Лабораторна Робота 5

Python. Файли

Мета роботи – вивчити та засвоїти базові принципи роботи з файлами у Python. Ознайомитися з існуючими режимами доступу, а також з особливостями використання різних методів і функцій для роботи з файлами.

1. Загальні відомості

1.1 Відкриття файлу

Відкрити файл можна за допомогою функції open:

```
open(name[, mode[, buffering]])
```

Функція повертає файловий об'єкт. Обов'язковий тільки перший аргумент. Якщо інші параметри відсутні, файл буде доступний на читання. Таблиця режимів (mode) функції open:

'r' - читання.
'w' - запис.
'a' - додавання.
'b' - бінарний режим.
'+' - читання / запис.

Режим '+' може бути доданий до решти режимів. За замовчуванням python відкриває файли в текстовому режимі. Для відкриття файлу в бінарному режимі на читання можна додати 'rb'. Третій параметр встановлює розмір буферизації при роботі з файлом. За замовчуванням він вимкнений, і читання / запис йде безпосередньо з диска на диск. Для включення буфера третій параметр повинен бути відмінним від нуля.

1.2 Базові файлові методи

У python багато об'єктів є файлами: стандартне введення sys.stdin, стандартний вивід sys.stdout, об'єкти, що відкриваються функцією urllib.urlopen і т.д.

Запис в файл:

```
3]: f = open('my_file', 'w')
    f.write('Hello, ')
    f.write('World!')
    f.close()
```

Читання:

```
[5]: f = open('my_file', 'r')
     s=f.read(5)
     print(s, end="")
     s=f.read()
     print(s, end="")

Hello, World!
```

За замовчуванням метод read() читає дані послідовно по порядку, від початку і до кінця файлу. Для довільного доступу до файлу є функція seek:

```
[ ]: seek(offset[, whence])
```

offset - зміщення в байтах відносно початку файлу;

whence - за замовчуванням дорівнює нулю, вказує на те, що зміщення береться щодо початку файлу.

```
[7]: f = open(r'my_file', 'w')
     f.write('01234567890123456789')
     f.seek(5)
     f.write('Hello, World!')
     f.close()
     f = open(r'my_file')
     s=f.read()
     print(s, end="")

01234Hello, World!89
```

Функція tell () повертає поточну позицію файлу.

1.3 Порядкова робота з файлами

Зазвичай ми маємо справу з текстовими файлами. Прочитати один рядок:

```
file.readline()
```

Функція readline () без параметра читає весь рядок, наявність параметра вказує функції максимальне число символів рядка, яке буде прочитано.

Прочитати всі рядки і повернути список рядків:

```
file.readlines()
```

Записати рядки в файл:

```
file.writelines()
```

Приклад. Прочитати файл і записати його вміст в інший файл:

[18]:
```
f = open(r'my_file')
lines = f.readlines()
f.close()
lines[0] = "This is a my_file2 \n" # изменяем 1-ю строку
f = open(r'my_file2','w')
f.writelines(lines)
f.close()
f = open(r'my_file2','r')
s=f.readlines()
print(s)
f.close()
```

```
['This is a my_file2 \n']
```

1.4 Закриття файлу

Для закриття файлу є метод close ( ). Зазвичай файл закривається сам після того, як ви виходите з програми, але файли потрібно закривати вручну з кількох причин.

Пітон може буферізувати запис в файл даних, що може привести до несподіваних ефектів і виникнення помилок.

У операційної системи є обмеження на число одночасно відкритих файлів.

При доступі до файлу з різних місць одночасно і на читання, і на запис необхідно синхронізувати файлові операції. Буферизація запису може привести до того, що запис вже стався, а даних в файлі ще немає.

Для повної впевненості в закритті файлу можна використовувати блок try / finally:

```
[ ]: try:
        # Тут іде запис до файлу
     finally:
        file.close()
```

Можна також використовувати менеджер контексту, який в будь-якому випадку закриє файл:

```
with open("my_file") as somefile:
    do_something(somefile)
```

Якщо ви все ж не хочете закривати файл, то синхронізувати розрахований на багато користувачів доступ до файлу на читання / запис можна за допомогою функції flush (), яка актуалізує всі операції запису на диск. При цьому можливе блокування файлу на читання.

1.5 Ітерація

Ітерація по файлу є базовою операцією і має безліч варіантів. Використання функції read () для байтового читання:

```
[25]: f = open("my_file")
      while True:
          char = f.read(1)
          if not char: break
          print(char,end="")
      f.close()
```

```
01234Hello, World!89
```

Прогресивне читання текстових файлів і функція readline ():

```
[26]: f = open("my_file")
      while True:
          line = f.readline()
          if not line: break
          print(line,end="")
      f.close()
```

```
01234Hello, World!89
```

1.6 Бінарні файли

Стандартний модуль struct дозволяє перетворювати об'єкти в структури C у вигляді рядків в бінарному форматі і назад. Дані в рядку розташовуються відповідно до рядку формату. Ці можливості можуть бути використані для читання і збереження в двійковому форматі.

