

ЛАБОРАТОРНАЯ РАБОТА 8

Содержание

| | |
|---|----|
| Введение в классы | 2 |
| Стандартный модуль <code>datetime</code> | 3 |
| Сортировка списков с помощью метода <code>sort()</code> и стандартной функции <code>sorted()</code> | 5 |
| Требования к программам | 6 |
| Базовый шаблон для всех лабораторных | 6 |
| Пример программы | 7 |
| Индивидуальные задания | 8 |
| Полезные ссылки | 11 |
| Что дальше? | 11 |

Введение в классы

Задание № 0.1. Определить результат выполнения следующего кода:

(a) `first_name, last_name = "John", "Smith"`

```
full_name = first_name + " " + last_name
name_with_initial = first_name[0] + ". " + last_name

print(full_name)
print(name_with_initial)
```

(b) `def full_name(person):`
 `first_name, last_name = person`
 `return first_name + " " + last_name`

`def name_with_initial(person):`
 `first_name, last_name = person`
 `return first_name[0] + ". " + last_name`

`person1 = ("John", "Smith")`
`person2 = ("Eva", "Smith")`

`print(full_name(person1))`
`print(name_with_initial(person2))`

(c) `class Person:`
 `def __init__(self, fname, lname):`
 `self.first_name, self.last_name = fname, lname`

`def full_name(self):`
 `return self.first_name + " " + self.last_name`

`def name_with_initial(self):`
 `return self.first_name[0] + ". " + self.last_name`

`student = Person("John", "Smith")`
`student.full_name()`
`student.name_with_initial()`
`student.first_name`
`student.last_name`

Стандартный модуль datetime

Задание № 0.2. Определить результат выполнения следующего кода:

- (a) `from datetime import date`
`date(2023, 5, 31)`
- (b) `date.fromisoformat("2023-04-30")`
- (c) `date.today()`
- (d) `d = date(2023, 6, 1)`
`print("year = ", d.year)`
`print("month = ", d.month)`
`print("day = ", d.day)`
- (e) `d = date(2023, 9, 1)`
`date_std = d.strftime("%d.%m.%y") # 01.09.23`
`date_short = d.strftime("%a %d %b %Y") # Fri 01 Sep 2023`
`date_long = d.strftime("%A %d %B %Y") # Friday 01 September 2023`

Задание № 0.3. Определить результат выполнения следующего кода:

- (a) `from datetime import time`
`time(12, 25, 59)`
- (b) `time.fromisoformat("04:23:01")`
- (c) `t = time(12, 25, 59)`
`print("hrs: ", t.hour)`
`print("mins:", t.minute)`
`print("secs:", t.second)`
- (d) `t = time(13, 5, 25)`
`time_24 = t.strftime("%H:%M:%S") # 13:05:25`
`time_12 = t.strftime("%I:%M:%S %p") # 01:05:25 PM`

Задание № 0.4. Определить результат выполнения следующего кода:

- (a) `from datetime import datetime`
`datetime(2023, 5, 30, hour=13, minute=5, second=25)`
- (b) `datetime.fromisoformat("2023-11-04 14:05:23")`

- (c)

```
present_moment = datetime.today()
present_moment.date()
present_moment.time()
```
- (d)

```
present_moment.strftime("%A %d.%m.%Y %H:%M:%S")
```
- (e)

```
from datetime import date, time, datetime
mydate = date(2022, 12, 31)
mytime = time(23, 59, 59)
last_year_moment = datetime.combine(mydate, mytime)
```

Задание № 0.5. Определить результат выполнения следующего кода:

- (a)

```
from datetime import timedelta
timedelta(days=1, seconds=10, minutes=20, hours=3, weeks=4)
```
- (b)

```
from datetime import date, timedelta
delta = timedelta(days=1)
d1 = date(2023, 5, 31)
d2 = d1 + delta
```
- (c)

```
start = date(2023, 2, 10)
delta = timedelta(weeks=1)

for i in range(16):
    date = start + i*delta
    print(date)
```
- (d)

```
from datetime import datetime

credit = datetime(2023, 6, 10, 10, 30, 00)
present = datetime.today()
rest = credit - present
```
- (e)

```
from datetime import date, time, timedelta, datetime

dt = timedelta(hours=1, minutes=35)
t1 = time(14, 30, 00)

date = date.today()
d1 = datetime.combine(date, t1)
d2 = d1 + dt

t2 = d2.time()
```

Сортировка списков с помощью метода `sort()` и стандартной функции `sorted()`

Задание № 0.6. Определить результат выполнения следующего кода:

