

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

**Факультет «Информатика и вычислительная техника»  
Кафедра ИУ5 «Система обработки информации и управления»**

**Отчет по лабораторной работе №5  
«Модульное тестирование в Python»**

**Выполнил  
студент ИУ5-35Б  
Рябова С.А.**

**Проверил  
Преподаватель каф. ИУ5  
Гапанюк Ю.Е.**

**Москва 2024**

## Описание задания

1. Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.
2. Модифицируйте код таким образом, чтобы он был пригоден для модульного тестирования.
3. Разработайте модульные тесты. В модульных тестах необходимо применить следующие технологии:
  - TDD - фреймворк (не менее 3 тестов).
  - BDD - фреймворк (не менее 3 тестов).
  - Создание Mock-объектов (необязательное дополнительное задание).

## Текст программ

### **main.py**

```
import json
import time
import sys
import random
```

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

```
def print_result(func):
    def inner1(*args, **kwargs):
        result = func(*args, **kwargs)
        print(func.__name__)
        if type(result) == list:
            for i in result:
                print(i)
        elif type(result) == dict:
            for key, val in zip(result.keys(), result.values()):
                print(key, '=', val)
        else:
            print(result)
    return result
```

```
return inner1
```

```
class cm_timer_1():
    def __init__(self):
        self.begin = None
        self.end = None

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

    def __exit__(self, exc_type, exc_value, exc_traceback):
        self.end = time.time()
        print('time: %.3f % (self.end - self.begin))
```

```
def field(items, *args):
    assert len(args) > 0
    for item in items:
        if len(args) == 1:
            value = item.get(args[0])
            if value is not None:
                yield value
        else:
            result = {}
            for arg in args:
                value = item.get(arg)
                if value is not None:
                    result[arg] = value

            if result:
                yield result
```

```
class Unique(object):
    def __init__(self, items, **kwargs):
        self.it = iter(items)
        self.cur = next(self.it)
        self.prev = ""
        if len(kwargs) > 0:
            self.cs = kwargs['ignore_case']
        else:
            self.cs = False

    def __next__(self):
```

```

if type(self.cur) == str and self.cs:
    while self.cur.lower() == self.prev.lower():
        self.cur = next(self.it)
    self.prev = self.cur
    return self.cur
else:
    while self.cur == self.prev:
        self.cur = next(self.it)
    self.prev = self.cur
    return self.cur

```

```

def __iter__(self):
    return self

```

```

path = "/Users/sonyaryabova/PycharmProjects/pythonProject2/data_light.json"

```

```

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

```

```

@print_result
def f1(data):
    return list(Unique(sorted(list((field(data, 'job-name')))), ignore_case=False))

```

```

@print_result
def f2(seq):
    return list(filter(lambda st: len(st) >= 11 and st[:11].lower() == 'программист',
seq))

```

```

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

```

```

@print_result
def f4(seq):
    return [x + ' ' + str(y) for x, y in zip(seq, list(gen_random(len(seq), 100000,
200000)))]

```

```

if __name__ == '__main__':
    with cm_timer_1():

```

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

### **test\_func1.py**

```
import json
from Project.main import f1

def test_sorted():
    path =
"/Users/sonyaryabova/PycharmProjects/pythonProject2/Tests/data_test.json"

    with open(path, encoding='utf-8') as file:
        data = json.load(file)
        l = f1(data)
        assert l == sorted(l)
```

