

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ

по лабораторной работе №2.4

Дисциплина: «Программирование на Python»
Тема: «Работа со списками в языке Python»

Выполнил:
Епифанов Алексей Александрович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных систем
», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Цель: приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Создал новый репозиторий, клонировал его, в нем создал ветку developer и перешел на нее.
2. Проработал примеры лабораторной работы:

```
[aleksejefifanov@MacBook-Pro program % ./prim1.py  
1 2 3 4 -1 -5 8 -3 2 -3  
5  
5  
[aleksejefifanov@MacBook-Pro program % ./prim1.py  
12 -2 34 3 -4 23 -2 19 -3 8  
-8  
-8  
aleksejefifanov@MacBook-Pro program % ./prim1.py  
2 3 4 5 6  
Неверный размер списка
```

Рисунок 1. Несколько запусков программы примера 1

```
● aleksejefifanov@MacBook-Pro program % ./prim2.py  
2 3 4 5  
2  
● aleksejefifanov@MacBook-Pro program % ./prim2.py  
3 2 -2 3 -23 33 4 2  
0  
● aleksejefifanov@MacBook-Pro program % ./prim2.py  
3 -2 33 -23 4 -22 5 -35 6 8 5  
2  
⊗ aleksejefifanov@MacBook-Pro program % ./prim2.py  
  
Заданный список пуст  
● aleksejefifanov@MacBook-Pro program % ./prim2.py  
1  
0  
○ aleksejefifanov@MacBook-Pro program % |
```

Рисунок 2. Несколько запусков программы примера 2

3. Выполнил индивидуальное задание 1 вариант 10: Ввести список A из 10 элементов, найти произведение положительных элементов кратных 3, их количество и вывести результаты на экран.

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-  
import sys  
if __name__ == '__main__':  
    print("Введите список элементов (через пробел):")  
    A = list(map(int, input().split()))  
    if len(A) != 10:
```

```

print("Неверный размер списка", file=sys.stderr)
exit(1)
product = 1
B = list(filter(lambda x: x > 0 and x % 3 == 0, A))
count = len(B)
for num in B:
    product *= num
print(f"Произведение положительных элементов, кратных 3: {product}")
print(f"Количество положительных элементов, кратных 3: {count}")
product = 1
count = 0
for num in A:
    if num > 0 and num % 3 == 0:
        product *= num
        count += 1
print(f"Произведение положительных элементов, кратных 3: {product}")
print(f"Количество положительных элементов, кратных 3: {count}")

```

```

❌ aleksejerifanov@MacBook-Pro program % ./ind1.py
Введите список элементов (через пробел):
1 2 3
Неверный размер списка
● aleksejerifanov@MacBook-Pro program % ./ind1.py
Введите список элементов (через пробел):
1 2 3 4 5 6 7 8 9 10
Произведение положительных элементов, кратных 3: 162
Количество положительных элементов, кратных 3: 3
Произведение положительных элементов, кратных 3: 162
Количество положительных элементов, кратных 3: 3
● aleksejerifanov@MacBook-Pro program % ./ind1.py
Введите список элементов (через пробел):
2 4 6 8 10 12 14 16 18 20
Произведение положительных элементов, кратных 3: 1296
Количество положительных элементов, кратных 3: 3
Произведение положительных элементов, кратных 3: 1296
Количество положительных элементов, кратных 3: 3
● aleksejerifanov@MacBook-Pro program % ./ind1.py
Введите список элементов (через пробел):
-3 -6 -9 2 3 4 5 6 -2 3
Произведение положительных элементов, кратных 3: 54
Количество положительных элементов, кратных 3: 3
Произведение положительных элементов, кратных 3: 54
Количество положительных элементов, кратных 3: 3
○ aleksejerifanov@MacBook-Pro program % |

```

Рисунок 4. Несколько запусков программы индивидуального задания 1

4. Выполнил индивидуальное задание 2 вариант 10: В списке, состоящем из вещественных элементов, вычислить:

- номер минимального по модулю элемента списка;
- сумму модулей элементов списка, расположенных после первого отрицательного элемента.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import math
import sys
if __name__ == '__main__':
    print("Введите список неповторяющихся вещественных чисел через пробел")
    input_list = list(map(float, input().split()))
    if len(input_list) != len(set(input_list)):
        print("Есть повторяющиеся элементы", file=sys.stderr)
        exit(1)
    print("Также введите интервал [a, b], то есть 2 числа через пробел")
    a, b = map(float, input().split())
    m_index = 0
    m_value = math.fabs(input_list[0])
    for i, num in enumerate(input_list):
        abs_value = math.fabs(num)
        if abs_value < m_value:
            m_index = i
            m_value = abs_value
    summ = 0
    negative_found = False
    for num in input_list:
        if negative_found:
            summ += abs(num)
        elif num < 0:
            negative_found = True
    ind = 0
    for num in input_list.copy():
        if a <= num <= b:
            input_list.pop(ind)
            input_list.append(0)
        else:
            ind += 1
    print(f"Номер минимального по модулю элемента: {m_index}")
    print(f"Сумма модулей элементов после первого отрицательного: {summ}")
    print(f"Сжатый список: {input_list}")

