

## **Лабораторная работа №2. Часть 2**

### **Работа с файлами. Чтение/запись файлов**

#### **Теоретические сведения**

#### **Открытие и закрытие файла в Python.**

Перед тем как прочесть и записать что-либо в файл, его следует открыть. Чтобы открыть файл в Python используется встроенная функция `open()`. При вызове, эта функция создает объект типа файл, с которым в дальнейшем можно работать.

Синтаксис функции `open()` в Python.

```
my_file = open(имя_файла [, режим_доступа] [, буферизация])
```

`имя_файла`: строка, содержащая имя файла с расширением. Например, "my\_file.txt".

`режим_доступа`: строка, которой мы указываем для чего открывается файл: для чтения, записи, добавления информации, и т.д. Например, "w". По умолчанию файл открывается для чтения - "r".

`буферизация`: Целое число. Если значение аргумента указано 0 - файл открывается без буферизации, 1 с построчной буферизацией, больше одного процесс буферизации выполняется с указанным размером буфера. Отрицательное число - размер буфера будет равен системному.

#### **Список режимов доступа к файлу в Python.**

- `r`      Открывает файл только для чтения. Указатель стоит в начале файла.
- `rb`     Открывает файл для чтения в двоичном формате. Указатель стоит в начале файла.
- `r+`     Открывает файл для чтения и записи. Указатель стоит в начале файла.
- `rb+`    Открывает файл для чтения и записи в двоичном формате. Указатель стоит в начале файла.
- `w`      Открывает файл только для записи. Указатель стоит в начале файла. Создает файл с именем `имя_файла`, если такового не существует.
- `wb`     Открывает файл для записи в двоичном формате. Указатель стоит в начале файла. Создает файл с именем `имя_файла`, если такового не существует.
- `w+`     Открывает файл для чтения и записи. Указатель стоит в начале файла. Создает файл с именем `имя_файла`, если такового не существует.

- wb+ Открывает файл для чтения и записи в двоичном формате. Указатель стоит в начале файла. Создает файл с именем имя\_файла, если такового не существует.
- a Открывает файл для добавления информации в файл. Указатель стоит в конце файла. Создает файл с именем имя\_файла, если такового не существует.
- ab Открывает файл для добавления в двоичном формате. Указатель стоит в конце файла. Создает файл с именем имя\_файла, если такового не существует.
- a+ Открывает файл для добавления и чтения. Указатель стоит в конце файла. Создает файл с именем имя\_файла, если такового не существует.
- ab+ Открывает файл для добавления и чтения в двоичном формате. Указатель стоит в конце файла. Создает файл с именем имя\_файла, если такового не существует.

### **Атрибуты файлового объекта в Python.**

Как только файл был открыт и у вас появился файловый объект, вы можете получить следующую информацию о нем:

file.closed Возвращает True если файл был закрыт.

file.mode Возвращает режим доступа, с которым был открыт файл.

file.name Возвращает имя файла.

file.softspace Возвращает False если при выводе содержимого файла следует отдельно добавлять пробел.

Например:

```
my_file = open("some.txt", "w")
print("Имя файла: ", my_file.name)
print("Файл закрыт: ", my_file.closed)
print("В каком режиме файл открыт: ", my_file.mode)
print("Пробелы: ", my_file.softspace)
Закрытие файла в Python. Метод close().
```

Метод файлового объекта close() автоматически закрывает файл, при этом теряется любая несохраненная информация. Работать с файлом (читать, записывать) после этого нельзя.

Python автоматически закрывает файл если файловый объект к которому он привязан присваивается другому файлу. Однако, хорошей практикой будет вручную закрывать файл командой close().

```
my_file = open("some.txt")
print("Имя файла: ", my_file.name)
print("Файл закрыт: ", my_file.closed)
```

```
my_file.close()
print("А теперь закрыт: ", my_file.closed)
Чтение и запись файлов в Python
```

Для файлового объекта доступен целый набор методов, чтобы облегчить нашу работу с файлом.

### **Запись в файл в Python. Метод write().**

Метод write() записывает любую строку в открытый файл. Важно помнить, что строки в Python могут содержать двоичные данные, а не только текст.

Метод write() не добавляет символ переноса строки ('\n') в конец файла.

Синтаксис метода write().  
`my_file.write(string);`

Например:

```
my_file = open("some.txt", "w")
my_file.write("Мне нравится Python!\nЭто классный язык!")
my_file.close()
```

Вышеприведенный код создаст файл some.txt и запишет в него указанную строку.

### **Чтение из файла в Python. Метод read().**

Метод read() читает строку из открытого файла.

Синтаксис метода read().

```
my_file.read([count])
```

Необязательный параметр count - это количество байт, которые следует прочитать из открытого файла. Этот метод читает информацию с начала файла и, если параметр count не указан, до конца файла.

Например, прочтем созданный нами файл some.txt:

```
my_file = open("some.txt")
my_string = my_file.read()
print("Было прочитано:")
print(my_string)
my_file.close()
```

работа с файлами в python, python fileio

Как узнать позицию указателя в файле в Python.

