|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Лабораторная работа № 3 по дисциплине «Разработка Интернет-Приложений»**

Студентка группы РТ5-51Б Бочарова М.А.

Вариант 2

Москва, 2020

1. **Цель лабораторной работы.**

Изучение объектно-ориентированных возможностей языка Python.

# Ход выполнения лабораторной работы.

**main.py**

from lab3.field import field

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

pass

**process\_data**

from lab3.field import field

from lab3.unique import Unique

from lab3.print\_result import print\_result

from lab3.gen\_random import gen\_random

from lab3.cm\_timer import cm\_timer\_1, cm\_timer\_2

import json

import sys

path = "C:\\Рета вуз\\пайтон\\lab\_python\_fp\\data\_light.json"

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

data = json.load(f)

@print\_result

def f1(arg):

return Unique([elem['job-name'] for elem in arg])

@ print\_result

def f2(arg):

return list(filter(lambda x: 'программист' in x, arg))

@ print\_result

def f3(arg):

return list(map(lambda x: x + " с опытом Python", arg))

@ print\_result

def f4(arg):

return list(map(lambda x: x + ", зарплата " + str(\*gen\_random(1, 100000, 200000)) + " руб", arg))

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

with cm\_timer\_1():

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

**cm\_timer**

from datetime import datetime

from time import sleep

import time

class cm\_timer\_1:

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

self.start = datetime.now()

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

self.result = datetime.now() - self.start

print(self.result)

class cm\_timer\_2:

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

self.\_start\_time = None

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

self.\_start\_time = time.perf\_counter()

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

elapsed\_time = time.perf\_counter() - self.\_start\_time

self.\_start\_time = None

print(elapsed\_time)

with cm\_timer\_1():

sleep(5.5)

with cm\_timer\_2():

sleep(2.5)

**field**

def field(dicts, \*args):

if len(args) > 1:

for dictionary in dicts:

for arg in args:

if arg not in dictionary.keys():

break

elif len(args) > 0:

for Key in dictionary.keys():

for a in args:

if Key == a:

yield dictionary.get(Key)

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

goods = [

{ "title": "Ковёр", "price": 1000, "color": "Зелёный" },

{ "title": "Диван", "price": 5300, "color": "Чёрный" },

{ "title": "Стул", "price": 500, "color": "Жёлтый" },

{ "title": "Стол", "price": 1200 },

{ "title": "Шкаф", "color": "Красный" },

{ "price": 2300, "color": "Бордовый" },

]

print("Один аргумент:")

rla = field(goods, 'title')

for i in rla:

print(i)

print("Наименование и цена:")

rla = field(goods, "title", "price")

for i in rla:

print(i)

print("Наименование и цвет:")

rla = field(goods, "title", "color")

for i in rla:

print(i)

print("Цены и цвет:")

rla = field(goods, "price", "color")

for i in rla:

print(i)