

Лабораторна робота №3

Тема: Робота з файлами у мові Python

Мета: навчитися здійснювати операції читання та запису для файлів у мові Python

Теоретична частина

Розглянемо на прикладах основні операції з файлами у мові Python

Запис даних у файл

```
filename = "mydata.txt"

# Відкриття файлу
fd = open(filename, "w")

# Запис у файл
for i in range(10):
    A = i*18
    fd.write("%i\t%.1f\n" % (i, A))

# Закриття файлу
fd.close()
```

Читання з файлу

```
import csv
import sys

filename = "mydata.txt"

# Відкриття файлу
fd = open(filename, "r")

# Читання даних
reader = csv.reader(fd, delimiter="\t")

# Виведення змісту файлу
for row in reader:
    print(row)

# Закриття файлу
fd.close()
```

Копіювання файлу

```
import shutil
shutil.copyfile("C:\\mydoc.doc", "C:\\My Documents\\mydoc_2.doc")
```

Перейменування файлу

```
import os
os.rename("C:\\mydoc.doc\\testfile.txt", "/home/user/test.txt")
```

Видалення файлу

```
import os
os.remove(" C:\\mydoc.doc \\testfile.txt")
```

Читання потрібного рядка з текстового файлу

Щоб прочитати рядок під певним номером - можна скористатися як стандартним читанням файлу в список, так і використовувати модуль `linecache`:

```
line = linecache.getline("C:\\boot.ini", 2)
# or
line = open("C:\\boot.ini").readlines()[1]
```

Перебір файлів у каталозі

```
for filename in os.listdir("../plugins"):
    print(filename)
```

Перебір файлів у каталозі за маскою

```
import glob
for filename in glob.glob("../plugins\\*.zip"):
    print(filename)
```

Порівняння файлів

Порівнювати файли можна як за змістом, так і за їх властивостями, що значно швидше. Обидва варіанти можливі за допомогою `filecmp`

```
import filecmp
similar = filecmp.cmp('C:\\file1.txt', 'C:\\file2.txt')
print(similar)
```

Синтаксис функції `Open()` в Python.

```
my_file = open(ім'я_файлу [, режим_доступу][, буферизация])
```

У функції `Open()` багато параметрів, нам поки важливі 3 аргументи: перший, це ім'я файлу. Шлях до файлу може бути відносним або абсолютним. Другий аргумент, це режим, в якому ми будемо відкривати файл.

Режим	Призначення
'r'	відкриття на читання (є значенням за замовчуванням).
'w'	відкриття на запис, вміст файлу видаляється; якщо файлу не існує, створюється новий.
'x'	ексклюзивне створення (збуджується виключення <code>FileExistsError</code> , якщо файл вже існує).
'a'	відкриття на дозапис, інформація додається в кінець файлу.
'b'	відкриття в двійковому режимі.
't'	відкриття в текстовому режимі (є значенням за замовчуванням).
'+'	відкриття на читання і запис

Приклади використання декількох режимів одночасно

Режим	Призначення
r+b	Відкрити бінарний файл на читання і запис. Показчик стоїть на початку файлу.
w+b	Відкрити бінарний файл на читання і запис, очистити до 0 байт. Показчик стоїть на початку файлу.
wb	Відкриває файл для запису в двійковому форматі. Показчик стоїть на початку файлу. Створює файл з ім'ям ім'я_файлу, якщо такого не існує.
a+	Відкриває файл для додавання і читання. Показчик стоїть в кінці файлу. Створює файл з ім'ям_файлу, якщо такого не існує.

Завдання:

Створіть файли, у яких будуть міститися рядки з іменами студентів та їх середніми балами. Кожен файл буде відповідати окремій групі. Реалізуйте читання файлів, запис та дозапис у файли, пошук файлів у каталозі та пошук даних у файлі. Також реалізуйте сортування даних у файлі за середнім балом.

Контрольні питання:

1. Як здійснюється запис даних у файл у мові Python?
2. Як здійснюється читання даних з файлу у мові Python?
3. Як здійснюється копіювання файлу?
4. Який синтаксис функції `Open()` у мові Python?
5. Які бувають режими роботи з файлами?