- (a)

```
mylist = [5, 9, 8, 3, 7, 6, 4, 2, 0, 1]
mylist.sort()
```
- (b)

```
mylist = [5, 9, 8, 3, 7, 6, 4, 2, 0, 1]
res = sorted(mylist)
```
- (c)

```
mylist = [5, 9, 8, 3, 7, 6, 4, 2, 0, 1]
res = sorted(mylist, reverse=True)
```

Задание № 0.7. Определить результат выполнения следующего кода:

- (a)

```
mydict = {"Dan": 22, "Adam": 20, "Bob": 18, "Eva": 19, "Frank": 21}
for name in mydict:
    print(f"{name:6}: {mydict[name]}")
```
- (b)

```
mydict = {"Dan": 22, "Adam": 20, "Bob": 18, "Eva": 19, "Frank": 21}
for name in sorted(mydict):
    print(f"{name:6}: {mydict[name]}")
```

Задание № 0.8. Определить результат выполнения следующего кода:

- (a)

```
def print_ages(ages_list):
    format_str = "{0:6}: {1:3}"
    print(format_str.format("name", "age"))
    for elem in ages_list:
        print(format_str.format(elem["name"], elem["age"]))

def sort_by_name(elem):
    return elem["name"]

mylist = [
    {"name" : "Dan",   "age" : 22},
    {"name" : "Adam",  "age" : 20},
    {"name" : "Bob",   "age" : 18},
    {"name" : "Eva",   "age" : 19},
    {"name" : "Frank", "age" : 21}
]
```

```
res_list = sorted(mylist, key=sort_by_name)
print_ages(res_list)
```

(b) `def sort_by_age(elem):`
 `return elem["age"]`

```
res_list = sorted(mylist, key=sort_by_age)
print_ages(res_list)
```

(c) `res_list = sorted(mylist, key=lambda elem : elem["age"])`
 `print_ages(res_list)`

(d) `from operator import itemgetter`
 `res_list = sorted(mylist, key=itemgetter("age"))`
 `print_ages(res_list)`

