

ЛАБОРАТОРНАЯ РАБОТА 7

Содержание

Модули	2
Стандартный модуль math	4
Стандартный модуль random	5
Стандартный модуль csv	6
Чтение csv-файлов	6
Запись csv-файлов	7
Требования к программам	9
Базовый шаблон для всех лабораторных	9
Пример программы	9
Индивидуальные задания	10
Источник данных	12
Полезные ссылки	12

Модули

Содержимое файла "lines.py" для задания [0.1](#):

```
def print_line(element="-", repeats=20):  
    print(element * repeats)  
  
def print_in_frame(word, symbol="#", width=20):  
    print_line(symbol, width)  
    print(f"{symbol}{word.center(width-2)}{symbol}")  
    print_line(symbol, width)
```

Задание № 0.1. Определить результат выполнения следующего кода:

- (a) `import lines`
 `lines.print_line()`
- (b) `lines.print_line(".", 3)`
- (c) `lines.print_in_frame("center")`
- (d) `lines.print_in_frame("module", symbol="/", width=10)`

Содержимое файла "lines_with_defaults.py" для задания [0.2](#):

```
default_symbol = "-"  
default_width = 20  
  
def print_line(element=None, repeats=None):  
    if element == None:  
        element = default_symbol  
    if repeats == None:  
        repeats = default_width  
    print(element * repeats)
```

Задание № 0.2. Определить результат выполнения следующего кода:

- (a) `import lines_with_defaults as lines`
 `lines.print_line()`
- (b) `lines.default_symbol = "x"`
 `lines.default_width = 10`

 `lines.print_line()`

Содержимое файла "verbose.py" для задания [0.3](#):

```
print("Module: ", __name__)
```

Задание № 0.3. Определить результат выполнения следующего кода:

- (a) `import verbose`
- (b) `import this`
- (c) `print("Module: ", __name__)`

Содержимое файла "digits.py" для задания [0.4](#):

```
def to_str(d):
    digits = [
        "ноль", "один", "два", "три", "четыре",
        "пять", "шесть", "семь", "восемь", "девять"
    ]

    if d in range(10):
        return digits[d]
    else:
        return None

def main():
    numbers = [8, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
    for d in numbers:
        print(to_str(d), end=" ")
    print()

if __name__ == "__main__":
    main()
```

Задание № 0.4. Определить результат выполнения

- (a) кода, записанного в файле "digits.py", как программы;
- (b) кода, импортирующего файл "digits.py", например:

```
import digits

year = 2023

for d in str(year):
    print(digits.to_str(int(d)))
```

Стандартный модуль math

Функция	Возвращаемое значение
<code>fabs(x)</code>	$ x $
<code>factorial(n)</code>	$n!$
<code>fsum($numbers$)</code>	Сумма чисел в списке $numbers$
<code>gcd(a, b)</code>	Наибольший общий делитель чисел a , b
<code>exp(x)</code>	e^x
<code>log(x [, $base$])</code>	$\log_{base} x$. Если $base$ не задано, то $\ln x$.
<code>pow(x, y)</code>	x^y
<code>sqrt(x)</code>	\sqrt{x}
<code>acos(x)</code>	$\arccos x$
<code>asin(x)</code>	$\arcsin x$
<code>atan(x)</code>	$\arctan x$
<code>cos(x)</code>	$\cos x$
<code>sin(x)</code>	$\sin x$
<code>tan(x)</code>	$\operatorname{tg} x$
<code>degrees(x)</code>	Угол в градусах. x — угол в радианах
<code>radians(x)</code>	Угол в радианах. x — угол в градусах

Константа	Значение
<code>pi</code>	Число π
<code>e</code>	Число e

Задание № 0.5. Определить следующие функции, и вычислить их значения при указанных значениях параметров и аргументов:

(a) $\varphi(t) = a \cos(\omega t) + b \sin(\omega t)$, при $a = 5$, $b = 3$, $\omega = 0.5$; $t = 2.0$

(b) $f(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2 \right]$, при $\mu = -2$, $\sigma = 0.5$; $x = -1$

(c) $p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$, при $\lambda = 4$; $k = 4$

