ОТЧЕТ

По лр-3-4

Дисциплина «Парадигмы и конструкции языков программирования»

Студент: Коваленко Е.

Группа: ИБМ3-34Б

Задание:

Задание лабораторной работы состоит из решения нескольких задач.

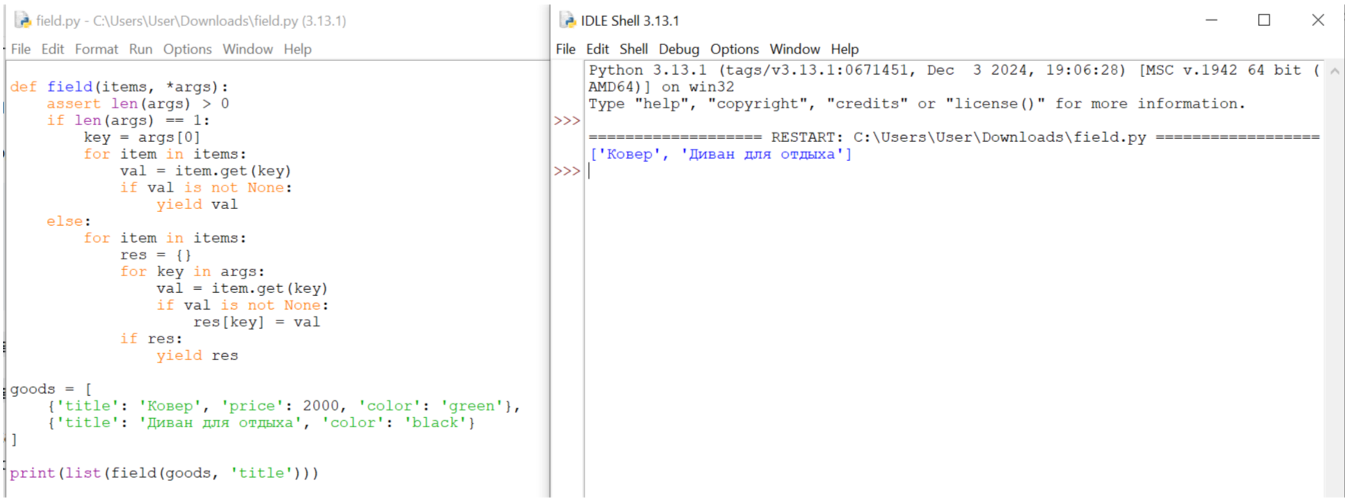
Файлы, содержащие решения отдельных задач, должны располагаться в пакете lab_python_fp. Решение каждой задачи должно располагаться в отдельном файле.

При запуске каждого файла выдаются тестовые результаты выполнения соответствующего задания.

Задача 1

Текст программы:

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

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

print(list(field(goods, 'title')))
```



field.py - C:\Users\User\Downloads\field.py (3.13.1)
File Edit Format Run Options Window Help
IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help

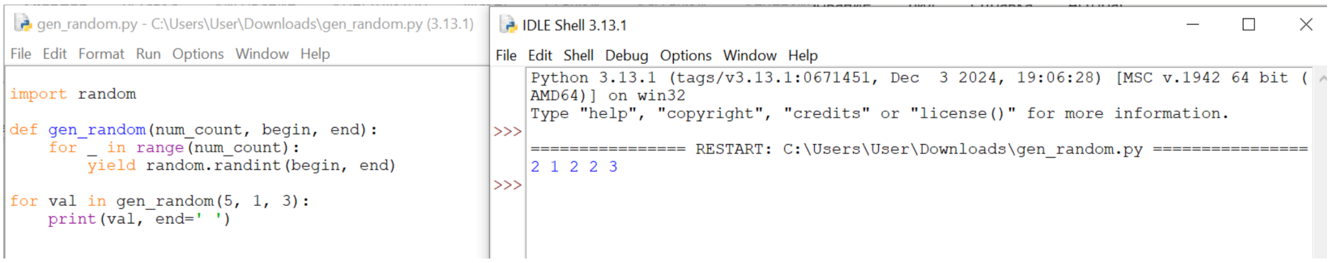
```
def field(items, *args):
    assert len(args) > 0
    if len(args) == 1:
        key = args[0]
        for item in items:
            val = item.get(key)
            if val is not None:
                yield val
    else:
        for item in items:
            res = {}
            for key in args: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

