Отчёт по лабораторной работе №3
по курсу «Разработка Интернет-приложений»

Тема: «Python. Функциональные возможности»

Выполнил:
студент группы РТ5-51
Наврузов Эмир

Дата: _____ Подпись: _____

Проверил:

Дата: _____ Подпись: _____

Москва, 2018

Задание

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 запрограммировать одной строкой.

Исходный код

gen.py

```python
import random

goods = [
    {'title': 'Ковер', 'price': 2000, 'color': 'green'},
    {'title': 'Диван для отдыха', 'color': 'black'}
]


def field(l, *args):
    # в списке смотрим каждый словарь
    for el in l:
        # условие на количество передаваемых args
        if len(args) > 1:
            d = {}
            # идем по ключам, которые даны
            for arg in args:
                # если в словаре есть такой ключ, то вписываем его
                # во временный словарь, который потом возвращаем
                if arg in el.keys():
                    if el[arg] is not None:
                        d[arg] = el[arg]
            if d:
                yield d
        elif len(args) == 1:
            for arg in args:
                if arg in el.keys():
                    if el[arg] is not None:
                        yield el[arg]


def gen_random(min_r, max_r, amount):
    for i in range(amount):
        # просто рандомим из указанного интервала
        # к max_r добавляем единицу, потому что
        # без единицы интервал справа будет на 1 меньше, чем надо
        yield (random.randrange(min_r, max_r + 1))


if __name__ == "__main__":
    # print("field:", [i for i in field(goods, 'title', 'price')])
    print("field:", [i for i in field(goods, 'title')])
    print("gen_random result:", [i for i in gen_random(2, 7, 10)])
```

Результат:

```
C:\Python34\python.exe C:/Users/student/emir/python_labs/3/gen.py
field: ['Ковер', 'Диван для отдыха']
gen_random result: [2, 3, 7, 6, 4, 4, 2, 6, 5, 7]

Process finished with exit code 0
```

iterators.py

```python
class Unique:
    def __init__(self, l, ignore_case=False):
        self.data = self.get_unique_list(data, ignore_case)
        self.n = len(self.data)
        self.i = 0

    def __iter__(self):
        return self

    # @staticmethod
    # def get_unique_list(l, ignore_case):
    #     if ignore_case:
    #         return list(set([i.lower() for i in l]))
    #     else:
    #         return list(set(l))

    @staticmethod
    def get_unique_list(l, ignore_case):
        unique_list = []

        if ignore_case:
            l = [i.lower() for i in l]

        for el in l:
            if el not in unique_list:
                unique_list.append(el)
        return unique_list

    def __next__(self):
        if self.i < self.n:
            ret = self.data[self.i]
            self.i += 1
            return ret
        else:
            raise StopIteration()


if __name__ == "__main__":
    # data = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2]
    data = ['a', 'A', 'b', 'B']

    x = Unique(data, ignore_case=True)
    print([i for i in x])
```

Результат:

```
C:\Python34\python.exe C:/Users/student/emir/python_labs/3/iterators.py
[1, 2]
['a', 'b']

Process finished with exit code 0
```

ex3.py

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

print(sorted(data, key=abs))
```

Результат:

```
C:\Python34\python.exe C:/Users/student/emir/python_labs/3/ex3.py
[0, 1, -1, 4, -4, -30, 100, -100, 123]

Process finished with exit code 0
```

decorators.py

```
def print_result(func):
    def printer():
        print(func.__name__)
        res_of_func = func()
        # print(res_of_func)

        if isinstance(res_of_func, list):
            print('\n'.join([str(i) for i in res_of_func]))
        elif isinstance(res_of_func, dict):
            print('\n'.join(["{} = {}".format(key, value) for key, value in
res_of_func.items()]))

    return printer


@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:

```
C:\Python34\python.exe C:/Users/student/emir/python_labs/3/decorators.py
test_1
test_2
test_3
a = 1
b = 2
test_4
1
2

Process finished with exit code 0
```

ex5.py

```
import time


class Foo:
    def __init__(self):
        self.script_time = 0

    def __enter__(self):
        self.script_time = time.time()

    def __exit__(self, exc_type, exc_val, exc_tb):
        print(time.time() - self.script_time)
        self.script_time = 0


with Foo():
    time.sleep(5.5)
```

Результат:

```
C:\Python34\python.exe C:/Users/student/emir/python_labs/3/ex5.py
5.500314950942993

Process finished with exit code 0
```

ex6.py

```
import json
import random
import time


class Foo:
    def __init__(self):
        self.script_time = 0

    def __enter__(self):
        self.script_time = time.time()

    def __exit__(self, exc_type, exc_val, exc_tb):
        print(time.time() - self.script_time)
        self.script_time = 0
```

```
def print_result(func):
    def printer(lst):
        print(func.__name__)
        res_of_func = func(lst)

        if isinstance(res_of_func, list):
            print('\n'.join([str(i) for i in res_of_func]))
        return res_of_func

    return printer


@print_result
def f1(*args):
    return sorted(list(set([data[prof_el]["job-name"].lower() for prof_el in
range(len(data))])))


@print_result
def f2(profs):
    return list(filter(lambda x: x.startswith("программист"), profs))


@print_result
def f3(profs):
    return [i + " с опытом Python" for i in profs]
    # return ' с опытом Python'.join(list(map(str, profs))) #с использованием
map


@print_result
def f4(profs):
    l_of_salaries = [random.randrange(100000, 200000) for i in
range(len(profs))]
    return ["{}, зарплата {}".format(x, y) for x, y in zip(profs,
l_of_salaries)]


if __name__ == "__main__":
    with Foo():
        with open("data_light.json", encoding="utf-8") as file:
            data = json.load(file)
        f4(f3(f2(f1([1, 2, 3]))))
```

Результат:

```
программист с опытом Python, зарплата 130689
программист / senior developer с опытом Python, зарплата 193443
программист 1с с опытом Python, зарплата 180865
программист c# с опытом Python, зарплата 157460
программист c++ с опытом Python, зарплата 139789
программист c++/c#/java с опытом Python, зарплата 145073
программист/ junior developer с опытом Python, зарплата 114921
программист/ технический специалист с опытом Python, зарплата 130986
программистр-разработчик  информационных систем с опытом Python, зарплата 198136
0.6450369358062744
```