

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Система обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Лабораторная работа №3-4  
«Функциональные возможности языка Python»

Выполнил:  
студент группы ИУ5-32Б:  
Бокатуев М. С.

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.

Москва, 2023 г.

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.

Код программы:

Файл cm\_timer.py

```
import time
from contextlib import contextmanager

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

    def __exit__(self, exc_type, exc_val, exc_tb):
        elapsed_time = time.time() - self.start_time
        print(f'time: {elapsed_time} seconds')

@contextmanager
def cm_timer_2():
    start = time.time()
    try:
        yield
    finally:
        end = time.time()
        print(f'time: {end - start} seconds')

def main():
    with cm_timer_1():
        time.sleep(5.5)

    with cm_timer_2():
        time.sleep(5.5)

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

Файл field.py

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

```

        for j in args:
            if not(j in i):
                continue
            d[j] = i[j]
        yield d
def main():
    goods = [
        {'title': 'Ковер', 'price': 2000, 'color': 'green'},
        {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
    ]
    for i in field(goods, 'title'):
        print(i)
    for i in field(goods, 'title', 'price'):
        print(i)

if __name__ == '__main__':
    main()

```

### Файл gen\_random.py

```

from random import randint
def gen_random(num_count, begin, end):
    for x in range(num_count):
        yield randint(begin, end)
    # Необходимо реализовать генератор
def main():
    for i in gen_random(5, 1, 3):
        print(i)
if __name__ == '__main__':
    main()

```

### Файл print\_result.py

```

def print_result(func):
    def inner(*args, **kwargs):
        print(func.__name__)
        if type(func(*args, **kwargs)) == list :
            print('\n'.join(map(str, func(*args, **kwargs))))
            return func(*args, **kwargs)
        elif type(func(*args, **kwargs)) == dict :
            for i, j in func(*args, **kwargs).items():
                print(i, j, sep = ' = ')
            return func(*args, **kwargs)
        else :
            print(func(*args, **kwargs))
            return func(*args, **kwargs)
    return inner

# Здесь должна быть реализация декоратора

@print_result
def test_1():

```

```

        return 1

@print_result
def test_2():
    return 'iu5'

@print_result
def test_3():
    return {'a': 1, 'b': 2}

@print_result
def test_4():
    return [1, 2]

if __name__ == '__main__':
    print('!!!!!!!')
    test_1()
    test_2()
    test_3()
    test_4()

```

Файл process\_data.py

```

import json
import sys
from cm_timer import cm_timer_1
from print_result import print_result
from unique import Unique
from gen_random import gen_random

path = "lab_python_fp/data_light.json"

with open(path, encoding='utf-8') as f:
    data = json.load(f)

@print_result
def f1(arg):
    return [i for i in Unique([i["job-name"].lower() for i in arg])]

@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):
    return ['{0}, зарплата {1} руб.'.format(i[0], i[1]) for i in list(zip(arg,
gen_random(len(arg), 100000, 200000)))]

if __name__ == '__main__':
    with cm_timer_1():
        f4(f3(f2(f1(data))))

```

Файл sort.py

```

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

if __name__ == '__main__':
    result = sorted(data, reverse = True, key = abs)
    print(result)

    result_with_lambda = sorted(data, reverse = True, key= lambda x: abs(x))
    print(result_with_lambda)

```

Файл unique.py

```

from gen_random import gen_random
class Unique(object):
    def __init__(self, items, **kwargs):
        self.ignore_case = kwargs.get('ignore_case', False)
        if self.ignore_case:
            self.items = iter(set(''.join(items).lower()))
        else:
            self.items = iter(set(items))
    def __next__(self):
        return next(self.items)

    def __iter__(self):
        return self

def main():
    data = gen_random(10, 1, 3)
    a = Unique(data)
    for i in a:
        print(i)

if __name__ == '__main__':
    main()

```

Результат выполнения программы:

Файл cm\_timer.py

```

time: 5.500487565994263 seconds
time: 5.500454664230347 seconds

```

### Файл field.py

```
Ковер
Диван для отдыха
{'title': 'Ковер', 'price': 2000}
{'title': 'Диван для отдыха', 'price': 5300}
```

### Файл gen\_random.py

```
3
1
2
1
3
```

### Файл print\_result.py

```
!!!!!!!
test_1
1
test_2
iu5
test_3
a = 1
b = 2
test_4
1
2
```

### Файл process\_data.py

```
программист-разработчик информационных систем с опытом Python
программист с++ с опытом Python
f4
программист с опытом Python, зарплата 115601 руб.
программист/ технический специалист с опытом Python, зарплата 133526 руб.
программист 1с с опытом Python, зарплата 133338 руб.
программист с++/с#/java с опытом Python, зарплата 144278 руб.
программист с# с опытом Python, зарплата 141840 руб.
программист / senior developer с опытом Python, зарплата 109498 руб.
программист/ junior developer с опытом Python, зарплата 185095 руб.
программист-разработчик информационных систем с опытом Python, зарплата 190737 руб.
программист с++ с опытом Python, зарплата 135832 руб.
time: 0.16776394844055176 seconds
```

### Файл sort.py

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

### Файл unique.py

```
on_fp/unique.py
1
2
3
```