Стандартный модуль random

Функция	Возвращаемое значение
<code>randint(<i>a</i>, <i>b</i>)</code>	Случайное целое число N такое, что $a \leq N \leq b$
<code>choice(<i>somelist</i>)</code>	Случайный элемент из списка <i>somelist</i>
<code>shuffle(<i>somelist</i>)</code>	None . Перемешивает порядок элементов в списке <i>somelist</i>
<code>sample(<i>seq</i>, <i>k</i>)</code>	Список из <i>k</i> случайно выбранных элементов из <i>seq</i>
<code>random()</code>	Случайное вещественное число X такое, что $0.0 \leq X < 1.0$
<code>uniform(<i>a</i>, <i>b</i>)</code>	Случайное вещественное число X такое, что $a \leq X \leq b$
<code>gauss(<i>mu</i>, <i>sigma</i>)</code>	Случайное вещественное число X , подчиняющееся распределению Гаусса с математическим ожиданием <i>mu</i> и среднеквадратичным отклонением <i>sigma</i>

Задание № 0.6. Определить результат выполнения следующего кода:

- (a)

```
import random
random.randint(1,10)
```
- (b)

```
color = ["Красный", "Зеленый", "Желтый", "Синий"]
random.choice(color)
```
- (c)

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
random.shuffle(numbers)
```
- (d)

```
probability = 0.6
total = 100
count = 0

for i in range(total):
    x = random.random()
    if x < probability:
        count += 1

print(count)
```
- (e)

```
random.uniform(-10.0, 10.0)
```
- (f)

```
random.gauss(0.0, 1.0)
```

Стандартный модуль csv

Чтение csv-файлов

Содержимое файла "months.csv" для задания 0.7(a):

Зима,Декабрь,Январь,Февраль
Весна,Март,Апрель,Май
Лето,Июнь,Июль,Август
Осень,Сентябрь,Октябрь,Ноябрь

Содержимое, отображаемое в табличном процессоре:

Зима	Декабрь	Январь	Февраль
Весна	Март	Апрель	Май
Лето	Июнь	Июль	Август
Осень	Сентябрь	Октябрь	Ноябрь

Содержимое файла "ages.csv" для заданий 0.7(b-c):

"Adam",20
"Bob",21
"Caron",21
"Dan",20
"Eva",22

Содержимое, отображаемое в табличном процессоре:

Adam	20
Bob	21
Caron	21
Dan	20
Eva	22

Содержимое файла "headers.csv" для задания 0.7(d):

"name","age"
"Adam",20
"Bob",21
"Caron",21
"Dan",20
"Eva",22

Содержимое, отображаемое в табличном процессоре:

name	age
Adam	20
Bob	21
Caron	21
Dan	20
Eva	22

Задание № 0.7. Определить результат выполнения следующего кода:

- (a)

```
import csv
with open("months.csv", newline="") as csvfile:
    datareader = csv.reader(csvfile)
    for line in datareader:
        print(line)
```
- (b)

```
import csv
with open("ages.csv", newline="") as csvfile:
    datareader = csv.reader(csvfile)
    for row in datareader:
        if int(row[1]) > 20:
            print(row[0])
```

```
(c) import csv
    with open("ages.csv", newline="") as csvfile:
        datareader = csv.reader(csvfile, quoting=csv.QUOTE_NONNUMERIC)
        for row in datareader:
            if row[1] > 20:
                print(row[0])

(d) import csv
    with open("headers.csv", newline="") as csvfile:
        datareader = csv.DictReader(csvfile, quoting=csv.QUOTE_NONNUMERIC)
        for record in datareader:
            if record["age"] > 20:
                print(record["name"])
```

Запись csv-файлов

Задание № 0.8. Определить результат выполнения следующего кода:

```
(a) import csv

row1 = ["Winter", "December", "January", "February"]
row2 = ["Spring", "March", "April", "May"]
row3 = ["Summer", "June", "July", "August"]
row4 = ["Autumn", "September", "October", "November"]

with open("out_months.csv", "w", newline="") as csvfile:
    datawriter = csv.writer(csvfile)
    datawriter.writerow(row1)
    datawriter.writerow(row2)
    datawriter.writerow(row3)
    datawriter.writerow(row4)

(b) import random
    import csv

names = ["Adam", "Bob", "Caron", "Dan", "Eva"]

with open("out_ages.csv", "w", newline="") as csvfile:
    datawriter = csv.writer(csvfile, quoting=csv.QUOTE_NONNUMERIC)
    for name in names:
        datawriter.writerow([name, random.randint(18,22)])
```

(c) `import csv`

```
headers = ["name", "language"]
names = ["Adam", "Bob", "Caron", "Dan", "Eva"]
langs = ["English", "German", "French"]
```