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

print(list(field(goods, 'title')))
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=================== RESTART: C:\Users\User\Downloads\field.py ==================
['Ковер', 'Диван для отдыха']
>>>
```

Задача 2

Текст программы:

```
import random


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


for val in gen_random(5, 1, 3):
    print(val, end=' ')
```


gen_random.py - C:\Users\User\Downloads\gen_random.py (3.13.1)
File Edit Format Run Options Window Help
IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help

```
import random

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

for val in gen_random(5, 1, 3):
    print(val, end=' ')
```

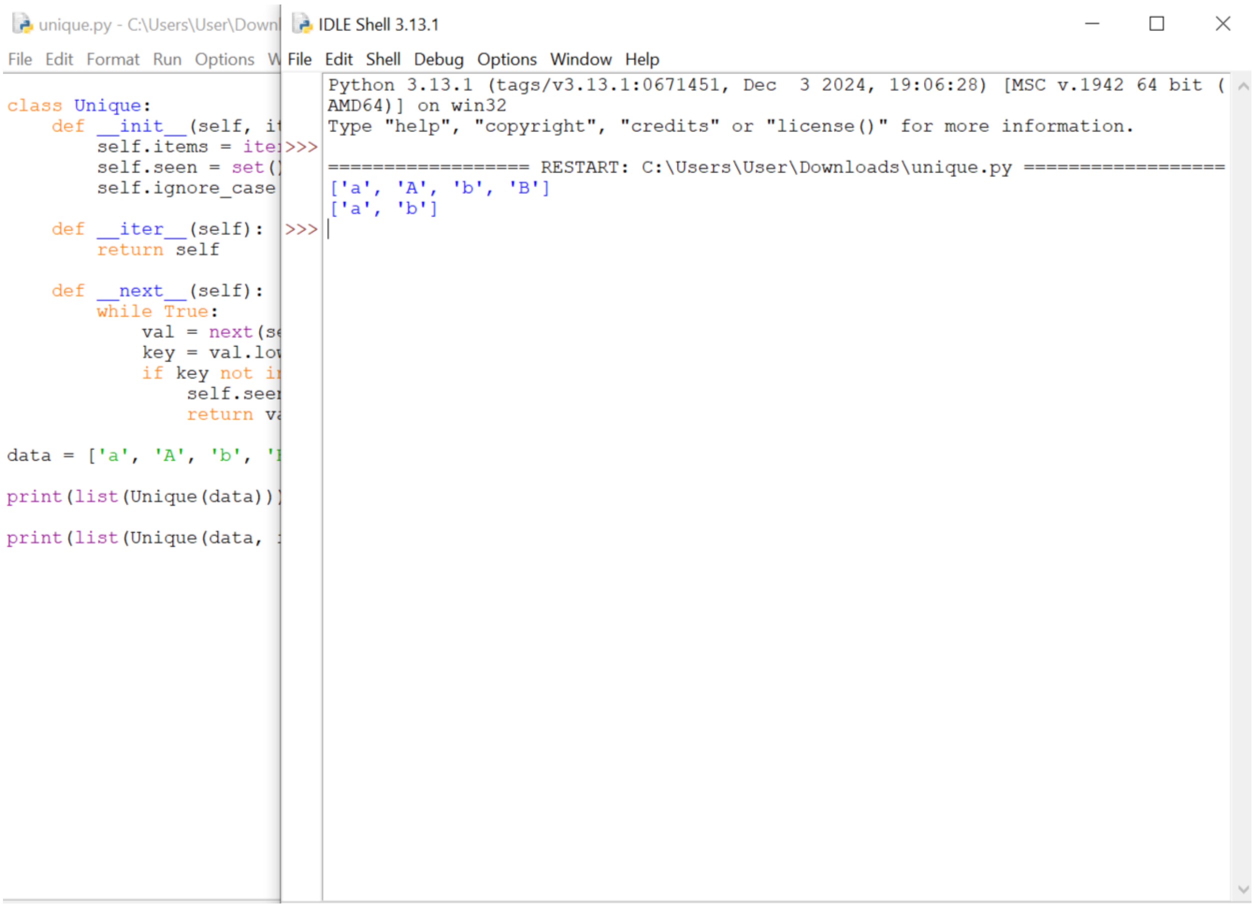
```
    Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
    AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    ================ RESTART: C:\Users\User\Downloads\gen_random.py ================
    2 1 2 2 3
>>>
```

Задача 3

Текст программы:

```
class Unique:
    def __init__(self, items, **kwargs):
        self.items = iter(items)
        self.seen = set()
        self.ignore_case = kwargs.get('ignore_case', False)


    def __iter__(self):
        return self


    def __next__(self):
        while True:
            val = next(self.items)
            key = val.lower() if (self.ignore_case and isinstance(val, str)) else val
```

```
      if key not in self.seen:
        self.seen.add(key)
        return val

data = ['a', 'A', 'b', 'B', 'a', 'A', 'b', 'B']

print(list(Unique(data)))

print(list(Unique(data, ignore_case=True)))
```



unique.py - C:\Users\User\Downl
File Edit Format Run Options W
IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help

```
class Unique:
    def __init__(self, i
        self.items = ite
        self.seen = set(
        self.ignore_case

    def __iter__(self):
        return self

    def __next__(self):
        while True:
            val = next(se
            key = val.low
            if key not i
                self.seen
                return va

data = ['a', 'A', 'b', 'I

print(list(Unique(data))

print(list(Unique(data,
```

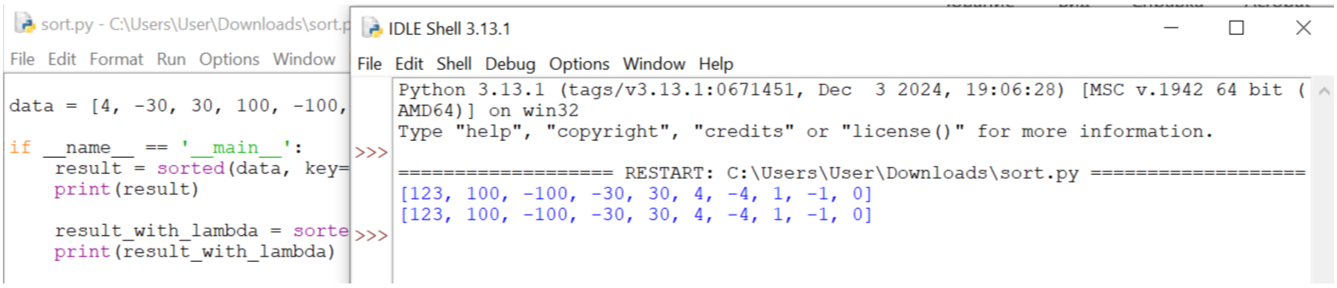
```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:\Users\User\Downloads\unique.py ==================
['a', 'A', 'b', 'B']
['a', 'b']
>>> |
```

Задача 4

Текст программы:

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

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

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

    print(result_with_lambda)
```



sort.py - C:\Users\User\Downloads\sort.p
File Edit Format Run Options Window
IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help

