

ЛАБОРАТОРНАЯ РАБОТА 6

Содержание

Чтение текстовых файлов	2
Запись текстовых файлов	3
Осуществление действий с файлами	4
Создание файлов	4
Создание каталога	5
Копирование файлов	5
Перемещение файлов	5
Переименование файлов	5
Удаление файлов	6
Получение списка файлов	6
Проверка существования файла	6
Требования к программам	6
Базовый шаблон для всех лабораторных	7
Пример программы	7
Индивидуальные задания	8
Источники текстов	13
Полезные ссылки	13

Чтение текстовых файлов

Пример содержимого файла "input.txt" для последующих заданий:

From [The Python Tutorial](#):

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

Задание № 0.1. Определить результат выполнения следующего кода:

- (a)

```
infile = open("input.txt")
for line in infile:
    print(line, end="")
infile.close()
```
- (b)

```
with open("input.txt") as infile:
    for line in infile:
        print(line, end="")
```
- (c)

```
with open("input.txt") as infile:
    first_line = infile.readline()
    print(first_line[:-1])

    print("."*70)

    read_data = infile.read()
    for line in read_data:
        print(line, end="")
```
- (d)

```
with open("input.txt") as infile:
    list_of_lines = infile.readlines()

    for line in list_of_lines:
        print(line, end="")
```

Запись текстовых файлов

! Внимание: функция `open()` в режиме записи `"w"` всегда открывает пустой файл. Если файла с таким именем не существовало, он будет создан; если он существовал, его содержимое будет стерто без предупреждения.

Задание № 0.2. Определить результат выполнения следующего кода:

- (a)

```
outfile = open("output.txt", "w")
outfile.write("Hello")
outfile.close()
```
- (b)

```
with open("output.txt", "w") as outfile:
    outfile.write("Hello")
```
- (c)

```
courses = [
    "Algebra",
    "Data Analysis",
    "Network Analysis",
    "English Language",
    "Demography",
    "Discrete Mathematics",
    "Big Data in the Social Sciences",
    "Programming in Python for Data Analysis"
]

with open("output.txt", "w") as outfile:
    for course in courses:
        outfile.write(course + "\n")
```
- (d)

```
courses = ["Social History", "Sociology of Consumption"]

with open("output.txt", "a") as outfile:
    write_data = "\n".join(courses)
    outfile.write(write_data)
```
- (e)

```
with open("multiplication_table.txt", "w") as outfile:
    for a in range(1, 10):
        for b in range(1, 10):
            print(f"{a*b:2d}", end="  ", file=outfile)
        print(file=outfile)
```

Осуществление действий с файлами

! Внимание: перед выполнением любого кода, в частности кода, работающего с файлами, необходимо убедиться, что в понимании, что он делает. Выполнение некорректных или вредоносных команд может привести к потере данных или к сбою операционной системы.

Создание файлов

Задание № 0.3. Создать текстовые файлы в соответствии с описанием.

- (a) 10 пустых файлов с именами, заданными форматом: "lab_n.txt", где n — число от 0 до 9.
- (b) 9 файлов с таблицами степеней числа n. Имена файлов задать в формате "powers_of_n.txt". Число n принимает значение от 2 до 10. Степени брать от 1 до 10.
- (c) Файлы с именами, заданными списком `fnames`, содержащие строку "This file will contain text from Name", где Name — имя файла без расширения.

```
fnames = ["whatsnew.txt", "library.txt", "tutorial.txt"]
```

```
1 '''Создание файлов по описаниям задания 0.4.
2
3 Каждый подпункт задания реализован в виде функции.
4 '''
5 def task_0_4_a():
6     for n in range(10):
7         fname = f"lab_{n}.txt"
8         outfile = open(fname, "w")
9         outfile.close()
10
11 def task_0_4_b():
12     for n in range(2, 11):
13         fname = f"powers_of_{n}.txt"
14         with open(fname, "w") as outfile:
15             for i in range(1, 11):
16                 print(f"{i:2d} {n**i:10d}", file=outfile)
17
18 def task_0_4_c():
19     fnames = ["whatsnew.txt", "library.txt", "tutorial.txt"]
20     for fname in fnames:
21         with open(fname, "w") as outfile:
22             outstr = f"This file will contain text from {fname[:-4]}\n"
23             outfile.write(outstr)
24
25 task_0_4_a()
26 task_0_4_b()
27 task_0_4_c()
```