```
row1 = {"name": "Adam", "language": "English"}
row2 = {"name": "Bob", "language": "French"}
row3 = {"name": "Caron", "language": "English"}
row4 = {"name": "Dan", "language": "English"}
row5 = {"name": "Eva", "language": "German"}
```

```
with open("out_languages.csv", "w", newline="") as csvfile:
    datawriter = csv.DictWriter(csvfile, headers)
    datawriter.writeheader()
    datawriter.writerow(row1)
    datawriter.writerow(row2)
    datawriter.writerow(row3)
    datawriter.writerow(row4)
    datawriter.writerow(row5)
```

```
(d) import random
import csv
```

```
fname = "out_rand_languages.csv"
headers = ["name", "language"]
```

```
names = ["Adam", "Bob", "Caron", "Dan", "Eva"]
languages = ["English", "German", "French"]
```

```
with open(fname, "w", newline="") as csvfile:
    datawriter = csv.DictWriter(csvfile, headers)
    datawriter.writeheader()
    for name in names:
        datawriter.writerow({"name":      name,
                              "language": random.choice(languages)})
```


Требования к программам

1. Файл с исходным кодом называть по шаблону: «Фамилия_номер_задания» английским алфавитом (пример: `Ivanov_7_42.py`).
2. Файл с исходным кодом должен начинаться с многострочного описательного комментария (см. базовый шаблон ниже).
3. Все имена выбирать разумными.
4. Оформлять понятный ввод данных / вывод результата.
5. Использовать входные файлы в формате `csv`. Демонстрационные файлы размещены в каталоге `"lab_7_data"`.
6. Чтение и запись файлов осуществлять с помощью функций и методов стандартного модуля `csv`.

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 № 7.0
2
3 Дан файл min_level.csv с величиной прожиточного минимума на 2023 г.
4 по субъектам Сибирского федерального округа РФ в рублях в месяц.
5 Файл содержит 4 столбца: "Субъект РФ", "Трудоспособное население",
6 "Пенсионеры", "Дети".
7
8 1. Вывести все субъекты РФ, величина прожиточного минимума
9 трудоспособного населения которых превышает 15000 рублей в месяц.
10 2. Определить среднюю величину прожиточного минимума для детей.
11 '''
12 import csv
13
14 infname = "lab_7_data/min_level.csv"
15
16 def task_example_1():
17     with open(infname, newline="") as csvfile:
18         datareader = csv.DictReader(csvfile, quoting=csv.QUOTE_NONNUMERIC)
19         print("1. Субъекты с величиной прожиточного минимума > 15000:")
20         for data in datareader:
21             level = data['Трудоспособное население']
22             if level > 15000:
23                 print(f"\t{data['Субъект РФ']:25} {level}")
```

```

24 def task_example_2():
25     with open(infile, newline="") as csvfile:
26         datareader = csv.DictReader(csvfile, quoting=csv.QUOTE_NONNUMERIC)
27         children = []
28         for data in datareader:
29             children.append(data["Дети"])
30     print("2. Средняя величина прожиточного минимума для детей:",
31           sum(children)/len(children), "руб.")
32
33 task_example_1()
34 task_example_2()

```

Индивидуальные задания

Задание № 7.1. Дан файл `cities_millionaires.csv` с численностью населения городов-миллионеров РФ. Файл содержит 4 столбца: «Город», «Субъект РФ», «Численность на 01.01.2022», «Численность на 01.01.2021».

Вывести все города (вместе с субъектом РФ), численность населения в которых на 01.01.2022 составляла от 1 100 000 до 1 200 000 человек.

Задание № 7.2. Дан файл `cities_millionaires.csv` с численностью населения городов-миллионеров РФ. Файл содержит 4 столбца: «Город», «Субъект РФ», «Численность на 01.01.2022», «Численность на 01.01.2021».

Вывести города (вместе с субъектом РФ) и изменение численности населения с 01.01.2021 по 01.01.2022.

Задание № 7.3. Дан файл `cities_millionaires.csv` с численностью населения городов-миллионеров РФ. Файл содержит 4 столбца: «Город», «Субъект РФ», «Численность на 01.01.2022», «Численность на 01.01.2021».

Вывести среднюю численность населения городов, численность населения которых на 01.01.2022 не превышала 1 500 000 человек.

