

Московский государственный технический университет им. Н.Э. Баумана  
Факультет «Информатика и системы управления» Кафедра «Автоматизированные  
системы обработки информации и управления»



**Отчет**  
**по лабораторной работе №3**  
**По курсу**  
**“Разработка интернет-приложений”**

**Выполнила:**  
**Студентка группы ИУ5-51**  
**Соболева Е.Д.**

## Цель 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.

Librip:

Decorators.py

```
def print_result(func):
    def dec(*args, **kwargs):
        print(func.__name__)
        f = func(*args, **kwargs)
        if type(f) is list:
            [print(i) for i in f]
        elif type(f) is dict:
            [print(key, ' = ', f[key]) for key in f]
        else:
            print(f)
        return f
    return dec
```

Gens.py

```
def field(items, *args):
    assert len(args) > 0
    if len(args) == 1:
        key = args[0]
        for d in items:
            dict1 = d[key]
            yield dict1
    else:
        for d in items:
            dict1 = {key: d[key] for key in args}
            yield dict1

def gen_random(begin, end, num_count):
    x = (random.randint(begin, end) for i in range(num_count))
    return x
```

iterators.py

```
class Unique(object):
    def __init__(self, items, **kwargs):
        if type(items) is list:
            self.items = iter(items)
        else:
            self.items = items
        self.ignore_case = kwargs.get('ignore_case', False)
        self.mas = set()

    def __next__(self):
        while True:
            try:
                item = next(self.items)
                if type(item) is str and self.ignore_case is True:
                    i = item.lower()
                else:
                    i = item
                if i not in self.mas:
                    self.mas.add(i)
                    return i
            except Exception:
                raise StopIteration

    def __iter__(self):
        return self
```

ctxmgrs.py

```
from time import time
```

```
import contextlib
@contextlib.contextmanager
def timer():
    t1 = time()
    yield
    t2 = time()
    t=t2-t1
    print(t)
```

ex\_1.py

```
#!/usr/bin/env python3
from librip.gens import field
from librip.gens import gen_random
goods = [
    {'title': 'Ковер', 'price': 2000, 'color': 'green'},
    {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'},
]

# Реализация задания 1
print([x for x in field(goods, 'title')])
print([x for x in field(goods, 'title', 'color')])

print([x for x in gen_random(1, 3, 5)])
```

Результат:

```
['Ковер', 'Диван для отдыха']
[{'title': 'Ковер', 'color': 'green'}, {'title': 'Диван для отдыха', 'color':
'black'}]
[2, 1, 3, 1, 1]
```

ex\_2.py

```
#!/usr/bin/env python3
from librip.gens 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)
data3 = ['A', 'a', 'C', 'B', 'b', 'c']
# Реализация задания 2
print([x for x in Unique(data1)])
print([x for x in Unique(data2)])
print([x for x in Unique(data3, ignore_case=True)])
print([x for x in Unique(data3)])
```

Результат:

```
[1, 2]
[2, 3, 1]
['a', 'c', 'b']
['A', 'a', 'C', 'B', 'b', 'c']
```

ex\_3.py

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

data = [4, -30, 100, -100, 123, 1, 0, -1, -4]
# Реализация задания 3
print(sorted(data, key=lambda x: (abs(x))))
```

Результат:

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

ex\_4.py

```
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()

```

Результат:

test\_1

1

test\_2

iu

test\_3

a = 1

b = 2

test\_4

1

2

ex\_5.py

```

from time import sleep
from librip.ctxmgrs import timer

with timer():
    sleep(5.5)

```

Результат:

5.5001890659332275

ex\_6.py

```

#!/usr/bin/env python3
import json
import sys
from librip.ctxmgrs import timer
from librip.decorators import print_result
from librip.gens import field, gen_random
from librip.iterators import Unique as unique

```

```

with open('G:\PMI\lab4-master\lab4-master\data_light_cp1251.json') as f:
    data = json.load(f)

#@print_result
def f1(arg):
    return list(set([(arg[i]['job-name']).lower() for i in range(len(arg))]))

f1(data)

#@print_result
def f2(arg):
    return list(filter((lambda x: x if x.startswith("программист") else
None), arg))

#@print_result
def f3(arg):
    return list(map(lambda x: x+' с опытом Python', arg))

@print_result
def f4(arg):
    a = list()
    [a.append('{},зарплата {}'.format(x, y)) for x, y in zip(arg, [x for x
in gen_random(100000, 200000, len(arg))])]
    return a

with timer():
    f4(f3(f2(f1(data))))

```

Результат:

f4

программист/ технический специалист с опытом Python, зарплата 135771руб  
 программист c++/c#/java с опытом Python, зарплата 198842руб  
 программист 1с с опытом Python, зарплата 169191руб  
 программист / senior developer с опытом Python, зарплата 178511руб  
 программист c++ с опытом Python, зарплата 193054руб  
 программист/ junior developer с опытом Python, зарплата 176548руб  
 программист с опытом Python, зарплата 119213руб  
 программист-разработчик информационных систем с опытом Python, зарплата 100126руб  
 программист c# с опытом Python, зарплата 170711руб  
 0.007018566131591797