

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**«Работа со списками в языке Python»**

**ОТЧЕТ**  
**по лабораторной работе №7**  
**дисциплины**  
**«Основы программной инженерии»**

Выполнил:

Гълбачева Доротея Андреева  
2 курс, группа ПИЖ-б-о-21-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка  
и сопровождение программного  
обеспечения», очная форма обучения

---

(подпись)

Проверил:

---

(подпись)

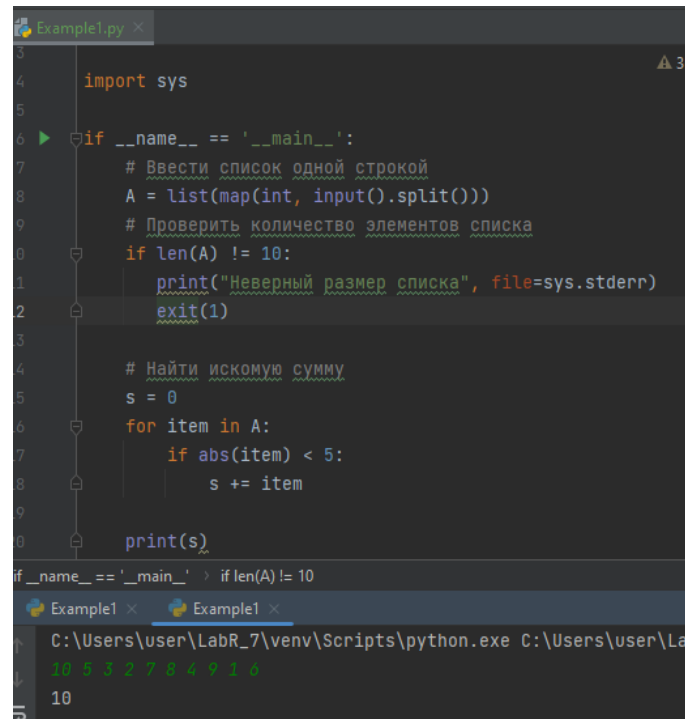
Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2022 г.

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

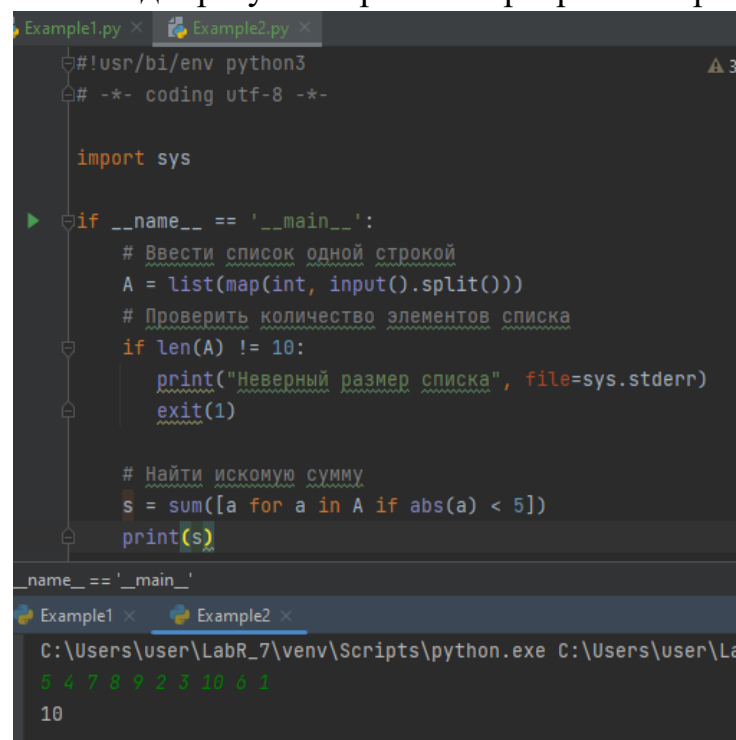
## Выполнение лабораторной работы:

### 1. Проработка примеров из лабораторной работы:



```
Example1.py
3
4 import sys
5
6 if __name__ == '__main__':
7     # Ввести список одной строкой
8     A = list(map(int, input().split()))
9     # Проверить количество элементов списка
10    if len(A) != 10:
11        print("Неверный размер списка", file=sys.stderr)
12        exit(1)
13
14    # Найти искомую сумму
15    s = 0
16    for item in A:
17        if abs(item) < 5:
18            s += item
19
20    print(s)
if __name__ == '__main__': if len(A) != 10
Example1 Example1
C:\Users\user\LabR_7\venv\Scripts\python.exe C:\Users\user\La
10 5 3 2 7 8 4 9 1 6
10
```

Рисунок 7.1 - Код и результат работы программы примера №1



```
Example1.py Example2.py
#!/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 = sum([a for a in A if abs(a) < 5])
    print(s)
__name__ == '__main__'
Example1 Example2
C:\Users\user\LabR_7\venv\Scripts\python.exe C:\Users\user\La
5 4 7 8 9 2 3 10 6 1
10
```

Рисунок 7.2 - Код и результат работы программы примера №1

```

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

_name__ == '__main__'

```

Example1 x Example3 x

C:\Users\user\LabR\_7\venv\Scripts\python.exe C:\Users\user\LabR\_7\Ex

5 4 7 -10 2 8 9 3 0

2

Рисунок 7.3 - Код и результат работы программы примера №2

## 2. Индивидуальные задания:

### Задание №1:

8. В заданном список подсчитать число нулевых элементов и вывести на экран их индексы.

```

Example1.py x Example2.py x Example3.py x Individual1
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    # Ввести список одной строкой
    a = list(map(int, input().split()))
    a1 = []
    s = 0
    for i, v in enumerate(a):
        if v == 0:
            a1.append(i)
            s += 1
    print(f"В списке 0 встречается: {s} раза")
    print("Их индексы: ", *a1)

_name__ == '__main__'