Задание № 7.4. Дан файл `Siberia.csv` с численностью населения субъектов Сибирского федерального округа РФ на 1 января 2022 г. Файл содержит 4 столбца: «Субъект РФ», «Все население», «Городское население», «Сельское население».

Вывести список субъектов РФ вместе с долей городского населения.

Задание № 7.5. Дан файл `Siberia.csv` с численностью населения субъектов Сибирского федерального округа РФ на 1 января 2022 г. Файл содержит 4 столбца: «Субъект РФ», «Все население», «Городское население», «Сельское население».

Вывести список субъектов РФ, городское население в которых составляет от 1 000 000 до 2 000 000 человек.

Задание № 7.6. Дан файл `Siberia.csv` с численностью населения субъектов Сибирского федерального округа РФ на 1 января 2022 г. Файл содержит 4 столбца: «Субъект РФ», «Все население», «Городское население», «Сельское население».

Вывести субъект РФ с максимальной долей сельского населения.

Задание № 7.7. Дан файл `labour.csv` с численностью рабочей силы в возрасте 15 лет и старше по субъектам Сибирского федерального округа РФ за 2020–2022 гг. в тысячах человек. Файл содержит 4 столбца: «Субъект РФ», «2020», «2021», «2022».

Вывести субъекты РФ, в которых численность рабочей силы за 2022 г. превышает 1 000 тыс. человек.

Задание № 7.8. Дан файл `labour.csv` с численностью рабочей силы в возрасте 15 лет и старше по субъектам Сибирского федерального округа РФ за 2020–2022 гг. в тысячах человек. Файл содержит 4 столбца: «Субъект РФ», «2020», «2021», «2022».

Вывести субъекты РФ и изменение численности рабочей силы с 2020 по 2022 г.

Задание № 7.9. Дан файл `labour.csv` с численностью рабочей силы в возрасте 15 лет и старше по субъектам Сибирского федерального округа РФ за 2020–2022 гг. в тысячах человек. Файл содержит 4 столбца: «Субъект РФ», «2020», «2021», «2022».

Вывести субъекты РФ вместе с годом, в котором наблюдалась максимальная численность рабочей силы.

Задание № 7.10. Дан файл `scientific_degrees.csv` с численностью исследователей, имеющих ученую степень, по субъектам Сибирского федерального округа РФ за 2021 г. Файл содержит 4 столбца: «Субъект РФ», «Всего», «Доктор наук», «Кандидат наук».

Вывести субъекты РФ вместе с долей исследователей, имеющих ученую степень кандидата наук.

Задание № 7.11. Дан файл `scientific_degrees.csv` с численностью исследователей, имеющих ученую степень, по субъектам Сибирского федерального округа РФ за 2021 г. Файл содержит 4 столбца: «Субъект РФ», «Всего», «Доктор наук», «Кандидат наук».

Вывести субъекты РФ, в которых число исследователей, имеющих ученую степень, превышает 500 человек.

Задание № 7.12. Дан файл `scientific_degrees.csv` с численностью исследователей, имеющих ученую степень, по субъектам Сибирского федерального округа РФ за 2021 г. Файл содержит 4 столбца: «Субъект РФ», «Всего», «Доктор наук», «Кандидат наук».

Вывести субъекты РФ, в которых число исследователей, имеющих ученую степень доктора наук, превышает 100 человек.

Задание № 7.13. Дан файл `census_2020.csv` с данными Всероссийской переписи населения 2020 года по Омской области. Файл содержит 4 столбца: «Возрастная группа», «Мужчины и женщины», «Мужчины», «Женщины».

Вывести возрастные группы (вместе с количеством мужчин и женщин), численность которых превышает 100 000 человек.

Задание № 7.14. Дан файл `census_2020.csv` с данными Всероссийской переписи населения 2020 года по Омской области. Файл содержит 4 столбца: «Возрастная группа», «Мужчины и женщины», «Мужчины», «Женщины».

Вывести возрастные группы вместе с долей женщин.

Задание № 7.15. Дан файл `census_2020.csv` с данными Всероссийской переписи населения 2020 года по Омской области. Файл содержит 4 столбца: «Возрастная группа», «Мужчины и женщины», «Мужчины», «Женщины».

Вывести возрастные группы вместе с долей мужчин, в которых эта доля менее 50 %.

Источник данных

Источник данных для индивидуальных заданий — Росстат. Официальный сайт Федеральной службы государственной статистики:

<https://rosstat.gov.ru/>

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/>