

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2.4**  
**дисциплины «Программирование на Python»**

**Вариант 15**

Выполнил:  
Кенесбаев Хилол Куат улы  
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 Р.А., доцент кафедры  
инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Работа со списками в языке Python

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

**Порядок выполнения работы:**

1. Проработал примеры лабораторной работы:

```
1 import sys
2
3 if __name__ == '__main__':
4     # Ввести список одной строкой.
5     A = list(map(int, input().split()))
6     # Проверить количество элементов списка.
7     if len(A) != 10:
8         print("Неверный размер списка", file=sys.stderr)
9         exit(1)
10    # Найти искомую сумму 1.
11    s = 0
12    for item in A:
13        if abs(item) < 5:
14            s += item
15    print(s)
16    s = 0
17    # Найти искомую сумму 2.
18    s = sum([a for a in A if abs(a) < 5])
19    print(s)
```

*Рисунок 1 primer1.py*

```
C:\Users\HP-PC\AppData\Local\Programs\Python\Python312\python.exe C:\Users\HP-PC\PycharmProjects\lab-7\primer1.py
1 2 3 4 5 6 7 8 9 10
10
10
Process finished with exit code 0
```

*Рисунок 2 Результат работы программы primer1.py*

```
1 import sys
2
3
4 if __name__ == '__main__':
5     # Ввести список одной строкой.
6     a = list(map(int, input().split()))
7     # Если список пуст, завершить программу.
8     if not a:
9         print("Заданный список пуст", file=sys.stderr)
10        exit(1)
11    # Определить индексы минимального и максимального элементов.
12    a_min = a_max = a[0]
13    i_min = i_max = 0
14    for i, item in enumerate(a):
15        if item < a_min:
16            i_min, a_min = i, item
17        if item >= a_max:
18            i_max, a_max = i, item
19    # Проверить индексы и обменять их местами.
```

*Рисунок 3 primer2.py*

```
C:\Users\HP-PC\AppData\Local\Programs\Python\Python312\python.exe C:\Users\HP-PC\PycharmProjects\lab-7\primer2.py
5 4 3 2 1
3
Process finished with exit code 0
```

*Рисунок 4 Результат работы программы primer2.py*

### **Выполнил индивидуальное задание 1 :**

Ввести список А из 10 элементов, найти произведение элементов, больших 8 и меньших 18 и кратных 10, их количество и вывести результаты на экран.

### **Код программы:**

```
def calculate_product_and_count(arr):
```

```
    product = 1
```

```
    count = 0
```

```
    for num in arr:
```

```
        if num > 8 and num < 18 and num % 10 == 0:
```

```
            product *= num
```

```
            count += 1
```

```
    return product, count
```

```
# Ввод списка А из 10 элементов
```

```
A = [int(input(f'Введите элемент {i + 1}: ')) for i in range(10)]
```

```
# Нахождение произведения элементов, удовлетворяющих условиям, и их количества
```

```
result_product, result_count = calculate_product_and_count(A)
```

```
print(f'Произведение элементов, больших 8, меньших 18 и кратных 10: {result_product}')
```

```
print(f'Количество таких элементов: {result_count}')
```

```
C:\Users\HP-PC\AppData\Local\Programs\Python\Python312\python.exe C:\Users\HP-PC\Desktop\study\python\непесдача\Lab2.4\код\individualnoe1.py
Введите элемент 1: 7
Введите элемент 2: 8
Введите элемент 3: 5
Введите элемент 4: 68
Введите элемент 5: 1
Введите элемент 6: 25
Введите элемент 7: 1
Введите элемент 8: 6
Введите элемент 9: 5
Введите элемент 10: 93
Произведение элементов, больших 8, меньших 18 и кратных 10: 1
Количество таких элементов: 0
```

*Рисунок 5 Результат работы программы individualnoe1.py*

## **Выполнил индивидуальное задание 2:**

В списке, состоящем из вещественных элементов, вычислить:

1. Количество отрицательных элементов списка;
2. Сумму модулей элементов списка, расположенных после минимального по модулю элемента.

Заменить все отрицательные элементы списка их квадратами и упорядочить элементы списка по возрастанию.

### **Код программы:**

```
def process_list(input_list):

    # 1. Подсчет отрицательных элементов
    neg_count = sum(1 for num in input_list if num < 0)

    # Находим минимальный элемент по модулю и его индекс
    min_abs_value = min(map(abs, input_list))

    min_abs_index = input_list.index(min_abs_value) if min_abs_value in input_list
    else input_list.index(-min_abs_value)

    # 2. Вычисляем сумму модулей элементов после минимального по модулю элемента

    sum_after_min = sum(abs(num) for num in input_list[min_abs_index+1:])

    # Замена отрицательных элементов и сортировка списка по возрастанию

    modified_list = [num**2 if num < 0 else num for num in input_list]

    modified_list.sort()

    return neg_count, sum_after_min, modified_list

# Ввод списка вещественных элементов

input_list = [float(input(f"Введите элемент {i + 1}: ")) for i in range(10)]

# Обработка списка

neg_count, sum_after_min, sorted_list = process_list(input_list)

print(f"Количество отрицательных элементов: {neg_count}")

print(f"Сумма модулей элементов после минимального по модулю элемента: {sum_after_min}")

print(f"Измененный и отсортированный список: {sorted_list}")
```

```
C:\Users\HP-PC\AppData\Local\Programs\Python\Python312\python.exe C:\Users\HP-PC\Desktop\study\python\пересдача\Lab2.4\код\individualnoe2.py
Введите элемент 1: -3
Введите элемент 2: 5
Введите элемент 3: -2
Введите элемент 4: 9
Введите элемент 5: 7
Введите элемент 6: -4
Введите элемент 7: 1
Введите элемент 8: 8
Введите элемент 9: 9
Введите элемент 10: 11
Количество отрицательных элементов: 3
Сумма модулей элементов после минимального по модулю элемента: 28.0
Измененный и отсортированный список: [1.0, 4.0, 5.0, 7.0, 8.0, 9.0, 9.0, 9.0, 11.0, 16.0]
```

*Рисунок 6 Результат работы программы individualnoe2.py*

## 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.