

## Отчет

По лабораторной работе – 3

Дисциплина «Парадигмы и конструкции языков программирования»

Студент: Артёмова Дарья

Группа: ИБМЗ – 34Б

### Задание №1

```
def field(items, *args):
    assert len(args) > 0 # Проверка, что передано хотя бы одно поле
    for item in items: # Проходим по каждому элементу списка
        if len(args) == 1: # Если передано только одно поле
            if item.get(args[0]) is not None: # Проверяем, есть ли значение по этому полю
                yield item.get(args[0]) # Выводим значение поля
        else: # Если передано несколько полей
            new_item = {} # Создаем новый словарь для результата
            for key in args: # Проходим по каждому полю
                if item.get(key) is not None: # Проверяем, есть ли значение по полю
                    new_item[key] = item.get(key) # Добавляем значение в новый словарь
            if new_item: # Если в новый словарь попали какие-либо значения
                yield new_item # Выводим новый словарь

goods = [
    {'title': 'Ковер', 'price': 2000, 'color': 'green'},
    {'title': 'Диван для отдыха', 'price': 5300, 'color': 'black'}
]

for item in field(goods, 'title'):
    print(item) # Вывод: 'Ковер', 'Диван для отдыха'

for item in field(goods, 'title', 'price'):
    print(item) # Вывод: {'title': 'Ковер', 'price': 2000}, {'title': 'Диван для отдыха', 'price': 5300}
```

Вывод:

```
Ковер
Диван для отдыха
{'title': 'Ковер', 'price': 2000}
{'title': 'Диван для отдыха', 'price': 5300}
```

### Задание №2

```
import random

def gen_random(num_count, begin, end):
    for _ in range(num_count):
        yield random.randint(begin, end)

for number in gen_random(5, 3, 8):
    print(number)
```

Вывод:

```
7
7
5
6
5
```

### Задание №3

```
# Итератор для удаления дубликатов
class Unique(object):
    def __init__(self, items, **kwargs):
        self.items = iter(items) # Преобразуем в итератор, если это не генератор
        self.seen = set()
        self.ignore_case = kwargs.get('ignore_case', False)

    def __next__(self):
        while True:
            try:
                item = next(self.items)
                if self.ignore_case and isinstance(item, str):
                    item = item.lower()
                if item not in self.seen:
                    self.seen.add(item)
                    return item
            except StopIteration:
                raise StopIteration

    def __iter__(self):
        return self

# Примеры использования
print("Пример 1:")
data = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2]
for item in Unique(data):
    print(item)

print("\nПример 2:")
data = ['a', 'A', 'b', 'B', 'a', 'A', 'b', 'B']
for item in Unique(data):
    print(item)

print("\nПример 3:")
for item in Unique(data, ignore_case=True):
    print(item)

print("\nПример 4:")
def gen_random(count, min_value, max_value):
    import random
    for _ in range(count):
        yield random.randint(min_value, max_value)

data = gen_random(10, 1, 3)
for item in Unique(data):
    print(item)
```

Вывод:

Пример 1:

1  
2

Пример 2:

a  
A  
b  
B

Пример 3:

a  
b

Пример 4:

3  
1  
2

#### Задание №4

```
data = [4, -30, 100, -100, 123, 1, 0, -1, -4]

if __name__ == '__main__':
    # С использованием lambda-функции
    result_with_lambda = sorted(data, key=lambda x: abs(x), reverse=True)
    print(result_with_lambda)

    # Без использования lambda-функции
    def abs_key(x):
        return abs(x)
    result = sorted(data, key=abs_key, reverse=True)
    print(result)
```

Вывод:

```
[123, 100, -100, -30, 4, -4, 1, -1, 0]
[123, 100, -100, -30, 4, -4, 1, -1, 0]
```