```

Individual1 x Individual1 x

C:\Users\user\LabR\_7\venv\Scripts\python.exe C:\U

2 3 4 0 8 0 4 0 3 0

В списке 0 встречается: 4 раза

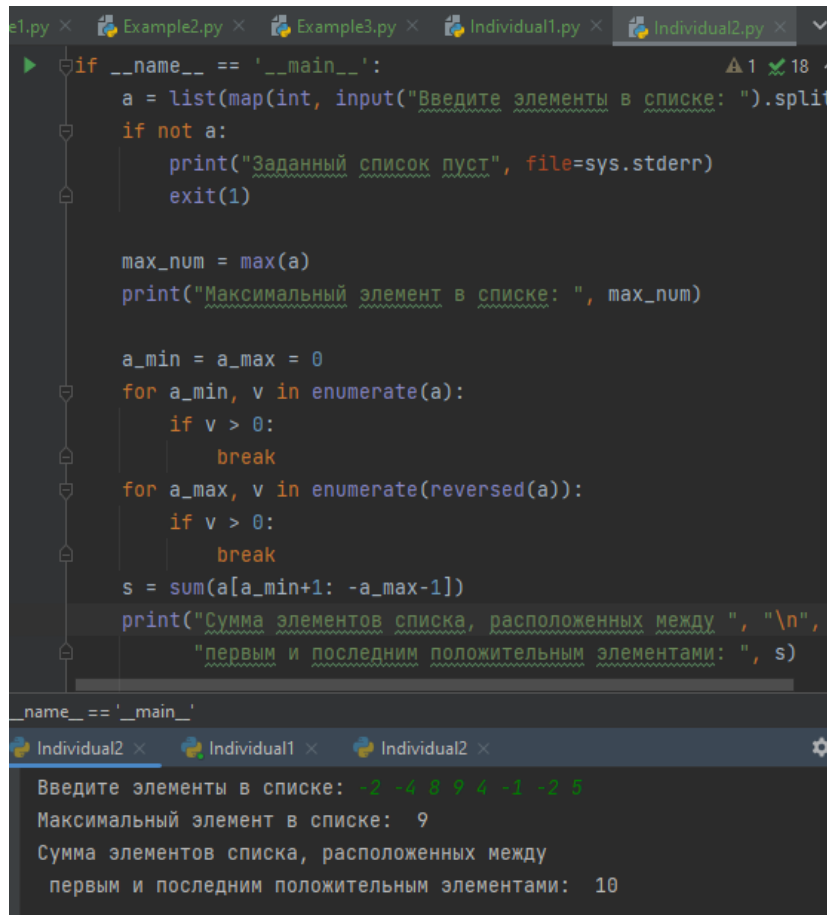
Их индексы: 3 5 7 9

Рисунок 7.4 - Код и результат работы программы инд. задание №1

## Задание №2:

8. В списке, состоящем из вещественных элементов, вычислить:

1. максимальный по модулю элемент списка;
2. сумму элементов списка, расположенных между первым и вторым положительными элементами.



```
if __name__ == '__main__':
    a = list(map(int, input("Введите элементы в списке: ").split()))
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    max_num = max(a)
    print("Максимальный элемент в списке: ", max_num)

    a_min = a_max = 0
    for a_min, v in enumerate(a):
        if v > 0:
            break
    for a_max, v in enumerate(reversed(a)):
        if v > 0:
            break
    s = sum(a[a_min+1: -a_max-1])
    print("Сумма элементов списка, расположенных между ", "\n",
          "первым и последним положительным элементами: ", s)
```

Individual2 x Individual1 x Individual2 x

Введите элементы в списке: -2 -4 8 9 4 -1 -2 5

Максимальный элемент в списке: 9

Сумма элементов списка, расположенных между  
первым и последним положительным элементами: 10

Рисунок 7.5 - Код и результат работы программы инд. задание №2

## 3. Контрольные вопросы:

### 1. Что такое списки в языке Python?

Список (list) – это Для создания списка нужно заключить элементы в квадратные скобки

### 2. Как осуществляется создание списка в Python?

В оперативной памяти списки хранятся в виде ссылок, в которых хранятся ссылки на другие элементы

### 3. Как организовано хранение списков в оперативной памяти?

Перебрать элементы списка можно с помощью цикла.

#### 4. Каким образом можно перебрать все элементы списка?

Читать элементы списка можно с помощью следующего цикла:

```
my_list = ['один', 'два', 'три', 'четыре', 'пять']
for elem in my_list:
    print(elem)
```

#### 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 можно использовать для определения числа сколько раз данный элемент встречается в списке.

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

Метод insert можно использовать, чтобы вставить элемент в список

```
my_list = [1, 2, 3, 4, 5]
my_list.insert(1, 'Привет')
print(my_list)
```

#### 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

```
my_list = ['один', 'два', 'три', 'четыре', 'пять']
removed = my_list.pop(2)
print(my_list)
print(removed)
```

Результат:

```
['один', 'два', 'четыре', 'пять']
три
```

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

Списковое включение (list comprehension) — это компактная и удобная конструкция, применяемая для создания списков из других списков с использованием гибких выражений и условий.

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

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?

Поэтому 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?

`Sort ()` сортирует список на месте, изменяя его индексы и возвращая `None`, тогда как `sorted ()` возвращает новый отсортированный список, оставляя исходный список неизменным. Другое отличие состоит в том, что `sorted ()` принимает любые итерации, в то время как `list`.

