Лабораторная работа №3 Программирование в Питоне – работа с файлами

1. Цель и порядок работы

Цель работы – изучить возможности работы с файлами в Питоне.

Порядок выполнения работы:

Самостоятельно проработать теоретические основы по языку программирования Питон, выполняя указанные в тексте примеры.

Выполнить индивидуальное задание.

2. Краткая теория

2.1 Общие замечания

2.1.1 Вопрос именования адресов

Пути задаются либо с удвоением симола бэкслэш (например 'D:\\txt.txt'), либо с указанием r ('raw') перед кавычками r'D:\\txt.txt'. Это делается для того, чтобы избежать случаев, когда символы \n или \t принимаются за спецсимволы.

2.1.2 Порядок работы с файлами

Общий порядок работы с файлами выглядит следующим образом - с произвольной переменной с помощью функции open ('путь к файлу', 'режим работы с файлом') связывается определенный файл, после чего все операции с данным файлом выполняются с помощью этой переменной. В конце операций файл отключается от переменной с помощью функции имя_переменной.close().

Для следующего примера нам понадобится текстовый файл. Создадим его и назовем 1.txt

(используйте любой текстовый редактор) со следующим текстом:

1, 2, 3
one two three
this is end of the file

Сохраните файл на диск C. Тогда путь к нему будет выглядеть как C:\ 1.txt.

Для начала посмотрим как файл подключается к произвольной переменной. В командной строке наберите:

```
>>>  a =  open(' C:\\ 1.txt',  'rb')
```

Этим вы свяжете файл 1.txt (файл-как-объект) с переменной a. Эта

переменная является "отражением" файла в пространстве Питона. Проверим это (вывод вашей среды может отличаться):

```
>>> a
<open file ' C:\\ 1.txt', mode 'rb' at 0xb780b430>
```

Это указывает на то, что переменная a связана с файлом 1.txt. Причем файл 1.txt открыт в режиме чтения.

Тип переменной a можно будет сделать при помощи функции type()

```
>>> type(a)
<type 'file'>
```

2.2 Режимы работы с файлом

Режим работы задается с помощью текстовой строки, которая может принимать следующие значения:

• r - (read) Файл открывается для чтения. Файл должен уже существовать.

• w - (write) Файл открывается для записи. Если файл не существует - он будет создан. При этом он будет перезаписан, если там что-то уже находилось.

• a - (append) Если файл не существует - он будет создан. Файл открывается для записи, при этом новая запись будет добавлена к содержимому файла.

• r+ - (read+) Позволяет сразу чтение и запись. Если файл не существует - он будет создан.

При этом он будет перезаписан, если там что-то уже находилось.

• a+ - (append+) Позволяет сразу чтение и запись. Если файл не существует - он будет создан.

При этом новая запись будет добавлена к содержимому файла.

Кроме того, к режиму дописывается постфикс обозначающий - как именно следует читать файл. Если там указано b - (binary) Питон будет работать с файлом, как с двоичным файлом любого формата, в противном случае - он будет открыт как текстовый файл.

Обратите внимание на то, что мы открыли файл, как двоичный (режим 'rb') - это основная практика при работе с платформой Windows.

Также обратите внимание на то, что режимы и путь к файлу - это обычные текстовые строки, которые могут быть заданы как явно, так и неявно

(в виде переменной, которая формируется в зависимости от результатов вычислений).

2.3 Методы работы с файлом

Все методы работы с файлом-объектом a можно посмотреть с помощью директивы dir():

```
>>> dir(a)
['__class__', '__delattr__', '__doc__', '__enter__', '__exit__', '__format__',
'__getattribute__', '__hash__', '__init__', '__iter__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__',
'__subclasshook__', 'close', 'closed', 'encoding', 'errors', 'fileno', 'flush',
'isatty', 'mode', 'name', 'newlines', 'next', 'read', 'readinto', 'readline',
'readlines', 'seek', 'softspace', 'tell', 'truncate', 'write', 'writelines', 'xreadlines']
```

С частью из этих функций мы познакомимся ниже.

2.4 Чтение файла: read()

Функция read() считывает файл целиком, одной длинной строкой с разделителями \n в тех местах, где стоят абзацы. Если указывается количество байтов (например read(15)) - то считывается именно это количество, если нет - то считывается весь файл целиком.

