

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

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по лабораторной работе №3
«Функциональные возможности языка Python.»

Выполнил:
студент группы ИУ5-34Б:
Такташова Дарья Юрьевна
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е
Подпись и дата:

Москва, 2022 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.

Код программы:

lab_python_fp/cm_timer.py

```
lab_python_fp > cm_timer.py > Python > cm_timer_2 > __init__
1  from contextlib import contextmanager
2  import time
3  from unicodedata import name
4
5  @contextmanager
6  def cm_timer_1():
7      start_time = time.time()
8      yield
9      print("time: ",time.time()-start_time)
10
11  class cm_timer_2:
12      def __init__(self):
13          self.start_time = 0
14      def __enter__(self):
15          self.start_time = time.time()
16      def __exit__(self, type, value, traceback):
17          print("time: ", time.time() - self.start_time)
18
19  if __name__ == "__main__":
20
21      with cm_timer_1():
22          time.sleep(5.5)
23
24      with cm_timer_2():
25          time.sleep(5.5)
26
```

Пример выполнения:

```
• dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT/lab_python_fp$ python3 cm_timer.py
time: 5.502336263656616
time: 5.503959894180298
```

lab_python_fp/field.py

```

lab_python_fp > field.py > ...
1  def field(items, *args):
2      result = {}
3      assert len(args) > 0
4      for d in items:
5          for i,j in d.items():
6              if i in args:
7                  result[i] = j
8              if len(result) == 1:
9                  s = result.popitem()
10                 s = "" + str(s[1]) + ""
11                 yield s
12             else :
13                 yield result
14
15  if __name__ == '__main__':
16      goods = [{'title': 'Ковер', 'price': 2000, 'color': 'green'},
17               {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}]
18      for i in field(goods, 'title'):
19          print(i)
20      for i in field(goods, 'price', 'title'):
21          print(i)
22

```

Пример выполнения:

```

dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT/lab_python_fp$ python3 field.py
'Ковер'
'Диван для отдыха'
{'title': 'Ковер', 'price': 2000}
{'title': 'Диван для отдыха', 'price': 5300}

```

lab_python_fp/gen_random.py

```

lab_python_fp > gen_random.py > ...
1  import random
2
3  def gen_random(num_count, begin, end):
4      for i in range(0, num_count, 1):
5          yield (random.randint(begin, end))
6

```

Пример выполнения:

```

dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT/lab_python_fp$ python3 gen_random.py
[2, 1, 2, 1, 3]

```

lab_python_fp/print_result.py

```

lab_python_fp > print_result.py > Jupyter > print_result > inner
1  def print_result(func):
2      def inner(*args, **kwargs):
3          print(func.__name__)
4          res = func(*args, **kwargs)
5          if type(res) == list:
6              for i in res:
7                  print(i)
8              elif type(res) == dict:
9                  for key in res.keys():
10                     print(f''{key} = {res[key]}''')
11             else:
12                 print(res)
13             return res
14         return inner

```

```

15
16 @print_result
17 def test_1():
18     return 1
19
20
21 @print_result
22 def test_2():
23     return 'iu5'
24
25
26 @print_result
27 def test_3():
28     return {'a': 1, 'b': 2}
29
30
31 @print_result
32 def test_4():
33     return [1, 2]
34
35
36 if __name__ == '__main__':
37     print('!!!!!!!')
38     test_1()
39     test_2()
40     test_3()
41     test_4()

```

Пример выполнения:

```

dasha@dasha-BONB-WAX9:~/Desktop/BIKT/BIKT/lab_python_fp$ python3 print_result.py
!!!!!!!
test_1
1
test_2
iu5
test_3
a = 1
b = 2
test_4
1
2

```

lab_python_fp/sort.py

```

lab_python_fp > sort.py > ...
1 data = [4, -30, 100, -100, 123, 1, 0, -1, -4]
2
3 if __name__ == '__main__':
4     result = sorted(data, key = abs, reverse = True)
5     print(result)
6
7     result_with_lambda = sorted(data, key = lambda n: -abs(n))
8     print(result_with_lambda)
9