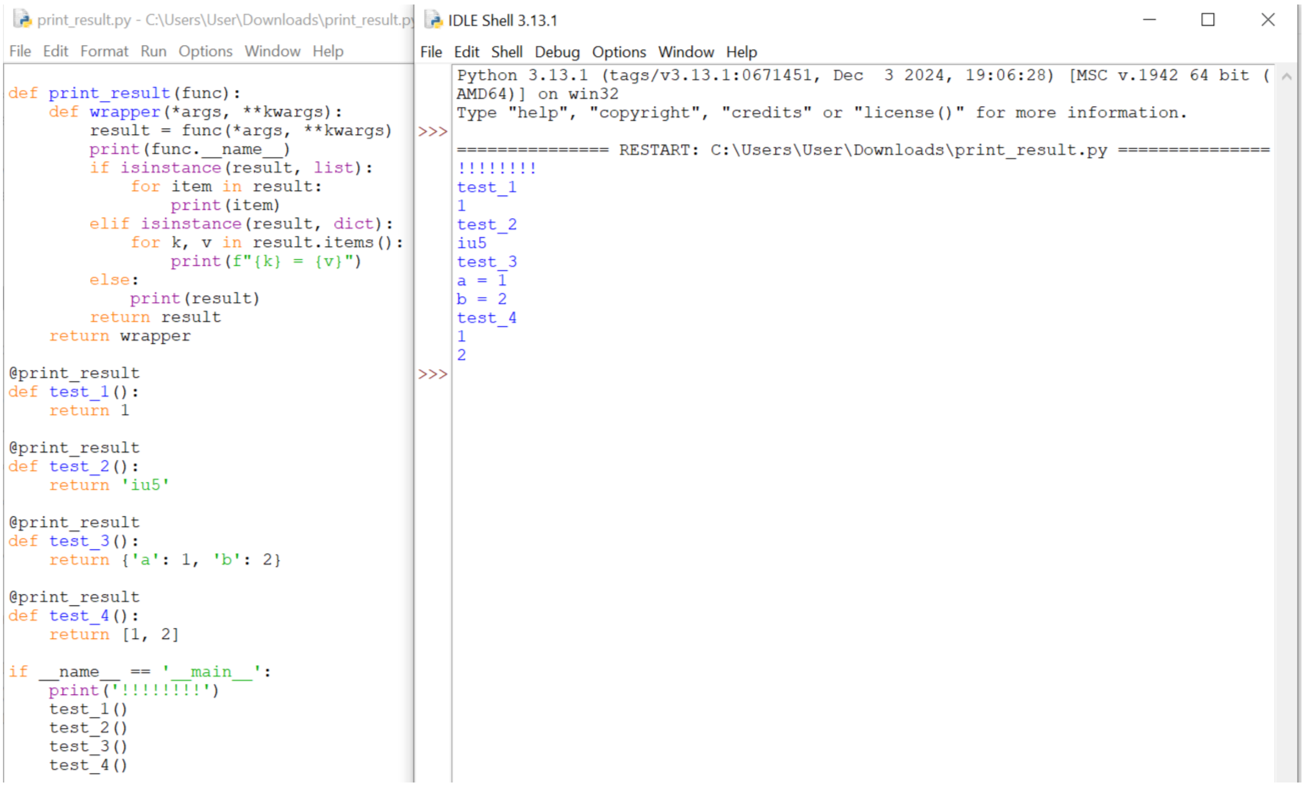
```
data = [4, -30, 30, 100, -100,

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

    result_with_lambda = sorte
    print(result_with_lambda)
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=================== RESTART: C:\Users\User\Downloads\sort.py ===================
[123, 100, -100, -30, 30, 4, -4, 1, -1, 0]
[123, 100, -100, -30, 30, 4, -4, 1, -1, 0]
>>>
```

Задача 5

Текст программы:

```
def print_result(func):

    def wrapper(*args, **kwargs):

        result = func(*args, **kwargs)

        print(func.__name__)

        if isinstance(result, list):

            for item in result:

                print(item)

        elif isinstance(result, dict):

            for k, v in result.items():

                print(f"{k} = {v}")

        else:

            print(result)

        return result

    return wrapper



@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()
```


print_result.py - C:\Users\User\Downloads\print_result.py
File Edit Format Run Options Window Help
IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help

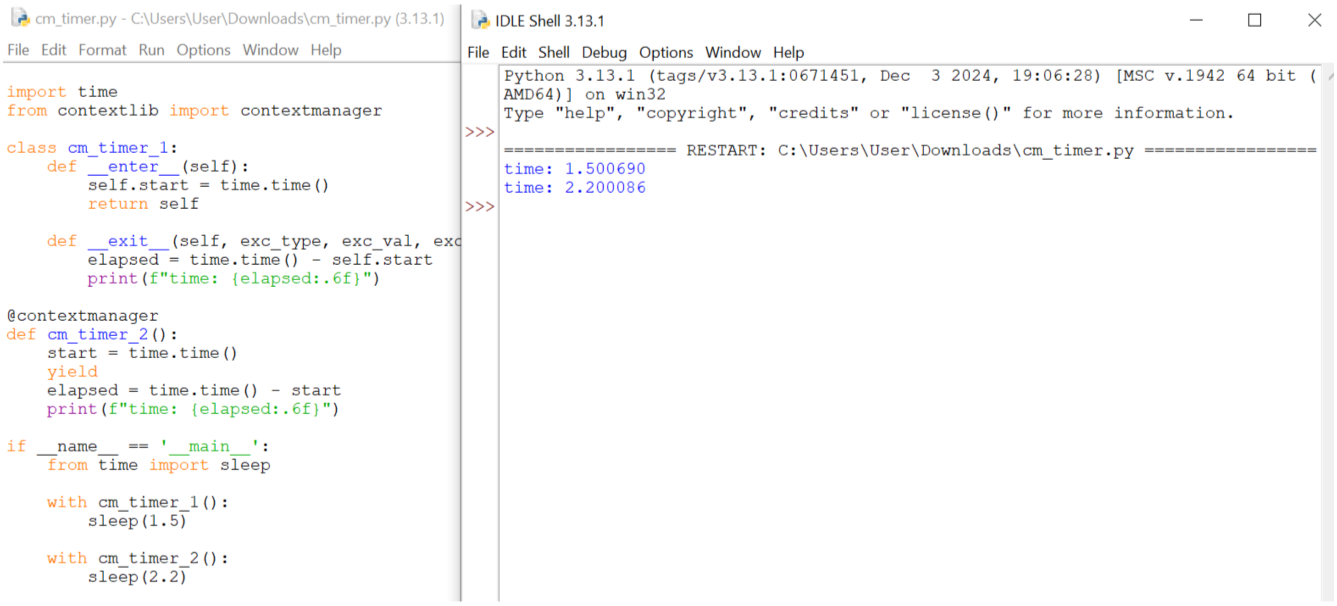
```
def print_result(func):
    def wrapper(*args, **kwargs):
        result = func(*args, **kwargs)
        print(func.__name__)
        if isinstance(result, list):
            for item in result:
                print(item)
        elif isinstance(result, dict):
            for k, v in result.items():
                print(f"{k} = {v}")
        else:
            print(result)
        return result
    return wrapper

@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()
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\User\Downloads\print_result.py ===============
!!!!!!!!
test_1
1
test_2
iu5
test_3
a = 1
b = 2
test_4
1
2
>>>
```