```

```

⊗ aleksejefifanov@MacBook-Pro program % ./ind2.py
Введите список неповторяющихся вещественных чисел через пробел
2 3.4 4.9 -2.3 7.8 -4.2 -4.1 3 -3 2 5 -2.34
Есть повторяющиеся элементы
● aleksejefifanov@MacBook-Pro program % ./ind2.py
Введите список неповторяющихся вещественных чисел через пробел
2 3.4 4.9 -2.3 7.8 -4.2 -4.1 3 -3 4 3.332 5.4323 -4.234
Также введите интервал [a, b], то есть 2 числа через пробел
3 6
Номер минимального по модулю элемента: 0
Сумма модулей элементов после первого отрицательного: 39.0983
Сжатый список: [2.0, -2.3, 7.8, -4.2, -4.1, -3.0, -4.234, 0, 0, 0, 0, 0, 0]
● aleksejefifanov@MacBook-Pro program % ./ind2.py
Введите список неповторяющихся вещественных чисел через пробел
23 34 32.32 -23.32 -5 5.7 1.8 4.343 -3
Также введите интервал [a, b], то есть 2 числа через пробел
2 30
Номер минимального по модулю элемента: 6
Сумма модулей элементов после первого отрицательного: 19.843
Сжатый список: [34.0, 32.32, -23.32, -5.0, 1.8, -3.0, 0, 0, 0]
○ aleksejefifanov@MacBook-Pro program % |

```

Рисунок 5. Несколько запусков программы индивидуального задания 2

Ответы на контрольные вопросы:

1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов.

2. Как осуществляется создание списка в Python?

Для создания списка нужно заключить элементы в квадратные скобки.

3. Как организовано хранение списков в оперативной памяти?

При создании списка в памяти резервируется область, которую можно условно назвать некоторым «контейнером», в котором хранятся ссылки на другие элементы данных в памяти. В отличие от таких типов данных как число или строка, содержимое «контейнера» списка можно менять.

4. Каким образом можно перебрать все элементы списка?

Можно воспользоваться циклом for: for i in list.

5. Какие существуют арифметические операции со списками?

Для объединения списков можно использовать оператор сложения (+).

Список можно повторить с помощью оператора умножения (*).

6. Как проверить есть ли элемент в списке?

Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор in.

7. Как определить число вхождений заданного элемента в списке?

Метод `count(элемент)` можно использовать для определения числа сколько раз данный элемент встречается в списке.

8. Как осуществляется добавление (вставка) элемента в список?

Метод `insert(индекс вставки, элемент)` можно использовать, чтобы вставить элемент в список.

Метод `append()` можно использовать для добавления элемента в конец списка.

9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort()`.

10. Как удалить один или несколько элементов из списка?

Удалить элемент можно, написав его индекс в методе `pop(индекс)`.

Элемент можно удалить с помощью метода `remove(значение)`.

Оператор `del` можно использовать для тех же целей.

Можно удалить все элементы из списка с помощью метода `clear`.

11. Что такое списковое включение и как с его помощью осуществлять обработку списков?

List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков. В языке Python есть две очень мощные функции для работы с коллекциями: `map` и `filter`. Они позволяют использовать функциональный стиль программирования, не прибегая к помощи циклов, для работы с такими типами как `list`, `tuple`, `set`, `dict` и т.п. Списковое включение позволяет обойтись без этих функций.

Примеры: `a = [i for i in range(n)]` создаст список `a`, содержащий все числа от 0 до `n` не включительно. `b = list(map(lambda x: x**2, a))` создаст список `b`, в котором каждый элемент будет являться квадратом соответствующего элемента из списка `a`. `b = list(filter(lambda x: x % 2 == 0, a))` создаст список `b`, содержащий только четные элементы списка `a`.

12. Как осуществляется доступ к элементам списков с помощью срезов?

Слайсы (срезы) являются очень мощной составляющей Python, которая позволяет быстро и лаконично решать задачи выборки элементов из списка. Слайс задается тройкой чисел, разделенных запятой: start:stop:step. Start – позиция, с которой нужно начать выборку, stop – конечная позиция, step – шаг. При этом необходимо помнить, что выборка не включает элемент определяемый stop.

13. Какие существуют функции агрегации для работы со списками?

Для работы со списками Python предоставляет следующие функции:

len(L) - получить число элементов в списке L.

min(L) - получить минимальный элемент списка L.

max(L) - получить максимальный элемент списка L.

sum(L) - получить сумму элементов списка L, если список L содержит только числовые значения.

Для функций min и max элементы списка должны быть сравнимы между собой.

14. Как создать копию списка?

Для создания копии списка необходимо использовать либо метод copy(), либо использовать оператор среза.

15. Самостоятельно изучите функцию sorted языка Python. В чем ее отличие от метода sort списков?

Отличие между sorted() и sort() заключается в том, что sorted(список) возвращает новый отсортированный список без изменения исходного, а sort() изменяет сам исходный список.

Вывод: в результате выполнения работы были приобретены навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.