|  |  |  |
| --- | --- | --- |
| Защищено:  Нардид А.Н.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |  | Демонстрация:  Нардид А.Н.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

**Отчет по лабораторной работе № 3-4 по курсу**

**Парадигмы и конструкции языков программирования**

**ГУИМЦ**

#### Тема работы: " Функциональные возможности языка Python. "

2

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-53Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Рабинович А. Д. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Москва, МГТУ - 2024 г.

# Тема и задание для выполнения лабораторной работы.

Тема работы: "изучение возможностей функционального программирования в языке Python.". Задание лабораторной работы состоит из решения нескольких задач. Файлы, содержащие решения отдельных задач, должны располагаться в пакете lab\_python\_fp. Решение каждой задачи должно раполагаться в отдельном файле. При запуске каждого файла выдаются тестовые результаты выполнения соответствующего задания.

# Листинг программы

**cm\_timer.py**

import contextlib

import time

import sys

class cm\_timer\_1(object):

def \_\_enter\_\_(self):

self.t = time.perf\_counter()

return self

def \_\_exit\_\_(self, type, value, traceback):

self.e = time.perf\_counter()

def \_\_float\_\_(self):

return float(self.e - self.t)

def \_\_str\_\_(self):

return f"{self.e - self.t:.4f}"

@contextlib.contextmanager

def cm\_timer\_2():

start\_time = time.perf\_counter()

yield

end\_time = time.perf\_counter()

print(f"{end\_time - start\_time:.4f}")

def main():

with cm\_timer\_1() as t1:

time.sleep(5.5)

print(t1)

with cm\_timer\_2() as t2:

time.sleep(5.5)

print(t2)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**field.py**

def field(items, \*args):

assert len(args) > 0

for item in items:

for arg in args:

if item[arg]:

print(item[arg], end = ' ')

print()

def main():

goods = [

{'title': 'Carpet', 'price': 2000, 'color': 'green'},

{'title': 'Sofa', 'price': 5300, 'color': 'black'}

]

field(goods, 'title')

field(goods, 'title', 'price')

if \_\_name\_\_ == "\_\_main\_\_":

main()

**gen\_random.py**

import random

def gen\_random(num\_count, begin, end):

for \_ in range(num\_count):

print(random.randrange(begin, end))

def main():

gen\_random(5, 1, 3)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**print\_result.py**

def print\_result(input\_func):

def output\_func():

print(input\_func.\_\_name\_\_)

result = input\_func()

if (isinstance(result, list)):

for elem in result:

print(elem)

elif (isinstance(result, dict)):

for key in result:

print(f"{key} = {result[key]}")

else:

print(result)

return output\_func

@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]

def main():

print('!!!!!!!!')

test\_1()

test\_2()

test\_3()

test\_4()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**process\_data.py**

import json

import sys

path = None

with open(path) as f:

data = json.load(f)

@print\_result

def f1(arg):

raise NotImplemented

@print\_result

def f2(arg):

raise NotImplemented

@print\_result

def f3(arg):

raise NotImplemented

@print\_result

def f4(arg):

raise NotImplemented

if \_\_name\_\_ == '\_\_main\_\_':

with cm\_timer\_1():

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

**sort.py**

def main():

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

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

print(result)

def sort\_by\_abs(x):

return abs(x)

result\_with\_lambda = sorted(data, key = sort\_by\_abs, reverse = True)

print(result\_with\_lambda)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**unique.py**

def main():

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

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

print(result)

def sort\_by\_abs(x):

return abs(x)

result\_with\_lambda = sorted(data, key = sort\_by\_abs, reverse = True)

print(result\_with\_lambda)

if \_\_name\_\_ == "\_\_main\_\_":

main()

# Результаты работы программы

**cm\_timer.py**

5.5002

5.5003

None

**field.py**

Carpet Sofa

Carpet 2000 Sofa 5300

**gen\_random.py**

2

2

2

2

1

**print\_result.py**

a = 1

b = 2

{'a': 1, 'b': 2}

1

2

!!!!!!!!

test\_1

test\_2

test\_3

test\_4

**sort.py**

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

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

**unique.py**

<\_\_main\_\_.Unique object at 0x000002A7962FB8C0>

<\_\_main\_\_.Unique object at 0x000002A7964AB390>