Задача 6

Текст программы:

```
import time
from contextlib import contextmanager


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

    def __exit__(self, exc_type, exc_val, exc_tb):
        elapsed = time.time() - self.start
        print(f"time: {elapsed:.6f}")


@contextmanager
def cm_timer_2():
    start = time.time()
    yield
    elapsed = time.time() - start
    print(f"time: {elapsed:.6f}")


if __name__ == '__main__':
    from time import sleep

    with cm_timer_1():
        sleep(1.5)

    with cm_timer_2():
        sleep(2.2)
```


cm_timer.py - C:\Users\User\Downloads\cm_timer.py (3.13.1)
File Edit Format Run Options Window Help
IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help

```
import time
from contextlib import contextmanager

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

    def __exit__(self, exc_type, exc_val, exc
        elapsed = time.time() - self.start
        print(f"time: {elapsed:.6f}")

@contextmanager
def cm_timer_2():
    start = time.time()
    yield
    elapsed = time.time() - start
    print(f"time: {elapsed:.6f}")

if __name__ == '__main__':
    from time import sleep

    with cm_timer_1():
        sleep(1.5)

    with cm_timer_2():
        sleep(2.2)
```

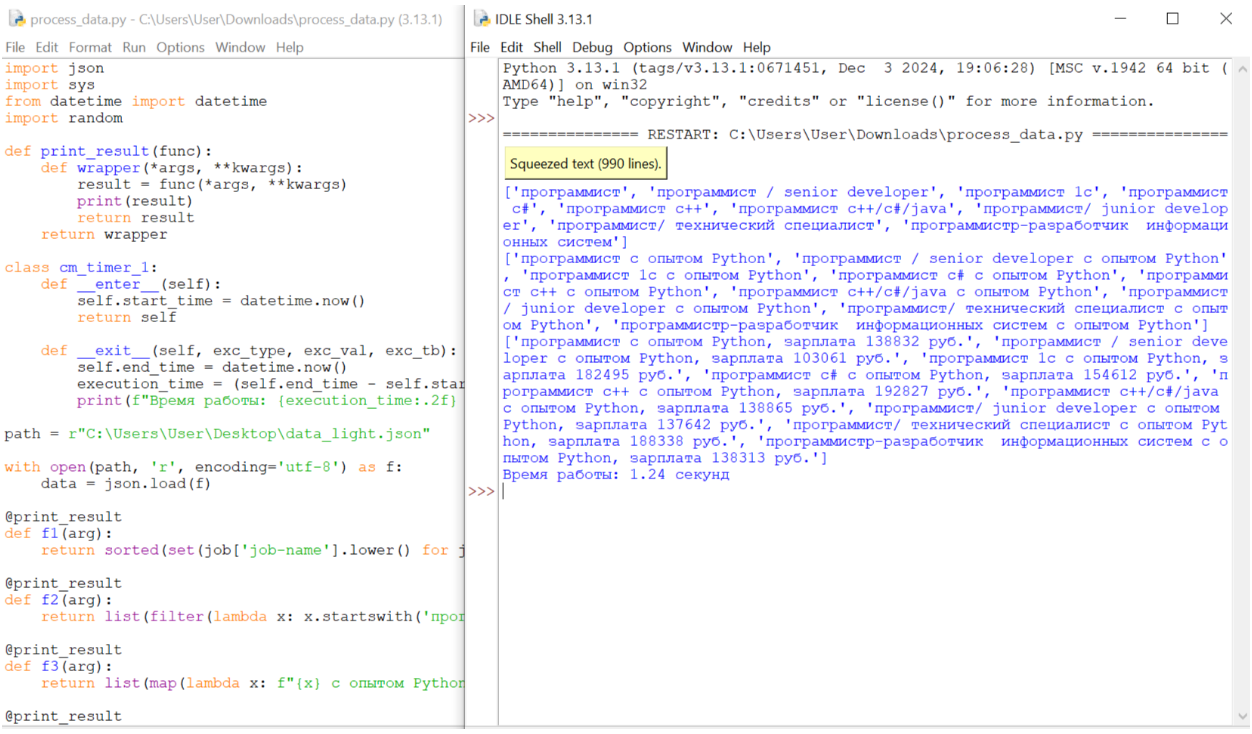
```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: C:\Users\User\Downloads\cm_timer.py =================
time: 1.500690
time: 2.200086
>>>
```

