**Московский государственный технический**

**университет им. Н. Э. Баумана**

Факультет «Радиотехнический»

Кафедра РТ5 «Системы обработки информации и управления»

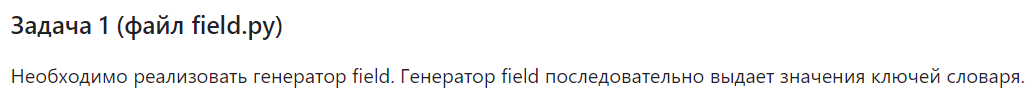
Курс «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №2

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы РТ5-31Б |  | преподаватель каф. ИУ5 |
| Суровец Е. А. |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2024 г.

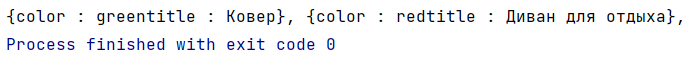
Задание:



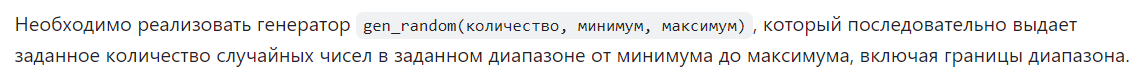
Код программы:

goods = [  
 {'title': 'Ковер', 'price': 2000, 'color': 'green'},  
 {'title': 'Диван для отдыха', 'price': 5300, 'color': 'red'}  
]  
  
def field(items, \*args):  
 assert len(args) > 0  
 for i in range(len(items)):  
 if len(args) == 1:  
 if args[0] in items[i]:  
 yield str(items[i][args[0]])  
 else:  
 print('{', end='')  
 for el in args:  
 if el in items[i]:  
 yield '{} : {}'.format(el, items[i][el])  
 print('}', end='')  
 print(', ', end='')  
  
for i in field(goods, 'color', 'title'):  
 print(i, end='')

Результат:



Задание:



Код:

from random import randint  
def get\_random(num\_count, begin, end):  
 for i in range(num\_count):  
 yield randint(begin, end)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 for i in get\_random(5, 1, 3):  
 print(i, end=' ')

Результат:



Задание:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Код:

class Unique(object):  
 def \_\_init\_\_(self, items, \*\*kwargs):  
 if 'ignore\_case' in kwargs:  
 self.items = list(set(map(str.lower, items[1:])))  
 else:  
 self.items = list(set(map(str, items)))  
 self.cur\_iter = 0  
  
 def \_\_iter\_\_(self):  
 return self  
  
 def \_\_next\_\_(self):  
 if self.cur\_iter <= len(self.items) - 1:  
 self.cur\_iter += 1  
 return self.items[self.cur\_iter - 1]  
 else:  
 raise StopIteration

data = ['a', 'A', 'b', 'B', 'a', 'A', 'b', 'B']  
print(' '.join(Unique(data, ignore\_case=True)))

Результат:

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Задание:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Код:

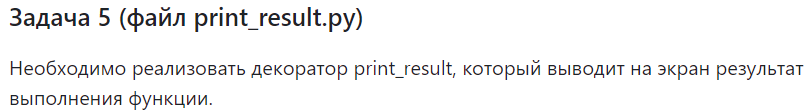
data = [4, -30, 100, -100, 123, 1, 0, -1, -4]  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 result = sorted(data, key=abs, reverse=True)  
 print(result)  
  
 result\_with\_lambda = sorted(data, key=lambda el: abs(el), reverse=True)  
 print(result\_with\_lambda)

Результат:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

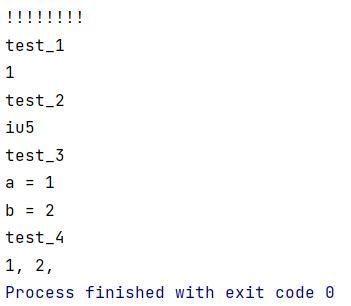
Автоматически созданное описание

Задание:

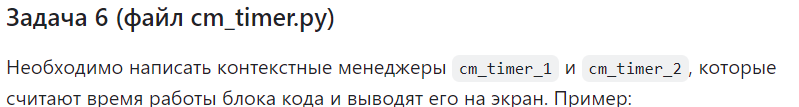


Код: def print\_result(func):  
 def wrapper():  
 print(func.\_\_name\_\_)  
 if type(func()) == list:  
 for el in func():  
 print(el, end=', ')  
 elif type(func()) == dict:  
 for key, val in func().items():  
 print('{} = {}'.format(key, val))  
 else:  
 print(func())  
 return wrapper  
  
  
@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()

Результат:



Задание:



Код:

from time import time, sleep  
from contextlib import contextmanager  
  
  
class cm\_timer\_1:  
 def \_\_enter\_\_(self):  
 self.start\_time = time()  
 return self  
  
 def \_\_exit\_\_(self, type, val, traceback):  
 go\_time = time() - self.start\_time  
 print(f"\ntime: {go\_time}")  
  
  
@contextmanager  
def cm\_timer2():  
 start\_time = time()  
 try:  
 yield  
 finally:  
 go\_time = time() - start\_time  
 print(f"time: {go\_time}")  
  
  
with cm\_timer\_1():  
 sleep(5.5)  
  
with cm\_timer2():  
 sleep(5.5)

Результат:

Изображение выглядит как текст, Шрифт, снимок экрана, белый

Автоматически созданное описание

Задание:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Код:

import json  
from Lab2.lab\_python\_fp.cm\_timer import cm\_timer\_1  
from Lab2.lab\_python\_fp.unique import Unique  
  
path = "C:\\Users\\User\\PycharmProjects\\PIKYAP\\Lab2\\data\_light.json"  
  
with open(path, encoding='utf-8') as f:  
 data = json.load(f)  
  
  
def print\_result(func):  
 def wrapper(arg):  
 print('\n', func.\_\_name\_\_)  
 for el in func(arg):  
 print(el, end=', ')  
 return func(arg)  
  
 return wrapper  
  
  
@print\_result  
def f1(datta):  
 if len(data) != 0:  
 return sorted(Unique([el['job-name'] for el in datta if 'job-name' in el]))  
 raise NotImplemented  
  
  
@print\_result  
def f2(items):  
 if len(items) != 0:  
 return list(filter(lambda x: x[:11] == 'программист', items))  
 raise NotImplemented  
  
  
@print\_result  
def f3(arg):  
 if len(arg) != 0:  
 return [el + ' с опытом Python' for el in arg]  
 raise NotImplemented  
  
  
@print\_result  
def f4(arg):  
 if len(arg) != 0:  
 from random import randint  
 res = [el + f' заработок {randint(100000, 300000)}' for el in arg]  
 return res  
 raise NotImplemented  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 with cm\_timer\_1():  
 f4(f3(f2(f1(data))))

Результат:

