

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 №7
дисциплины «Основы программной инженерии»

Выполнила:
Ламская Ксения Вячеславовна
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: работа со списками в языке Python.

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

Порядок выполнения работы


1. Создала репозиторий GitHub.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 ksenia-lamskaya ▾

Repository name *

/ 7laba

✓ 7laba is available.

Great repository names are short and memorable. Need inspiration? How about [special-funicular](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: Python ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  main as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория GitHub

2. Проработала примеры из лабораторной работы.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':

    # Ввести список одной строкой.
    A = list(map(int, input().split()))

    # Проверить количество элементов списка.
    if len(A) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

Рисунок 2.1 – Код из примера 1

```
2 3 4 1 5 6 -1 0 4 5
13
```

Рисунок 2.2 – Вывод программы из примера 1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    # Ввести список одной строкой.
    a = list(map(int, input().split()))

    # Если список пуст, завершить программу.
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    # Определить индексы минимального и максимального элементов.
    a_min = a_max = a[0]
    i_min = i_max = 0
    for i, item in enumerate(a):
        if item < a_min:
            i_min, a_min = i, item
        if item >= a_max:
            i_max, a_max = i, item

    # Проверить индексы и обменять их местами.
    if i_min > i_max:
        i_min, i_max = i_max, i_min

    # Посчитать количество положительных элементов.
    count = 0
    for item in a[i_min+1:i_max]:
        if item > 0:
            count += 1

    print(count)
```

Рисунок 2.3 – Код из примера 2

```
3 2 4 1 5 6 1 3 -3 -4 -4
2
```

Рисунок 2.4 – Вывод программы из примера 2

9. Составить программу, выдающую индексы заданного элемента или сообщаящую, что такого элемента в списке нет.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    print('Введите элементы списка: ')
    lst = [input() for _ in range(10)]
    element = input('Введите элемент: ')
    index = []

    for ind, value in enumerate(lst):
        if value == element:
            index.append(ind)

    if len(index) == 0:
        print("Такого элемента в списке нет")
    else:
        print('Индексы этого элемента: ')
        [print(i) for i in index]
```

Рисунок 3.1 – Код программы

```
Введите элементы списка:
1
2
3
4
5
6
7
8
9
10
Введите элемент: 4
Индексы этого элемента:
3
```

Рисунок 3.2 – Вывод программы

9. В списке, состоящем из целых элементов, вычислить:

- минимальный по модулю элемент списка;
- сумму модулей элементов списка, расположенных после первого элемента, равного нулю.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    numbers = [int(input('Введите элементы списка: ')) for _ in range(10)]

    # 1. минимальный по модулю элемент списка
    min_abs = min(numbers, key=abs)
    print("Минимальный по модулю элемент списка:", abs(min_abs))

    # 2. сумма модулей элементов списка после первого элемента, равного нулю
    if 0 in numbers:
        zero_index = numbers.index(0)
        sum_abs = sum(abs(x) for x in numbers[zero_index+1:])
        print("Сумма модулей элементов списка, расположенных после первого элемента, равного нулю:", sum_abs)
    else:
        print('В списке нет элемента 0')
```

Рисунок 4.1 – Код программы

```
Введите элементы списка: 0
Введите элементы списка: 1
Введите элементы списка: 2
Введите элементы списка: 3
Введите элементы списка: 4
Введите элементы списка: 6
Введите элементы списка: 7
Введите элементы списка: 8
Введите элементы списка: 9
Минимальный по модулю элемент списка: 0
Сумма модулей элементов списка, расположенных после первого элемента, равного нулю: 45
```

Рисунок 4.2 – Вывод программы

Ответы на контрольные вопросы

1. Что такое списки в языке Python?

Список – это структура данных, которая хранит различные объекты.

2. Как осуществляется создание списка в Python?

Создание списка осуществляется с помощью квадратных скобок:

```
a = []
```

3. Как организовано хранение списков в оперативной памяти?

Переменная, которая объявлена как список, хранит ссылку на структуру (можно сказать контейнер) в памяти, которая в свою очередь хранит ссылки на элементы данных.

4. Каким образом можно перебрать все элементы списка?

Перебрать все элементы списка можно с помощью циклов.

5. Какие существуют арифметические операции со списками?

Списки можно складывать (конкатенация) и умножать на число (в данном случае список повторится n раз).

6. Как проверить есть ли элемент в списке?

С помощью оператора in.

7. Как определить число вхождений заданного элемента в списке?

С помощью метода count.

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

Вставка – метод insert(), который принимает в качестве аргументов индекс, по которому вставляется, и сам вставляемый элемент; добавление элемента – метод append.

9. Как выполнить сортировку списка?

С помощью метода sort().

10. Как удалить один или несколько элементов из списка?

Удалить один элемент: методы pop, remove. Удалить несколько при помощи функции del и среза.

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

Списковое включение – это способ построения списков [i for i in range(10)], можно также использовать условие, для обработки.

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

[start:stop:step]

13. Какие существуют функции агрегации для работы со списками?

min(), max(), sum(), len().

14. Как создать копию списка?

С помощью функции copy() или среза [:].

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

Функция sorted может применяться не только к спискам, и она возвращает новый список.