Требования к программам

1. Файл с исходным кодом называть по шаблону: «Фамилия_номер_задания» английским алфавитом (пример: `Ivanov_8_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. Входные и выходные файлы должны быть в формате `csv`. Демонстрационный входной файл размещен в каталоге `"lab_8_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 № 8.0
2
3  Дан файл lab_8_data/example_data.csv с 4 столбцами:
4      "name" (имя),
5      "birthday" (дата рождения),
6      "language" (изучаемый иностранный язык),
7      "score" (балл).
8
9  Создать новый файл sorted_example_data.csv,
10 в который записать данные в 4 столбца:
11 "name", "language", "age", "score", в отсортированном порядке:
12     сначала по изучаемому языку (в алфавитном порядке),
13     затем по возрасту (в возрастающем порядке),
14     затем по баллу (в убывающем порядке).
15 '''
16 from operator import itemgetter
17 from datetime import date
18 import csv
19
20 infname = "lab_8_data/example_data.csv"
21 outfname = "sorted_example_data.csv"
22
23 #####
24 # Чтение входного файла, формирование списка всех данных
25
26 mylist = []
27 with open(infname, newline="") as csvfile:
28     datareader = csv.DictReader(csvfile, quoting=csv.QUOTE_NONNUMERIC)
29     for data in datareader:
30         birthday = date.fromisoformat(data["birthday"])
31         age = date.today().year - birthday.year
32         data["age"] = age
33         del data["birthday"]
34
35         mylist.append(data)
36
37 #####
38 # Сортировка
39
40 mylist.sort(key=itemgetter("score"), reverse=True)
41 mylist.sort(key=itemgetter("age"))
42 mylist.sort(key=itemgetter("language"))
43
44 #####
45 # Запись выходного файла
46
47 headers = ["name", "language", "age", "score"]
48
49 with open(outfname, "w", newline="") as csvfile:
50     datawriter = csv.DictWriter(csvfile, headers, quoting=csv.QUOTE_NONNUMERIC)
51     datawriter.writeheader()
52     for data in mylist:
53         datawriter.writerow(data)
54
55 #####
```

Индивидуальные задания

Задание № 8.1. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по возрасту (по убыванию), затем по домашнему животному (в алфавитном порядке).

| | | | | | |
|------|-------|--------|-----------|--------|-------|
| name | age ↑ | family | education | height | pet ↓ |
|------|-------|--------|-----------|--------|-------|

Задание № 8.2. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по имени (в алфавитном порядке), затем по количеству членов семьи (по убыванию).

| | | | | | |
|--------|-----|----------|-----------|--------|-----|
| name ↓ | age | family ↑ | education | height | pet |
|--------|-----|----------|-----------|--------|-----|

Задание № 8.3. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по наличию образования (есть / нет), затем по росту (по возрастанию).

| | | | | | |
|------|-----|--------|-------------|----------|-----|
| name | age | family | education ↑ | height ↓ | pet |
|------|-----|--------|-------------|----------|-----|

Задание № 8.4. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по имени (в алфавитном порядке), затем по домашнему животному (в алфавитном порядке).

| | | | | | |
|--------|-----|--------|-----------|--------|-------|
| name ↓ | age | family | education | height | pet ↓ |
|--------|-----|--------|-----------|--------|-------|

Задание № 8.5. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по возрасту (по убыванию), затем по количеству человек в семье (по возрастанию).

| | | | | | |
|------|-------|----------|-----------|--------|-----|
| name | age ↑ | family ↓ | education | height | pet |
|------|-------|----------|-----------|--------|-----|

Задание № 8.6. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по наличию высшего образования (есть / нет), затем по домашнему животному (в алфавитном порядке).

| | | | | | |
|------|-----|--------|-------------|--------|-------|
| name | age | family | education ↑ | height | pet ↓ |
|------|-----|--------|-------------|--------|-------|

Задание № 8.7. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по количеству членов семьи (по возрастанию), затем по росту (по убыванию).

| | | | | | |
|------|-----|----------|-----------|----------|-----|
| name | age | family ↓ | education | height ↑ | pet |
|------|-----|----------|-----------|----------|-----|

Задание № 8.8. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по имени (в алфавитном порядке), затем по росту (по убыванию).

| | | | | | |
|--------|-----|--------|-----------|----------|-----|
| name ↓ | age | family | education | height ↑ | pet |
|--------|-----|--------|-----------|----------|-----|

Задание № 8.9. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по росту (по убыванию), затем по домашнему животному (в алфавитном порядке).

| | | | | | |
|------|-----|--------|-----------|----------|-------|
| name | age | family | education | height ↑ | pet ↓ |
|------|-----|--------|-----------|----------|-------|

Задание № 8.10. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по имени (в алфавитном порядке), затем по возрасту (по убыванию).

| | | | | | |
|--------|-------|--------|-----------|--------|-----|
| name ↓ | age ↑ | family | education | height | pet |
|--------|-------|--------|-----------|--------|-----|

Задание № 8.11. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по количеству членов семьи (в убывающем порядке), затем по наличию высшего образования (есть / нет).

| | | | | | |
|------|-----|----------|-------------|--------|-----|
| name | age | family ↑ | education ↑ | height | pet |
|------|-----|----------|-------------|--------|-----|

Задание № 8.12. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по возрасту (по убыванию), затем по росту (по убыванию).

| | | | | | |
|------|-------|--------|-----------|----------|-----|
| name | age ↑ | family | education | height ↑ | pet |
|------|-------|--------|-----------|----------|-----|

Задание № 8.13. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по имени (в алфавитном порядке), затем по наличию высшего образования (есть / нет).

| | | | | | |
|--------|-----|--------|-------------|--------|-----|
| name ↓ | age | family | education ↑ | height | pet |
|--------|-----|--------|-------------|--------|-----|

Задание № 8.14. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по количеству членов семьи (в убывающем порядке), затем по домашнему животному (в алфавитном порядке).

| | | | | | |
|------|-----|----------|-----------|--------|-------|
| name | age | family ↑ | education | height | pet ↓ |
|------|-----|----------|-----------|--------|-------|

Задание № 8.15. Дан файл `lab_8_data/task_data.csv` с 6 столбцами: «name» (имя), «age» (возраст), «family» (количество человек в семье), «education» (наличие высшего образования), «height» (рост в метрах), «pet» (домашнее животное).

Записать данные в файл `sorted_task_data.csv`, отсортировав их сначала по возрасту (по возрастанию), затем по наличию высшего образования (есть / нет).

| | | | | | |
|------|-------|--------|-------------|--------|-----|
| name | age ↓ | family | education ↑ | height | pet |
|------|-------|--------|-------------|--------|-----|

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/>

Что дальше?

NumPy — библиотека для научных вычислений (ядро библиотеки — многомерные массивы однородных данных):

<https://numpy.org/>

pandas — библиотека для обработки и анализа данных (ядро библиотеки — массивы разнородных данных):

<https://pandas.pydata.org/>

SciPy — библиотека для научных вычислений (ядро библиотеки — фундаментальные алгоритмы и методы вычисления специальных функций):

<https://scipy.org/>

Matplotlib — библиотека для визуализации данных:

<https://matplotlib.org/>