После того как вы вызвали метод `read()` на файловом объекте, если вы повторно вызовете `read()`, то увидите лишь пустую строку. Это происходит потому, что после первого прочтения указатель находится в конце файла. Для того чтобы узнать позицию указателя можно использовать метод `tell()`.

Например:

```
my_file = open("some.txt")
my_file.read(10)
print ("Я на позиции:", my_file.tell())
my_file.close()
```

Говоря проще, метод `tell()` сообщает в скольких байтах от начала файла мы сейчас находимся.

Чтобы перейти на нужную нам позицию, следует использовать другой метод - `seek()`.

Синтаксис метода `seek()`.

```
my_file.seek(offset, [from])
```

Аргумент `offset` указывает на сколько байт перейти. опциональный аргумент `from` означает позицию, с которой начинается движение. 0 - означает начало файла, 1 - нынешняя позиция, 2 - конец файла.

Например:

```
my_file = open("some.txt", "r")
print(my_file.read(10))
print("Мы находимся на позиции: ", my_file.tell())
# Возвращаемся в начало
my_file.seek(0)
print(my_file.read(10))
my_file.close()
Добавление в файл. Метод write()
```

Если вы хотите не перезаписать файл полностью (что делает метод `write` в случае открытия файла в режиме `'w'`), а только добавить какой-либо текст, то файл следует открывать в режиме `'a'` - appending. После чего использовать все тот же метод `write`.

Например:

```
# Удалит существующую информацию в some.txt и запишет "Hello".
my_file = open("some.txt", 'w')
my_file.write("Hello")
my_file.close()
# Оставит существующую информацию в some.txt и добавит "Hello".
my_file = open("some.txt", 'a')
my_file.write("Hello")
my_file.close()
Расширенная работа с файлами в Python.
```

## **Задание**

### **Варианты**

#### **№1**

В магазине формируется список лиц, записавшихся на покупку товара повышенного спроса. Каждый элемент этого списка содержит: порядковый номер, Ф.И.О., домашний адрес покупателя и дату постановки на учет. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№2**

Список товаров, имеющихся на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№3**

Для получения места в общежитии формируется список студентов, который включает Ф.И.О. студента, группу, средний балл, доход на члена семьи. Общежитие в первую очередь предоставляется тем, у кого доход на члена семьи меньше двух минимальных зарплат, затем остальным в порядке уменьшения среднего балла. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№4**

В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№5**

На междугородной АТС информация о разговорах содержит дату разговора, код и название города, время разговора, тариф, номер телефона в этом

городе и номер телефона абонента. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№6**

Информация о сотрудниках фирмы включает: Ф.И.О., табельный номер, количество проработанных часов за месяц, почасовой тариф. Рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№7**

Информация об участниках спортивных соревнований содержит: наименование страны, название команды, Ф.И.О. игрока, игровой номер, возраст, рост, вес. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№8**

Для книг, хранящихся в библиотеке, задаются: регистрационный номер книги, автор, название, год издания, издательство, количество страниц. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№9**

Различные цехи завода выпускают продукцию нескольких наименований. Сведения о выпущенной продукции включают: наименование, количество, номер цеха. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№10**

Информация о сотрудниках предприятия содержит: Ф.И.О., номер отдела, должность, дату начала работы. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№11**

Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит: Ф.И.О., адрес, оценки. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№12**

В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны: номер рейса, тип самолета, пункт назначения, время вылета. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№13**

У администратора железнодорожных касс хранится информация о свободных местах в поездах дальнего следования на ближайшую неделю в следующем виде: дата выезда, пункт назначения, время отправления, число

свободных мест. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№14**

Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит: Ф.И.О. абитуриента, оценки. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№15**

В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждая квитанция содержит следующую информацию: наименование группы изделий (телевизор, радиоприемник и т.п.), марку изделия, дату приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№16**

Разработать программу формирования ведомости об успеваемости студентов. Каждая запись этой ведомости должна содержать: номер группы, Ф.И.О. студента, оценки за последнюю сессию. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№17**

В исполкоме формируется список учета нуждающихся в улучшении жилищных условий. Каждый элемент этого списка содержит: порядковый номер, Ф.И.О., величину жилплощади на одного члена семьи и дату постановки на учет. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№18**

Имеется список женихов и список невест. Каждая запись списка содержит пол, имя, возраст, рост, вес, а также требования к партнеру: наименьший и наибольший возраст, наименьший и наибольший вес, наименьший и наибольший рост. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№19**

В библиотеке имеется список книг. Каждая запись этого списка содержит: фамилии авторов, название книги, год издания. Написать программу со следующими функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.

#### **№20**

В магазине имеется список поступивших в продажу автомобилей. Каждая запись этого списка содержит: марку автомобиля, стоимость, расход топлива на 100 км, надежность (число лет безотказной работы), комфортность (отличная, хорошая, удовлетворительная). Написать программу со следующими

функциями: добавление/удаление записей в файл, вывод списка на консоль, главное меню.