



Министерство науки и высшего образования 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)

ФАКУЛЬТЕТ «ИНЖЕНЕРНЫЙ БИЗНЕС И МЕНЕДЖМЕНТ»

КАФЕДРА «ПРОМЫШЛЕННАЯ ЛОГИСТИКА» (ИБМ-3)

Лабораторная работа № 3-4

«Парадигмы и конструкции языков программирования»

38.03.05 «Бизнес-информатика» (бакалавриат)

Студент ИБМ3-24Б

(Подпись, дата)

А. А. Нургалиева

2024 г.

Задача 1 (файл field.py)

Необходимо реализовать генератор field. Генератор field последовательно выдает значения ключей словаря.

```
#№1
def field(items, *args):
    assert len(args) > 0, "Необходимо указать хотя бы один ключ"

    if len(args) == 1:
        key = args[0]
        for item in items:
            value = item.get(key)
            if value is not None:
                yield value
    else:
        for item in items:
            result = {key: item[key] for key in args if item.get(key) is not None}
            if result:
                yield result

# Пример использования
if __name__ == "__main__":
    goods = [
        {'title': 'Ковер', 'price': 2000, 'color': 'green'},
        {'title': 'Диван для отдыха', 'color': 'black'}
    ]
    print("Задание №1")
    print(list(field(goods, 'title')))
    print(list(field(goods, 'title', 'price')))
```

Задача 2 (файл gen_random.py)

Необходимо реализовать генератор gen_random(количество, минимум, максимум), который последовательно выдает заданное количество случайных чисел в заданном диапазоне от минимума до максимума, включая границы диапазона.

```
import random

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

# Пример использования
if __name__ == "__main__":
    print("Задание №2")
    print(list(gen_random(5, 1, 7))) # Пример вывода: [2, 2, 3, 2, 1]
```

Задача 3 (файл unique.py)

- Необходимо реализовать итератор Unique(данные), который принимает на вход массив или генератор и итерируется по элементам, пропуская дубликаты.

- Конструктор итератора также принимает на вход именованный bool-параметр `ignore_case`, в зависимости от значения которого будут считаться одинаковыми строки в разном регистре. По умолчанию этот параметр равен `False`.
- При реализации необходимо использовать конструкцию `**kwargs`.
- Итератор должен поддерживать работу как со списками, так и с генераторами.
- Итератор не должен модифицировать возвращаемые значения.

```
class Unique:
    def __init__(self, items, **kwargs):
        self.ignore_case = kwargs.get('ignore_case', False)
        self.seen = set()
        self.items = iter(items)

    def __iter__(self):
        return self

    def __next__(self):
        while True:
            item = next(self.items)
            check_item = item.lower() if self.ignore_case and isinstance(item, str)
        else item

            if check_item not in self.seen:
                self.seen.add(check_item)
                return item

# Пример использования
if __name__ == "__main__":
    data = [1, 1, 1, 2, 2, 3, 'a', 'A', 'b', 'B']
    print("Задание №3")
    print(list(Unique(data))) # Вывод: [1, 2, 3, 'a', 'A', 'b', 'B']
    print(list(Unique(data, ignore_case=True)))
```

Задача 4 (файл sort.py)

Дан массив 1, содержащий положительные и отрицательные числа. Необходимо **одной строкой кода** вывести на экран массив 2, который содержит значения массива 1, отсортированные по модулю в порядке убывания. Сортировку необходимо осуществлять с помощью функции `sorted`.

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

if __name__ == '__main__':
    # Без использования lambda-функции
    result = sorted(data, key=abs, reverse=True)
    print(result)

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

Результаты выполнения программы:

```
/usr/local/bin/python3.10 /Users/azaljanurgalieva/PycharmProjects/па
Задание №1
['Ковер', 'Диван для отдыха']
[{'title': 'Ковер', 'price': 2000}, {'title': 'Диван для отдыха'}]
Задание №2
[4, 6, 2, 2, 1]
Задание №3
[1, 2, 3, 'a', 'A', 'b', 'B']
[1, 2, 3, 'a', 'b']
Задание №4
[123, 100, -100, -30, 4, -4, 1, -1, 0]
[123, 100, -100, -30, 4, -4, 1, -1, 0]
```