Прочитаем файл:

```
>>>  a.read()
'1,  2,  3\none  two  three\nthis  is end of  the  file\n'
```

Перемещение "указателя", с которого начинается чтение, вдоль байтовой структуры файла осуществляется с помощью двух функций - tell() и seek(). Первая из них возвращает число байтов от начала файла до текущего положения, вторая - перемещает указатель на нужное количество байтов. Функция seek() задается в форме seek('на сколько байтов переместиться', 'точка отсчета'), при этом точка отсчета может принимать следующие значения:

- 0 - начало файла
- 1 - текущее положение указателя
- 2 - конец файла

Переместим указатель к началу файла:

```
>>> a.seek(0,0)
>>> a.tell()
0L
```

Теперь считаем из файла четыре байта:

```
>>> a.read(4)
'1, 2'
>>> a.tell()
4L
```

Мы считали четыре байта и указатель сместился на четыре байта к концу файла. Обратите внимание, что в данной кодировке один байт равен одному символу. Это далеко не всегда так, (например, это неверно если файл записан в кодировке Unicode), но в данном случае позволяет легко ориентироваться в работе файловых функций).

```
>>> a.seek(2,1)
>>> a.tell()
6L
>>> s = a.read(7)
>>> s
'3\none t'
```

И так далее.

```
>>> a.seek(-12,2)
>>> a.read()
'of the file\n'
>>> a.seek(14,0)
>>> a.tell()
14L
>>> a.read()
'o three\nthis is end of the file\n'
>>> a.tell()
46L
```

После работы с файлом, его нужно закрыть "отвязав" от переменной a:

```
>>> a.close()
>>> a
<closed file '1.txt', mode 'rb' at 0xb780b430>
```

2.5 Чтение файла: readlines()

Функция read(), является базовой, но не всегда удобной для работы с файлом функцией. Наиболее удобная в Питоне функция называется readlines() - она считывает весь файл сразу в виде списка из строк:

```
>>> a = open('1.txt', 'rb')
>>> a.readlines()
['1, 2, 3\n', 'one two three\n', 'this is end of the file\n']
```

Здесь мы считываем сразу весь файл и дальше можем работать со списком всеми традиционными средствами питона. Например, создадим скрипт, распечатывающий первые буквы каждой строки файла:

```
a = open('1.txt','rb')
s = a.readlines()
for i in s:
        print i
```

Результат работы:

```
1
o
t
```

Обратите внимание на то, что строки записываются в список вместе со знаком перевода строки \n. Если вам требуется строка "сама по себе" ее можно, обрезать - например с помощью среза:

```
a = open('1.txt','rb')
s = a.readlines()

print 'all lines:' print
for i in s:
        print i,

print 'all lines in one line:' print
for i in s:
        print i[:-1],
```

Результат работы:

```
all lines:

1, 2, 3
```

one two three
this is end of the file all lines in one line:

1, 2, 3 one two three this is end of the file

2.6 Запись в файл: write()

Запись в файл производится с помощью функции write(). Вызов функции вида имя_файла.write('строка') записывает в файл (дописывает или записывает - определяется режимом открытия файла) строку 'строка'. Режим открытия файла определяет - как именно будет туда записана строка.

Обратите внимание - при записи в файл нужно самому заботиться о сохранении знаков перевода строки.

Также обратите внимание, что чтобы записать в файл что-то "не-строчное" его нужно преобразовать в строку с помощью функции str()

Введем скрипт следующего примера:

```
b = open('2.txt','wb')
for i in range(29):
        b.write(str(i))
```

Результатом работы будет файл 2.txt со следующим содержанием:

0123456789101112131415161718192021222324252627 28

Если мы изменим его:

```
b = open('2.txt','wb')
for i in range(29):
        b.write(str(i) + ' ')
```

То цифры в файле будут разделены пробелами:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

А если мы позаботимся о абзацах:

```
b = open('2.txt','wb')
for i in range(29):
        b.write(str(i) + '\n')
```

