

Московский Государственный Технический университет им. Н. Э.  
Баумана

Отчёт по Лабораторной работе №4  
«Функциональные возможности Python»

Выполнил:

Зубаиров В. А.

## Исходный код

### Gens.py

```
import random

# Генератор вычленения полей из массива словарей
def field(items, *args):
    assert len(args) > 0

    if len(args) == 1:
        for it in items:
            for key in args:
                elem = it.get(key)
                if elem is not None:
                    yield elem
    else:
        for it in items:
            res_dict = {}
            for key in args:
                elem = it.get(key)
                if elem is not None:
                    res_dict[key] = elem
            if len(res_dict) > 0:
                yield res_dict

# Генератор списка случайных чисел
def gen_random(begin, end, num_count):
    for i in range(num_count):
        yield random.randint(begin, end)
```

### iterators.py

```
# Итератор для удаления дубликатов
class Unique(object):
    def __init__(self, items, **kwargs):
        # В качестве ключевого аргумента, конструктор должен принимать bool-
        # параметр ignore_case,
        # в зависимости от значения которого будут считаться одинаковые
        # строки в разном регистре
        # По-умолчанию ignore_case = False
        self.ignore_case = kwargs.get('ignore_case', False)
        if isinstance(items, list):
            self.items = (x for x in items)
        else:
            self.items = items
        self._s = set()

    def __next__(self):
        for i in self.items:
            is_str = isinstance(i, str)
            if (not is_str) and (i not in self._s):
                self._s.add(i)
                return i
            elif is_str:
                if self.ignore_case and (i.lower() not in self._s):
                    self._s.add(i.lower())
```

```

        return i
    elif (not self.ig_case) and (i not in self._s):
        self._s.add(i)
        return i
    else:
        raise StopIteration()

def __iter__(self):
    return self

```

## Decorators.py

```

def print_result(func):
    def decor(*args):
        print("*****", func.__name__, sep='\n')
        collect = func(*args)
        if isinstance(collect, list):
            print('\n'.join(map(str, collect)))
        elif isinstance(collect, dict):
            print('\n'.join(map(lambda k: str(k) + ' = ' + str(collect[k]),
                                collect)))
        else:
            print(collect)
        print("*****")
        return collect

    return decor

```

## Скриншоты результатов

```

"C:\Program Files\Python36\python.exe" C:/Users/Тигрь/labPy/lab4/ex_1.py
Ковер Диван для отдыха Стелаж Вешалка для одежды
{'title': 'Ковер', 'price': 2000} {'title': 'Диван для отдыха', 'price': 5300} {'title': 'Стелаж', 'price': 7000} {'title': 'Вешалка для одежды', 'price': 800}
5 1 3 1
Process finished with exit code 0

```

```

#!/usr/bin/env python3
import ...

from librip.gens import gen_random
from librip.iterators import Unique

data1 = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2]
data2 = gen_random(1, 3, 10)
data3 = ['A', 'a', 'b']

un = Unique(data1)
un2 = Unique(data2)
un3 = Unique(data3, ignore_case=True)
un4 = Unique(data3)

print('list:', ' '.join(map(str, un)), 'generator:', ' '.join(map(str, un2)), sep='\n')
print(' '.join(map(str, un3)))
print(' '.join(map(str, un4)))

```

3

```
"C:\Program Files\Python36\python.exe" C:/Users/Тигрь/labPy/lab4/ex_3.py  
[0, 1, -1, 4, -4, -30, 100, -100, 123]
```

Process finished with exit code 0

```
"C:\Program Files\Python36\python.e
```

```
*****
```

```
test_1
```

```
1
```

```
*****
```

```
*****
```

```
test_2
```

```
iu
```

```
*****
```

```
*****
```

```
test_3
```

```
a = 1
```

```
b = 2
```

```
*****
```

```
*****
```

```
test_4
```

```
1
```

```
2
```

```
*****
```

Process finished with exit code 0

```
"C:\Program Files\Python36\python.exe" C:/Users/Тигрь/labPy/lab
*****
f1
Администратор на телефоне
Медицинская сестра
Охранник сутки-день-ночь-вахта
ВРАЧ АНЕСТЕЗИОЛОГ РЕАНИМАТОЛОГ
теплотехник
разнорабочий
Электро-газосварщик
Водитель Gett/Гетт и Yandex/Яндекс такси на личном автомобиле
Монолитные работы
Организатор - тренер
Помощник руководителя
Автоэлектрик
Врач ультразвуковой диагностики в детскую поликлинику
Менеджер по продажам ИТ услуг (B2B)
Менеджер по персоналу
Аналитик
Воспитатель группы продленного дня
Инженер по качеству
Инженер по качеству 2 категории (класса)
Водитель автомобиля
```

```
newtask.py
ewtask

"C:\Program Files\Python36\python.exe"
1 16 25 36
1 25 64 100

Process finished with exit code 0
```