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

Факультет «Информатика и управление»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе № 3

«Функциональные возможности языка Python»

Выполнил: Проверил:

студент группы ИУ5-31Б преподаватель каф. ИУ5  
 Калинников Даниил Игоревич Гапанюк Юрий Евгеньевич

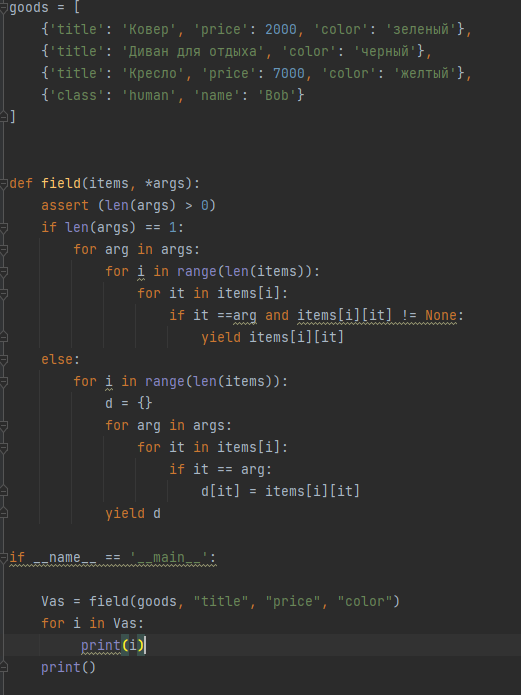
Подпись и дата: Подпись и дата:

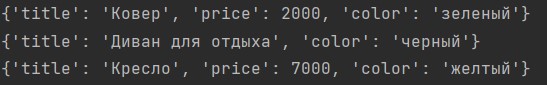
Москва, 2021 г.

Задание. Задание лабораторной работы состоит из решения нескольких задач.

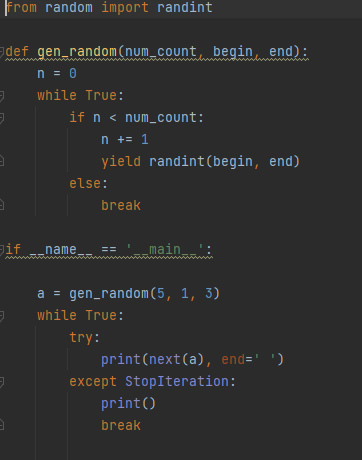
Файлы, содержащие решения отдельных задач, должны располагаться в пакете lab\_python\_fp. Решение каждой задачи должно располагаться в отдельном файле. При запуске каждого файла выдаются тестовые результаты выполнения соответствующего задания.

# Задача 1



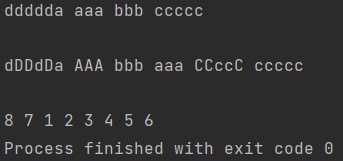


# Задача 2



Пример выполнения – 2 1 3 3 2

# Задача 3



# Задача 4

data = [5, -7, 10000, -1000, 1243, 551, 20, -41, -45]

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

result = sorted((data.copy()), key=abs, reverse=True)

print(result)

resultlambda = (lambda x: sorted((x), key=abs, reverse=True))(data.copy())

print(result\_with\_lambda)

Выполение – ( -1000 -45 -41 -7 5 20 551 1243 10000)

# Задача 5

def print\_result(funct):

def wrapper(\*args, \*\*kwargs):

print(funct.\_\_name\_\_)

res = funct(\*args, \*\*kwargs)

if isinstance(res, list):

for i in res:

print(i)

elif isinstance(res, dict):

for i in res:

print(i, '=', res[i])

else:

print(res)

return funct(\*args, \*\*kwargs)

return wrapper

# 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()

Выполенение программы -   
тест 1  
1

Тест 2

Iu5

Тест 3

А = 1

Б = 2

Тест 4

1

2

# Задача 6

import time

class cm\_timer\_1:

def \_\_init\_\_(self):

self.start\_time=time.time()

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

return 0

def \_\_exit\_\_(self, exp\_type, exp\_value, traceback):

if exp\_type is not None:

print(exp\_type, exp\_value, traceback)

else:

print('time=' + str(time.time() - self.start\_time))

from contextlib import contextmanager

@contextmanager

def cm\_timer\_2():

start\_time=time.time()

yield 0

print('time='+str(time.time()-start\_time))

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

with cm\_timer\_1():

time.sleep(1.5)

with cm\_timer\_2():

time.sleep(1.0)

time = 1.5012368514562

time = 1.125212546651

# Задача 7

import json

import sys

from print\_result import \*

from cm\_timer import \*

from unique import Unique as uniqum

from gen\_random import gen\_random as gRand

path = "data\_light.json"

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

data = json.load(f)

# Далее необходимо реализовать все функции по заданию, заменив `raise NotImplemented`

# Предполагается, что функции f1, f2, f3 будут реализованы в одну строку

# В реализации функции f4 может быть до 3 строк

def IT\_filter(el):

if el['job-name'][0:11].lower() == 'программист':

return True

else:

return False

@print\_result

def f1(arg):

return [u for u in uniqum([el.get('job-name') for el in arg], ignore\_case=True)]

@print\_result

def f2(arg):

return list(filter(lambda el: el[0:11].lower() == 'программист', arg))

@print\_result

def f3(arg):

return list(el+' с опытом Python' for el in arg)

@print\_result

def f4(arg):

zipped=list(zip(arg, gRand(len(arg), 100000, 200000)))

return [x+' с зарплатой '+str(y) for x, y in zipped]

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

with cm\_timer\_1():

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