Функції цього модуля:

```
pack(format, value1, value2 ...)
```

Повертає рядок, що містить значення value1 ..., упаковані відповідно до формату. Кількість і тип аргументів повинні відповідати значенням, які вимагає рядок формату format.

```
unpack(format, string)
```

Розпаковує рядок string відповідно до формату format і повертає кортеж об'єктів.

```
calcsize(format)
```

Повертає розмір структури (тобто довжину рядка), що відповідає формату format.

Перед символом формату може йти число, що позначає кількість повторень. Наприклад, рядок формату '4h' повністю еквівалентна рядку 'hhhh'. Символи пропуску між символами формату ігноруються, проте символи пропуску між числом і символом формату не допускаються.

Число перед символом формату 's' інтерпретується як довжина рядка, а не число повторень. Тобто '10s' позначає рядок з 10 символів, в той час як '10c' - 10 раз по одному символу.

Можна змінити порядок проходження байтів вручну:

```
< - little-endian
> - big-endian
```

У наступному прикладі ми пакуємо в структуру два числа - ціле і float, рядок з п'яти символів, зберігаємо в бінарний файл, а потім витягуємо з файлу:

[19]:

```
from struct import *
out = open("123.bin", "wb")
f = "if5s"
data = pack(f, 24, 12.48,b"12345")
out.write(data)
out.close()
input = open("123.bin", "rb")
data = input.read()
input.close()
format = "if5s"                    # one integer
value,value2,value3 = unpack(format, data) # note the ',' in 'value,':
print(value)
print(value2)
print(value3)
print(calcsize(format))
```

```
24
12.479999542236328
b'12345'
13
```

2. Варіанти завдання

Для довідника реалізованого згідно власного варіанту у лабораторній роботі №4, створити програму з підтримкою функцій роботи з файлами, а саме:

- Збереження довідника до файлу.
- Читання довідника з файлу.
- Збереження змін у довіднику.

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 більший 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)

№	Завдання
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 взутя»** 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 товари 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

3. Приклад

Варіант - 16.

Лістинг

```
[*]: person = {}.fromkeys(['name', 'age', 'phone'])
person_list = list()
c=-1
def print_f(person_list):
    f = open('Person_List', 'w+')
    for i in range(len(person_list)):
        f.write("%10s" %person_list[i]['name'])
        f.write("%2d" %person_list[i]['age'])
        f.write("%15s" %person_list[i]['phone'])
        f.write("\n")
    f.close();
def read_f(person_list):
    f = open('Person_List', 'r+')
    i=0
    while True:
        s=f.read(10)
        if not s:
            break
        person_list+=[person.copy()]
        person_list[i]['name'] = s
        s=f.read(2)
        person_list[i]['age'] = int(s)
        s=f.read(15)
        person_list[i]['phone'] = s
        i+=1
        s=f.read(1)
    f.close()
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:  ")
    print_f(person_list)
def delete(person_list):
    print_p(person_list)
    buf=int(input("Введіть номер видаляємого запису:  "))
    del person_list[buf-1]
    print_f(person_list)
def print_p(person_list):
    for i in range(len(person_list)):
        print(i+1,"%10s" %person_list[i]['name'],"%2d" %person_list[i]['age'],"%15s" %person_list[i]['phone'])

read_f(person_list)
while c!=0:
    print("Меню")
    print("1. Додати запис")
    print("2. Переглянути усі записи")
    print("3. Підрахувати середній вік")
    print("4. Видалити запис")
    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))
    if c==4:
        delete(person_list)
```

Результат роботи

```
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
4. Видалити запис
0. завершити роботу
 2
1      Petro 54      0959559959
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
4. Видалити запис
0. завершити роботу
 1
Введіть ім'я:   Kurulo
Введіть вік:   35
Введіть телефон:   0958685478
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
4. Видалити запис
0. завершити роботу
 2
1      Petro 54      0959559959
2     Kurulo 35      0958685478
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
4. Видалити запис
0. завершити роботу
 4
1      Petro 54      0959559959
2     Kurulo 35      0958685478
Введіть номер видаляємого запису:   1
Меню
1. Додати запис
2. Переглянути усі записи
3. Підрахувати середній вік
4. Видалити запис
0. завершити роботу
 2
1     Kurulo 35      0958685478
```

4. Контрольні запитання

1. Опишіть існуючі режими доступу до файлу.
2. Вкажіть базові методи для роботи з файлами, та особливості їх застосування.
3. Наведіть основні функції для порядкової роботи з файлами.
4. Опишіть що таке бінарні файли, та основні особливості їх застосування.
5. Наведіть функції відкриття/закриття файлів.
6. Опишіть можливі ризики одночасного читання та запису до файлу.