Создание каталога

Задание № 0.4. Определить результат выполнения следующего кода:

(a) `import os`

```
dir_name = "powers"  
os.mkdir(dir_name)
```

Копирование файлов

Задание № 0.5. Определить результат выполнения следующего кода:

(a) `import shutil`

```
for n in range(2, 11):  
    fname      = f"powers_of_{n}.txt"  
    copyfname = f"{n}.txt"  
    shutil.copy(fname, copyfname)
```

Перемещение файлов

Задание № 0.6. Определить результат выполнения следующего кода:

(a) `import os`

```
destination = "powers/"  
  
for n in range(2, 11):  
    fname = f"{n}.txt"  
    os.replace(fname, destination + fname)
```

Переименование файлов

Задание № 0.7. Определить результат выполнения следующего кода:

(a) `import os`

```
for n in range(10):  
    oldfname = f"lab_{n}.txt"  
    newfname = f"labwork_{n}.txt"  
    os.rename(oldfname, newfname)
```

Удаление файлов

Задание № 0.8. Определить результат выполнения следующего кода:

```
(a) import os

for n in range(0, 10, 2):
    fname = f"labwork_{n}.txt"
    os.remove(fname)
```

Получение списка файлов

Задание № 0.9. Определить результат выполнения следующего кода:

```
(a) import os
    list_of_files = os.listdir()
    print(list_of_files)

(b) import os
    list_of_files = os.listdir("powers")
    print(list_of_files)
```

Проверка существования файла

Задание № 0.10. Определить результат выполнения следующего кода:

```
(a) import os

fname = "input.txt"

if os.path.exists(fname):
    print("yes")
else:
    print("no")
```

Требования к программам

1. Файл с исходным кодом называть по шаблону: «Фамилия_номер_задания» английским алфавитом (пример: `Ivanov_6_42.py`).
2. Файл с исходным кодом должен начинаться с многострочного описательного комментария (см. базовый шаблон ниже).
3. Все имена выбирать разумными; оформлять понятный ввод данных / вывод результата.

Базовый шаблон для всех лабораторных

```
1 '''Фамилия Имя. Номер задания
2
3 Краткая формулировка задания
4 '''
5
6 # Код программы
```

Пример программы

```
1 '''Фамилия Имя. Задание № 6.0
2
3 Записать в файл "output.txt" список слов файла "input.txt"
4 вместе с количеством вхождений.
5
6 Словом считать любую последовательность символов,
7 разделенных пробельными символами и/или знаками препинания (,.)
8
9 Алгоритм:
10 1. Определить пустой словарь
11
12 2. Для каждой строки файла:
13 3.     разбить строку на список строк по пробельным символам;
14
15 4.     для каждой строки списка:
16 5.         получить слово, удалив справа знаки препинания (,.);
17
18 6.         для каждого слова:
19 7.             если этого слова еще нет в словаре:
20 8.                 добавить его со значением 1;
21 9.             если есть:
22 10.                увеличить значение для этого слова на 1.
23
24 11. Для каждого элемента словаря:
25 12.     записать в выходной файл слово и количество вхождений.
26 '''
27
28 words = dict()
29
30 with open("input.txt") as infile:
31     for line in infile:
32         list_of_strings = line.split()
33
34         for string in list_of_strings:
35             word = string.rstrip(",.")
36
37             if word not in words:
38                 words[word] = 1
39             else:
40                 words[word] = words[word] + 1
41
42 with open("output.txt", "w") as outfile:
43     for word in words:
44         print(f"{word:20s} {words[word]:5d}", file=outfile)
```

Индивидуальные задания

Задание № 6.1. Дан текстовый файл. Вывести количество пустых и количество непустых строк в этом файле.