Задача 7

Текст программы:

```
import json

import sys

from datetime import datetime

import random


def print_result(func):

   def wrapper(*args, **kwargs):

       result = func(*args, **kwargs)

       print(result)

       return result

   return wrapper


class cm_timer_1:

   def __enter__(self):

       self.start_time = datetime.now()

       return self


   def __exit__(self, exc_type, exc_val, exc_tb):

       self.end_time = datetime.now()

       execution_time = (self.end_time - self.start_time).total_seconds()
```

```
        print(f"Время работы: {execution_time:.2f} секунд")

path = r"C:\Users\User\Desktop\data_light.json"

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

@print_result
def f1(arg):
    return sorted(set(job['job-name'].lower() for job in arg), key=str.lower)

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

@print_result
def f3(arg):
    return list(map(lambda x: f"{x} с опытом Python", arg))

@print_result
def f4(arg):
    salaries = [random.randint(100000, 200000) for _ in arg]
    return [f"{job}, зарплата {salary} руб." for job, salary in zip(arg, salaries)]

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


process_data.py - C:\Users\User\Downloads\process_data.py (3.13.1)
File Edit Format Run Options Window Help
IDLE Shell 3.13.1
File Edit Shell Debug Options Window Help
Squeezed text (990 lines).

```python
import json
import sys
from datetime import datetime
import random

def print_result(func):
    def wrapper(*args, **kwargs):
        result = func(*args, **kwargs)
        print(result)
        return result
    return wrapper

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

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.end_time = datetime.now()
        execution_time = (self.end_time - self.star
        print(f"Время работы: {execution_time:.2f}

path = r"C:\Users\User\Desktop\data_light.json"

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

@print_result
def f1(arg):
    return sorted(set(job['job-name'].lower() for j

@print_result
def f2(arg):
    return list(filter(lambda x: x.startswith('прог

@print_result
def f3(arg):
    return list(map(lambda x: f"{x} с опытом Python

@print_result
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
=============== RESTART: C:\Users\User\Downloads\process_data.py ===============
Squeezed text (990 lines).
['программист', 'программист / senior developer', 'программист 1с', 'программист
 с#', 'программист с++', 'программист с++/с#/java', 'программист/ junior develop
er', 'программист/ технический специалист', 'программистр-разработчик  информаци
онных систем']
['программист с опытом Python', 'программист / senior developer с опытом Python'
, 'программист 1с с опытом Python', 'программист с# с опытом Python', 'программи
ст с++ с опытом Python', 'программист с++/с#/java с опытом Python', 'программист
/ junior developer с опытом Python', 'программист/ технический специалист с опыт
ом Python', 'программистр-разработчик  информационных систем с опытом Python']
['программист с опытом Python, зарплата 138832 руб.', 'программист / senior deve
loper с опытом Python, зарплата 103061 руб.', 'программист 1с с опытом Python, з
арплата 182495 руб.', 'программист с# с опытом Python, зарплата 154612 руб.', 'п
рограммист с++ с опытом Python, зарплата 192827 руб.', 'программист с++/с#/java
с опытом Python, зарплата 138865 руб.', 'программист/ junior developer с опытом
Python, зарплата 137642 руб.', 'программист/ технический специалист с опытом Pyt
hon, зарплата 188338 руб.', 'программистр-разработчик  информационных систем с о
пытом Python, зарплата 138313 руб.']
Время работы: 1.24 секунд
>>> 
```