### **test\_func2.py**

```
import json
from Project.main import f1, f2, f3

def test_ends_with_python():
    path =
"/Users/sonyaryabova/PycharmProjects/pythonProject2/Tests/data_test.json"

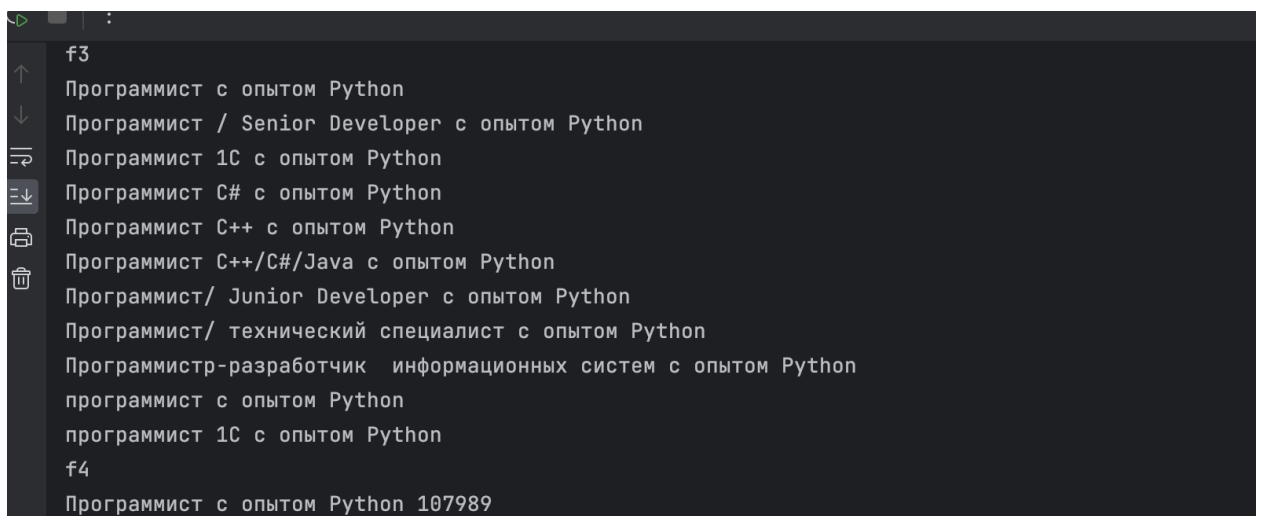
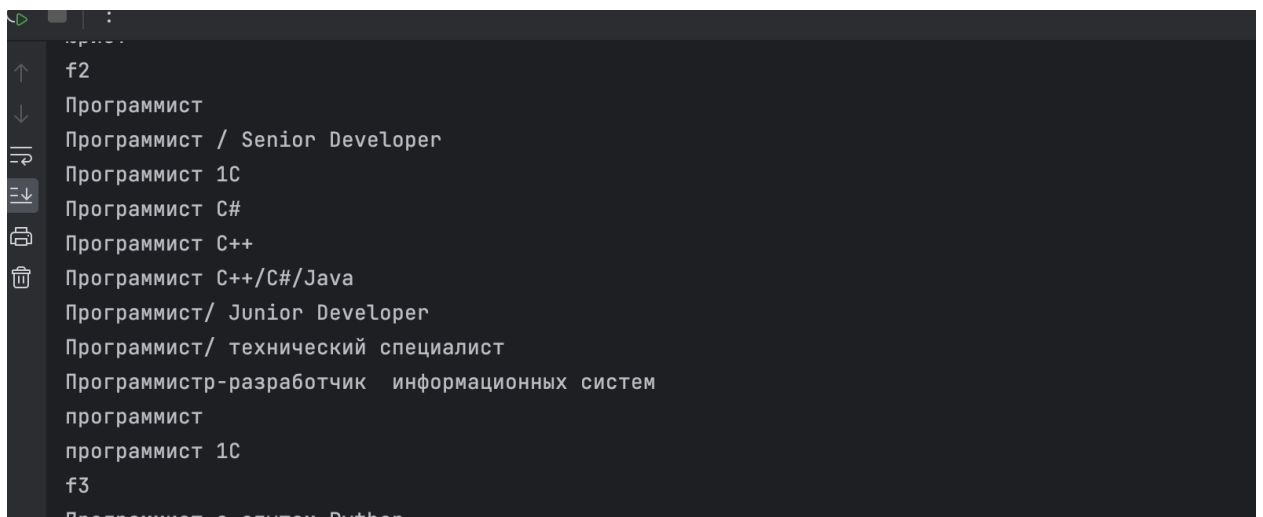
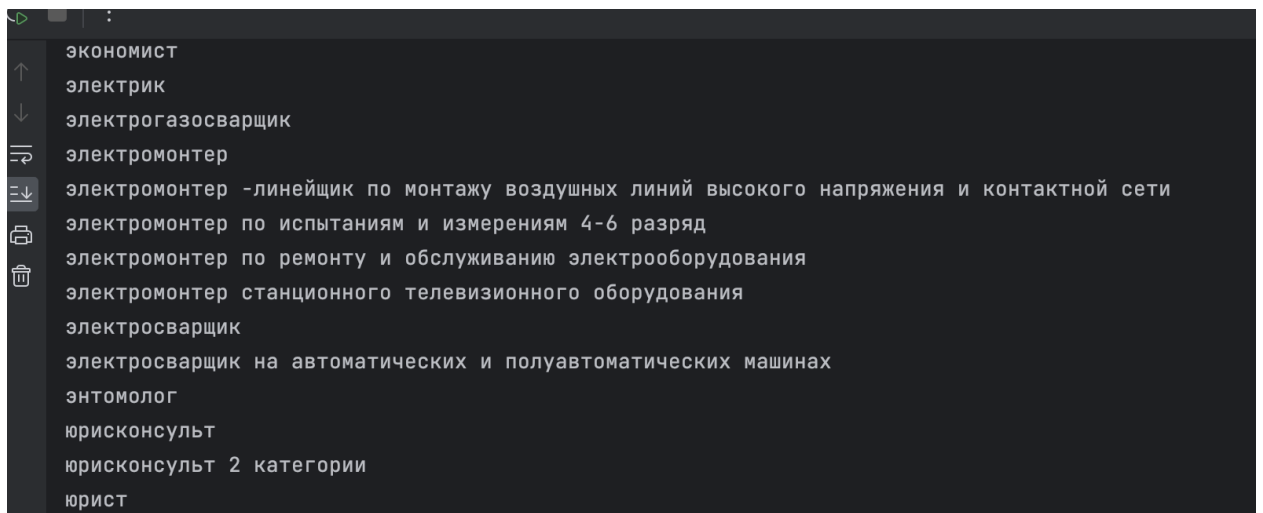
    with open(path, encoding='utf-8') as file:
        data = json.load(file)
        l = f3(f2(f1(data)))
        for i in l:
            assert i[-16:] == ' с опытом Python'
```

### **test\_unique.py**

```
from Project.main import Unique

def test_Unique():
    prev = 0
    l = [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3]
    for i in Unique(l):
        assert prev != i
        prev = i
```

## Экранные формы выполнения программ



программист 1С с опытом Python

f4

Программист с опытом Python 107989

Программист / Senior Developer с опытом Python 117000

Программист 1С с опытом Python 194370

Программист C# с опытом Python 144472

Программист C++ с опытом Python 136437

Программист C++/C#/Java с опытом Python 108440

Программист/ Junior Developer с опытом Python 174324

Программист/ технический специалист с опытом Python 125319

Программист-разработчик информационных систем с опытом Python 127917

программист с опытом Python 141776

программист 1С с опытом Python 189648

time: 0.020