

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Базовые компоненты интернет-технологий»

**Отчет по лабораторной работе №3-4
«Функциональные возможности языка Python»**

Выполнил:

студент группы ИУ5-32
Вольвач Михаил

Подпись и дата: 29.12.2022

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата: 29.12.2022

Москва, 2022

Задание

Задание лабораторной работы состоит из решения нескольких задач. Файлы, содержащие решения отдельных задач, должны располагаться в пакете `lab_python_fr`. Решение каждой задачи должно располагаться в отдельном файле. При запуске каждого файла выдаются тестовые результаты выполнения соответствующего задания.

Текст программы

Файл cm_timer_1.py

```
import time

class Cm_timer_1:

    def __init__(self):
        self.result = 0

    def __enter__(self):
        self.start_time = time.time()
        return "cm_timer_1"

    def __exit__(self, exp_type, exp_value, traceback):
        if exp_type is not None:
            print(exp_type, exp_value, traceback)
        else:
            self.end_time = time.time()
            self.result = self.end_time - self.start_time
            print("time:", self.result)
```

Файл cm_timer_2.py

```
from contextlib import contextmanager
from time import time

@contextmanager
def Cm_timer_2():
    start_time = time()
    yield "cm_timer_2"
    print("time:", time() - start_time)
```

Файл field.py

```
from print_result import print_result

@print_result
def field(items, *args):
    assert len(args) > 0

    for item in items:
        tmp_res = {}

        if len(args) > 1:
            for arg in args:
                tmp_res[arg] = item[arg]
            yield tmp_res
        else:
            yield item[args[0]]
```

Файл gen_random.py

```
import random
from print_result import print_result

# @print_result
def gen_random(num_count, begin, end):
    return [random.randrange(begin, end) for _ in range(num_count)] if num_count >
1 else random.randrange(begin, end)
```

Файл print_result.py

```
from types import GeneratorType

def print_result(function_to_decorate):
    """
    :rtype: object
    """
    def decorated_func(*args):
        # print("start")
        result = function_to_decorate(*args)
        if type(result) == dict:
            for item in result:
                print(item, "=", result[item])
        elif type(result) == list or type(result) == GeneratorType:
            for item in result:
                print(item)
        else: print(result)
        # print("end")

    return decorated_func
```

Файл sort.py

```
def num_sort(data, rev=False):
    return sorted(data, key=int.__abs__, reverse=rev)

def num_lambda_sort(data, rev=False):
    return sorted(data, key=lambda elem: abs(elem), reverse=rev)

def sort(data, rev=False):
    return sorted(data, reverse=rev)
```

Файл unique.py

```
class Unique(object):
    def __init__(self, items, **kwargs):
        self._data = items
        self._ignore_case = kwargs['ignore_case'] if 'ignore_case' in kwargs.keys()
    else False
```

```

def __next__(self):
    result = []
    for elem in self._data:
        elem = elem.lower() if type(elem) == str and self._ignore_case else
elem
        if elem not in result:
            result.append(elem)
    return result

def __iter__(self):
    return self

```

Файл process_data.py

```
import json
```

```

from cm_timer_1 import Cm_timer_1
from gen_random import gen_random
from print_result import print_result
from sort import sort
from unique import Unique

```

```

path = "../files/data_light.json"
path_to_write = "../files/tmp_res.json"

```

```

with open(path) as f:
    data = json.load(f)

```

```

def f1(arg):
    return sort(Unique([i["job-name"] for i in arg], ignore_case =
True).__next__())

```

```

def f2(arg):
    return list(filter(lambda a: "программист" == a.split()[0].strip(), arg))

```

```

def f3(arg):
    return [item + " с опытом Python" for item in arg]

```

```
@print_result
```

```

def f4(arg):
    sals = [gen_random(1, 100000, 200000) for _ in range(len(arg))]
    res_list = list(zip(arg, sals))
    return [i[0] + ", с зарплатой " + str(i[1]) + " руб." for i in res_list]

```

```
if __name__ == "__main__":  
    with Cm_timer_1():  
        f4(f3(f2(f1(data))))
```

Пример выполнения программы

```
/usr/local/bin/python3.10 /Users/mikhail/Documents/BKIT/Lab1/lab_python_fp/process_data.py  
программист с опытом Python, с зарплатой 168288 руб.  
программист / senior developer с опытом Python, с зарплатой 186291 руб.  
программист 1с с опытом Python, с зарплатой 111767 руб.  
программист с# с опытом Python, с зарплатой 164596 руб.  
программист с++ с опытом Python, с зарплатой 105561 руб.  
программист с++/с#/java с опытом Python, с зарплатой 104163 руб.  
time: 0.1391751766204834
```

```
Process finished with exit code 0
```