```

Пример выполнения:

```

dasha@dasha-BONB-WAX9:~/Desktop/BIKT/BIKT/lab_python_fp$ python3 sort.py
[123, 100, -100, -30, 4, -4, 1, -1, 0]
[123, 100, -100, -30, 4, -4, 1, -1, 0]

```

lab_python_fp/unique.py

```

lab_python_fp > unique.py > ...
1  # Итератор для удаления дубликатов
2  import gen_random
3
4  class Unique(object):
5      def __init__(self, items, **kwargs):
6          self.seen = []
7          for i in items: #AbC
8              if len(kwargs) > 0 and kwargs["ignore_case"]:
9                  flag = True
10                 for j in self.seen:
11                     if j.lower() == i.lower():
12                         flag = False
13                 if flag:
14                     (self.seen).append(i)
15             else:
16                 if i in self.seen:
17                     continue
18                 self.seen.append(i)
19
20     def __next__(self):
21         if len(self.seen) == 0:
22             raise StopIteration
23         item = self.seen[0]
24         del self.seen[0]
25         return item
26
27     def __iter__(self):
28         return self
29
30 if __name__ == '__main__':
31     print("1st test")
32
33     data = ['a', 'A', 'b', 'B', 'a', 'A', 'b', 'B']
34
35     t = Unique(data, ignore_case = True)
36
37     print(next(t))
38     print([next(t)])
39
40     #print(next(t))
41
42
43     print("2nd test")
44
45     data = [1,1,1,1,1,1,2,2,2,2,2]
46
47     t = Unique(data)
48
49     print(next(t))
50     print(next(t))
51
52
53     print("3rd test")
54     data = gen_random.gen_random(10, 1, 3)
55
56     t = Unique(data)
57
58     print(next(t))
59     print(next(t))
60     print(next(t))
61

```

Пример выполнения:

```

dasha@dasha-B0HB-WAX9:~/Desktop/BIKT/BIKT/lab_python_fp$ python3 unique.py
[1, 2, 2, 1, 1]
1st test
a
b
2nd test
1
2
3rd test
3
1
2

```

lab_python_fp/process_data.py

```

32
33 if __name__ == '__main__':
34     with cm_timer.cm_timer_1():
35         f4(f3(f2(f1(data))))
36
37
38 import cm_timer
39 from field import field
40
41 path = '/home/dasha/Desktop/BIKT/BIKT/lab_python_fp/data_light.json'
42
43 with open(path) as f:
44     data = json.load(f)
45
46 @print_result
47 def f1(arg):
48     return list(Unique([x['job-name'] for x in arg]))
49
50
51 @print_result
52 def f2(arg):
53     return list(filter(lambda s: s.startswith('Программист'), arg))
54
55
56 @print_result
57 def f3(arg):
58     return list(map(lambda s: s + ' с опытом Python', arg))
59
60
61 @print_result
62 def f4(arg):
63     return list(zip(arg, gen_random(len(arg), 100000, 200000)))
64
65

```

Анализ результатов

```
f2
Программист
Программист C++/C#/Java
Программист 1C
Программист-разработчик информационных систем
Программист C++
Программист/ Junior Developer
Программист / Senior Developer
Программист/ технический специалист
Программист C#
f3
Программист с опытом Python
Программист C++/C#/Java с опытом Python
Программист 1C с опытом Python
Программист-разработчик информационных систем с опытом Python
Программист C++ с опытом Python
Программист/ Junior Developer с опытом Python
Программист / Senior Developer с опытом Python
Программист/ технический специалист с опытом Python
Программист C# с опытом Python
f4
('Программист с опытом Python', 134536)
('Программист C++/C#/Java с опытом Python', 122936)
('Программист 1C с опытом Python', 107324)
('Программист-разработчик информационных систем с опытом Python', 164341)
('Программист C++ с опытом Python', 181252)
('Программист/ Junior Developer с опытом Python', 147702)
('Программист / Senior Developer с опытом Python', 192248)
('Программист/ технический специалист с опытом Python', 180219)
('Программист C# с опытом Python', 106485)
time: 0.09866547584533691
```