Пример:

Содержимое входного файла:

```
Это первая непустая строка

Это вторая непустая строка
Это третья непустая строка


Это четвертая непустая строка
```

Вывод программы:

```
Количество непустых строк: 4
Количество пустых строк: 2
```

Задание № 6.2. Дан текстовый файл и целое положительное число `count`. Вывести первые `count` строк.

Пример:

`count = 6`

Содержимое входного файла:

```
Первая строка.
Вторая строка.
Третья строка.
Четвертая строка.
Пятая строка.
Шестая строка.
Седьмая строка.
Восьмая строка.
```

Вывод программы:

```
Первая строка.
Вторая строка.
Третья строка.
Четвертая строка.
Пятая строка.
Шестая строка.
```

Задание № 6.3. Дан текстовый файл, в строках которого записаны вещественные числа. Вывести среднее арифметическое данных чисел.

Пример:

Содержимое входного файла:

```
0.270018
0.697319
0.694041
0.391147
0.496460
0.498781
0.500896
0.500719
0.499529
0.499796
```

Вывод программы:

```
Среднее значение: 0.504871
```


Задание № 6.4. Дано нечетное число `width` и символ `symbol`. Создать новый текстовый файл с именем `"diamond_w.txt"`, где `w` — значение переменной `width`. В файле «нарисовать» ромб из данного символа заданной ширины.

Пример:

```
width = 11
symbol = "*"
```

Содержимое выходного файла
`"diamond_11.txt"`:

```
      *
     ***
    *****
   *********
  ***********
 *****
  ***********
   *********
    *****
     ***
      *
```

Задание № 6.5. Дан текстовый файл, в строках которого записаны числа, разделенные пробелами. Вывести номера строк файла и количество чисел в данной строке.

Пример:

Содержимое входного файла:

```
1 2 3 4 5
5 7 9
10 9 8 7
1 2 3
0 0 0
```

Вывод программы:

```
Строка 1: 5
Строка 2: 3
Строка 3: 4
Строка 4: 3
Строка 5: 3
```

Задание № 6.6. Дан текстовый файл со словами, записанными по одному в каждой строке. Создать новый текстовый файл со словами из входного файла, записанными в алфавитном порядке.

Пример:

Содержимое входного файла:

```
return
import
with
break
if
in
for
else
```

Содержимое выходного файла:

```
break
else
for
if
import
in
return
with
```

Задание № 6.7. Дан файл с текстом. Создать новый файл, записав в него каждую строку в перевернутом виде.

Пример:

Содержимое входного файла:

```
а роза упала на лапу Азора
Лёша на полке клопа нашёл
нам сила - талисман
я и мумия
лезу в узел
```

Содержимое выходного файла:

```
арозА упал ан алапу азор а
лёшан аполк еклоп ан ашёЛ
намсилат - алис ман
яимум и я
лезу в узел
```

Задание № 6.8. Создать файл с именем "squares_and_cubes_up_to_N.txt", где N — значение заданного целого положительного числа `upper_number`. В файл записать таблицу квадратов и кубов чисел от 1 до `upper_number`.

Пример:

`upper_number = 10`

Содержимое выходного файла

"squares_and_cubes_up_to_10.txt":

1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Задание № 6.9. Дан текстовый файл и две строки `oldstr` и `newstr`. Создать новый файл, в который переписать содержимое входного файла, заменяя все вхождения строки `oldstr` на `newstr`.

Пример:

`oldstr = " May "`

`newstr = ".05."`

Содержимое входного файла:

```
01 May 2023
02 May 2023
03 May 2023
04 May 2023
05 May 2023
06 May 2023
```

Содержимое выходного файла:

```
01.05.2023
02.05.2023
03.05.2023
04.05.2023
05.05.2023
06.05.2023
```

Задание № 6.10. Дан текстовый файл с датами в формате: dd.mm.yyyy. Создать файл с теми же датами в формате: dd Mon yyyy, где Mon — первые три буквы названия соответствующего месяца.

Пример:

Содержимое входного файла:

```
01.05.2023
02.06.2023
03.07.2023
04.08.2023
05.09.2023
06.10.2023
```

Содержимое выходного файла:

```
01 May 2023
02 Jun 2023
03 Jul 2023
04 Aug 2023
05 Sep 2023
06 Oct 2023
```

Задание № 6.11. Дан текстовый файл со списком фамилий и инициалов. Каждый элемент списка записан в отдельной строке. Фамилии и инициалы разделены пробелом, инициалы написаны без пробелов. Получить новый файл, в котором в строках инициалы поставлены перед фамилией.

Пример:

Содержимое входного файла:

```
Пушкин А.С.
Лермонтов М.Ю.
Есенин С.А.
Ахматова А.А.
Маяковский В.В.
```

Содержимое выходного файла:

```
А.С. Пушкин
М.Ю. Лермонтов
С.А. Есенин
А.А. Ахматова
В.В. Маяковский
```

Задание № 6.12. Дан текстовый файл. Создать новый текстовый файл с автоматически пронумерованными строками.

Пример:

Содержимое входного файла:

```
with open("multiplication_table.txt", "w") as outfile:
    for a in range(1, 10):
        for b in range(1, 10):
            print(f"{a*b:2d}", end="  ", file=outfile)
        print(file=outfile)
```

Содержимое выходного файла:

```
1: with open("multiplication_table.txt", "w") as outfile:
2:     for a in range(1, 10):
3:         for b in range(1, 10):
4:             print(f"{a*b:2d}", end="  ", file=outfile)
5:             print(file=outfile)
```

Задание № 6.13. Дан файл с текстом. Создать новый файл с тем же текстом, предварительно каждую непустую строку указанием длины строки.

Пример:

Содержимое входного файла:

```
If you do much work on computers, eventually you find
that there's some task you'd like to automate.
For example, you may wish to perform a search-and-replace
over a large number of text files, or rename and
rearrange a bunch of photo files in a complicated way.

Perhaps you'd like to write a small custom database,
or a specialized GUI application, or a simple game.
Python is just the language for you.
```

Содержимое выходного файла:

```
54: If you do much work on computers, eventually you find
47: that there's some task you'd like to automate.
58: For example, you may wish to perform a search-and-replace
49: over a large number of text files, or rename and
55: rearrange a bunch of photo files in a complicated way.

53: Perhaps you'd like to write a small custom database,
53: or a specialized GUI application, or a simple game.
37: Python is just the language for you.
```

Задание № 6.14. Дан список ключевых слов `keywords` и текстовый файл. Вывести количество вхождений слов из списка в данном файле.

Пример:

```
keywords = ["file", "mode", "argument"]
```

Содержимое входного файла:

```
The first argument is a string containing
the filename. The second argument is another
string containing a few characters describing
the way in which the file will be used. mode
can be 'r' when the file will only be read,
'w' for only writing (an existing file with
the same name will be erased), and 'a' opens
the file for appending; any data written to
the file is automatically added to the end.
```

Вывод программы:

```
file      : 6
mode      : 1
argument:  2
```

Задание № 6.15. Дан текстовый файл с абзацами, разделенными пустой строкой, и число `paragraph` — номер абзаца. Вывести абзац с указанным номером или сообщение, что такого абзаца нет, если номер слишком большой.

Пример:

`paragraph = 2`

Содержимое входного файла:

Первый абзац. Первый абзац.
Первый абзац. Первый абзац.
Первый абзац. Первый абзац.

Второй абзац. Второй абзац.
Второй абзац. Второй абзац.
Второй абзац. Второй абзац.

Третий абзац. Третий абзац.
Третий абзац. Третий абзац.
Третий абзац. Третий абзац.

Вывод программы:

Второй абзац. Второй абзац.
Второй абзац. Второй абзац.
Второй абзац. Второй абзац.

Источники текстов

Источник текстов на английском языке — официальная документация по языку Python 3.8:

<https://docs.python.org/3.8/>

Источник палиндромов — Викисловарь:

https://ru.wiktionary.org/wiki/Приложение:Русские_палиндромы

Полезные ссылки

Официальный сайт по языку Python:

<https://www.python.org/>

Официальная документация по языку Python 3:

<https://docs.python.org/3/>

Онлайн-интерпретаторы языка Python:

https://www.onlinegdb.com/online_python_compiler

<https://www.online-python.com/>

<https://www.programiz.com/python-programming/online-compiler/>