Получим:

```
0
1
2
3
4
5

...

23
24
25
26
27
28
```

2.7 Полный цикл работы чтения-записи файла

Пример ниже иллюстрирует работу с чтением и записью файла. Он меняет местами первую строку файла с последней. В данном случае мы читаем файл 1.txt, считываем из него информацию, меняем местами первую и последнюю строки файла и записываем получившийся список в файл 3.txt.

```
a =  open('1.txt','rb')
old  =  a.readlines()
a.close()

first  =  old[0]
last  =  old[-1]
new  =  old
new[0]  =  last
new[-1]  =  first

b  =  open('3.txt','wb')
for  i in  new:
        b.write(i)
```

Файл 1.txt (исходный файл):

```
1,  2,  3
one two  three
```

this is end of the file

Файл 3.txt (файл с результатом работы программы):

this is end of the file one two three
1, 2, 3

3. Вопросы для самопроверки

• Где Питон будет искать файл, если указано только имя файла?

• Каких правил следует придерживаться, задавая путь к файлу в системе MS Windows?

• Опишите последовательность действий с файлом при его открытии. В каком режиме возможно открытие файла? Как указывается путь к нему, какие есть варианты указания такого пути? Приведите пример.

• Перечислите и опишите режимы работы с файлом и их отличия друг от друга.

• Опишите последовательность действий с файлом при его закрытии. Как указывается путь к нему, какие есть варианты указания такого пути? Приведите пример.

• Как в общем случае организуется последовательность чтения-записи из-в файл? Приведите пример.

• Опишите работу команды read() Приведите пример ее использования.

• Опишите работу команды seek() Приведите пример ее использования.

• Опишите работу команды tell() Приведите пример ее использования.

• Опишите работу команды readline() Приведите пример ее использования.

• Опишите работу команды readlines() Приведите пример ее использования.

• Опишите работу команды write() Приведите пример ее использования.

• Какие преимущества предоставляет команда readlines() по сравнению с командой read()?

4. Варианты заданий

Вариант определяется по последней цифре в списке студентов группы.

Вариант 1.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- фамилия и инициалы;
- номер группы;
- успеваемость (пять оценок).

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на дисплей фамилий и номеров групп для всех студентов, если средний балл студента больше 4.0, если таких студентов нет, вывести соответствующее сообщение.

Вариант 2.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- фамилия и инициалы;
- номер группы;
- успеваемость (пять оценок).

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2, если таких студентов нет, вывести соответствующее сообщение.

Вариант 3.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- название пункта назначения рейса;
- номер рейса;
- тип самолета.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на экран информации о рейсе, номер которого введен с клавиатуры, если таких рейсов нет, вывести соответствующее сообщение.

Вариант 4.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- название пункта назначения рейса;

- номер рейса;
- тип самолета.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры, если таких рейсов нет, вывести соответствующее сообщение.

Вариант 5.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- фамилия и инициалы работника;
- название занимаемой должности;
- зарплату;
- год поступления на работу.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры, если таких работников нет, вывести соответствующее сообщение.

Вариант 6.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- название пункта назначения;
- номер поезда;
- время отправления.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени, если таких поездов нет, вывести соответствующее сообщение.

Вариант 7.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- название пункта назначения;
- номер поезда;
- время отправления.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на экран информации о пункте назначения, в который отправляется поезд, номер которого введен с клавиатуры, если таких поездов нет, вывести соответствующее сообщение.

Вариант 8.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на экран информации о маршруте, номер которого введен с клавиатуры, если таких маршрутов нет, вывести соответствующее сообщение.

Вариант 9.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- фамилия и имя;
- номер телефона;
- дата рождения.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на экран информации о человеке, номер телефона которого введен с клавиатуры, если таких людей нет, вывести соответствующее сообщение.

Вариант 0.

Текстовый файл может содержать данные следующего типа записанные через соответствующий разделить:

- расчетный счет плательщика;
- расчетный счет получателя;
- перечисляемая сумма в руб.

Написать программу, выполняющую следующие действия:

1. ввод с клавиатуры данных и запись их в файл;
2. вывод на экран информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры, если таких счетов нет, вывести соответствующее сообщение.

5. Рекомендуемая литература

1. Васильев А. Н. Python на примерах. Практический курс ·по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.
2. http://python.org Официальный сайт языка Питон.
3. http://www.python.ru Русскоязычная документация языка Питон.

4. http://ru.wikipedia.org/wiki/Python Статья, посвященная Питону в Википедии.