

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования Московский государственный технический университет им.
Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки информации и управления»



Курс «Оперативный анализ данных»

Лабораторная работа № 3

Выполнил: Чувашов Е.Е., ИУ5-51 _____

Цель лабораторной работы

Важно выполнять все задачи последовательно . С 1 по 5 задачу формируется модуль librip , с помощью которого будет выполняться задание 6 на реальных данных из жизни. Весь вывод на экран (даже в столбик) необходимо реализовывать одной строкой .

ex1

```
#!/usr/bin/env python3
```

```
from librip.gen import field, gen_random
```

```
goods = [
```

```
    {'title': 'Ковер', 'price': 2000, 'color': 'green'},
```

```
    {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
```

```
]
```

```
#print([x for x in field(goods, 'title', 'price')])
```

```
#print([x for x in field(goods, 'title')])
```

```
print([x for x in gen_random(1, 5, 5)])
```

ex2

```
#!/usr/bin/env python3
```

```
from librip.gen import gen_random
```

```
from librip.iterators import Unique
```

```
data1 = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2]
```

```
data2 = gen_random(1, 3, 10)
```

```
# Реализация задания 2
```

```
#print([x for x in Unique(data1)])
```

```
#print([x for x in Unique(data2)])
```

```
#print([x for x in Unique(['A','a','B','b'])])
```

```
print([x for x in Unique(['A','a','B','b'], ignore_case=True)])
```

ex3

```
#!/usr/bin/env python3
```

```
data = [4, -30, 100, -100, 123, 1, 0, -1, -4]
```

```
res = sorted(data, key=lambda x: abs(x))
```

```
print(res)
```

```
ex4
```

```
from librip.decorators import print_result
```

```
# Необходимо верно реализовать print_result
```

```
# и задание будет выполнено
```

```
@print_result
```

```
def test_1():
```

```
    return 1
```

```
@print_result
```

```
def test_2():
```

```
    return 'iu'
```

```
@print_result
```

```
def test_3():
```

```
    return {'a': 1, 'b': 2}
```

```
@print_result
```

```
def test_4():
```

```
    return [1, 2]
```

```
test_1()
```

```
test_2()
```

```
test_3()
```

```
test_4()
```

```
ex5
```

```
from time import sleep
```

```
from librip.ctxmngers import timer
```

```
with timer():
```

```
    sleep(2.5)
```

```
ex6
```

```
#!/usr/bin/env python3
```

```
import json
```

```
import sys
```

```
from librip.ctxmngers import timer
```

```
from librip.decorators import print_result
```

```
from librip.gen import field, gen_random
```

```
from librip.iterators import Unique as unique
```

```
path = sys.argv[1]
```

```
#print(path)
```

```
with open(path, encoding="utf8") as f:
```

```
    data = json.load(f)
```

```
@print_result
```

```
def f1(arg):
```

```
    return sorted([st for st in unique(field(arg,'job-name'), ignore_case=True)])
```

```
@print_result
```

```
def f2(arg):
```

```
    return list(filter(lambda x: x.startswith("Программист") , arg))
```

```
@print_result
```

```
def f3(arg):
```

```
return list(map(lambda x: x + ' с опытом Python', arg))
```

```
@print_result
```

```
def f4(arg):
```

```
    salary = gen_random(100000, 200000, len(arg))
```

```
    return [i + ', зарплата ' + str(j) + ' руб.' for i,j in zip(arg, salary)]
```

```
#print(f4(f3(f2(f1(data)))))
```

```
with timer():
```

```
    f4(f3(f2(f1(data))))
```

Вывод:

Часть вывода опущена

учитель технологии

учитель технологии (мальчики), ОБЖ, английского языка

учитель физики и информатики

учитель химии

федеральное государственное унитарное предприятие "Российская телевизионная и

радиовещательная сеть

фельдшер - лаборант бактериологической лаборатории

фельдшер отделения скорой медицинской помощи

фельдшер поликлиники

фельдшер скорой помощи

фельдшер фельдшерско - акушерского пункта

физик-эксперт

формовщик

фрезеровщик

фтизиатрия

химик

художник-постановщик

швея - мотористка

шиномонтаж

шлифовщик 5 разряда

шлифовщик механического цеха

эколог

электромонтер -линейщик по монтажу воздушных линий высокого напряжения и контактной сети

электромонтер по испытаниям и измерениям 4-6 разряд

электромонтер станционного телевизионного оборудования

электросварщик

энтомолог

юрисконсульт 2 категории

f2

Программист

Программист / Senior Developer

Программист 1С

Программист C#

Программист C++

Программист C++/C#/Java

Программист/ Junior Developer

Программист/ технический специалист

Программист-разработчик информационных систем

f3

Программист с опытом Python

Программист / Senior Developer с опытом Python

Программист 1С с опытом Python

Программист C# с опытом Python

Программист C++ с опытом Python

Программист C++/C#/Java с опытом Python

Программист/ Junior Developer с опытом Python

Программист/ технический специалист с опытом Python

Программист-разработчик информационных систем с опытом Python

f4

Программист с опытом Python, зарплата 152470 руб.

Программист / Senior Developer с опытом Python, зарплата 175872 руб.

Программист 1С с опытом Python, зарплата 183062 руб.

Программист C# с опытом Python, зарплата 137075 руб.

Программист C++ с опытом Python, зарплата 177673 руб.

Программист C++/C#/Java с опытом Python, зарплата 139468 руб.

Программист/ Junior Developer с опытом Python, зарплата 188472 руб.

Программист/ технический специалист с опытом Python, зарплата 184804 руб.

Программист-разработчик информационных систем с опытом Python, зарплата 129037 руб.

Execution time 0.76916