довідник - « Футбольний Матч » Поля - [Команда 1] [Команда 2] [Рахунок] [Попереджень] [Видаленнь] Вивести матчі у яких суддя діставав картку більше ніж N разів (N вводити з клавіатури)
6.	Довідник - « Магазин Техніки » Поля - [ID товару] [Назва] [Вартість за шт.] [Кількість на складі] [Термін гарантії (місяців, або років)] Вивести усі товари гарантія на які більша за N (N вводити з клавіатури)

7.	Довідник - « Beauty bloggers » Поля - [Нікнейм] [Назва каналу] [Посилання] [Вік] [Кількість підписчиків]
	Вивести N найпопулярніших блогерів за зростанням віку (N вводити з клавіатури)
8.	Довідник - « Кінолог » Поля - [Порода] [Середня вага] [Середній мак. Вік] [Регіон розповсюдження] [Середня вартість]
	Вивести усі породи у яких середня вага менше за N, а середній вік більший за M (M-N ввести з клавіатури)
9.	Довідник - « Бабусині заготовки » Поля - [Назва] [Об'єм] [Рік] [Вид] [Термін придатності]
	Вивести усі заготовки зроблені до N року (N ввести з клавіатури)
10.	Довідник - « Художня галерея » Поля - [Назва полотна] [Автор] [Рік] [Розмір] [Ціна]
	Вивести усі полотна певного автора (автора вводити з клавіатури)
11.	Довідник - « Ремон взуття » Поля - [ID взуття] [дата прийому] [Вид роботи] [Тел. власника] [Ціна]
	Вивести усі прийняті пари взуття за номером телефону власника (номер телефону вводити з клавіатури)
12.	Довідник - « Розклад занять » Поля - [Група] [День тижня] [№ пари] [Аудиторія] [Предмет]
	Вивести усі пари певної групи (Групу вводити з клавіатури)
13.	Довідник - « Бібліотека » Поля - [Назва] [Автор] [Видавництво] [Тираж] [Рік]
	Вивести усі книжки певного видавництва (видавництво вводити з клавіатури)
14.	Довідник - « Орнітолог » Поля - [Назва виду] [Сімейство] [Кількість особин] [Регіон розповсюдження] [Середня вартість]
	Вивести усіх птахів з вартістю більше N (N ввести з клавіатури)
15.	Довідник - « Вокзал » Поля - [Пункт відбуття] [Пункт Прибуття] [Маршрут] [Місце] [Вартість квитка]
	Розрахувати середню вартість квитка
16.	Довідник - « Контакти » Поля - [Ім'я] [Телефон] [Вік]
	Розрахувати середній вік

4. Приклад

Варіант - 16.

Лістинг

```
[*]: person = {}.fromkeys(['name', 'age', 'phone'])
person_list = list()
c=-1
def age(person_list):
    i=0
    s=0
    for i in range(len(person_list)):
        s+=person_list[i]['age']
    return s/len(person_list)
def add(person_list):
    person_list+=person.copy()
    person_list[-1]['name']=input("Введіть ім'я: ")
    person_list[-1]['age']=int(input("Введіть вік: "))
    person_list[-1]['phone']=input("Введіть телефон: ")
def print_p(person_list):
    i=0
    for i in range(len(person_list)):
        print(person_list[i],sep="/n")
while c!=0:
    print("Меню")
    print("1. Додати запис")
    print("2. Переглянути усі записи")
    print("3. Підрахувати середній вік")
    print("0. завершити роботу")
    c=int(input());
    if c==1:
        add(person_list)
    if c==2:
        i=0
        print_p(person_list)
    if c==3:
        print("Середній вік =",age(person_list))
```

Результат роботи

```
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
1
Введіть ім'я: Михайло
Введіть вік: 23
Введіть телефон: 0959595955
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
1
Введіть ім'я: Петро
Введіть вік: 25
Введіть телефон: 0999995999
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
1
Введіть ім'я: Кирило
Введіть вік: 28
Введіть телефон: 0509550999
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
2
{'name': 'Михайло', 'age': 23, 'phone': '0959595955'}
{'name': 'Петро', 'age': 25, 'phone': '0999995999'}
{'name': 'Кирило', 'age': 28, 'phone': '0509550999'}
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
0. завершити роботу
3
Середній вік = 25.333333333333332
..
```

3. Контрольні запитання

1. Які існують можливі операції з кортежами.
2. Якого типу можуть бути ключі в межах одного словника.
3. Якого типу можуть бути значення в межах одного словника.
4. Які існують можливі операції з словниками.
5. Наведіть приклади вбудованих функцій для роботи зі словниками.
6. Опишіть механізми звернення, заміни та видалення елементу словника.
7. Вкажіть способи звернення до елементів вложеного словника.