

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФГАОУ высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №5

Работа со списками в языке Python

Выполнил студент группы ИТС-б-з-22-1

Дубинин Олег Александрович

« » \_\_\_\_\_ 2023г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 2023г.

Проверил доцент, кандидат технических  
наук, доцент кафедры инфокоммуникаций

Воронкин Роман Александрович

---

(подпись)

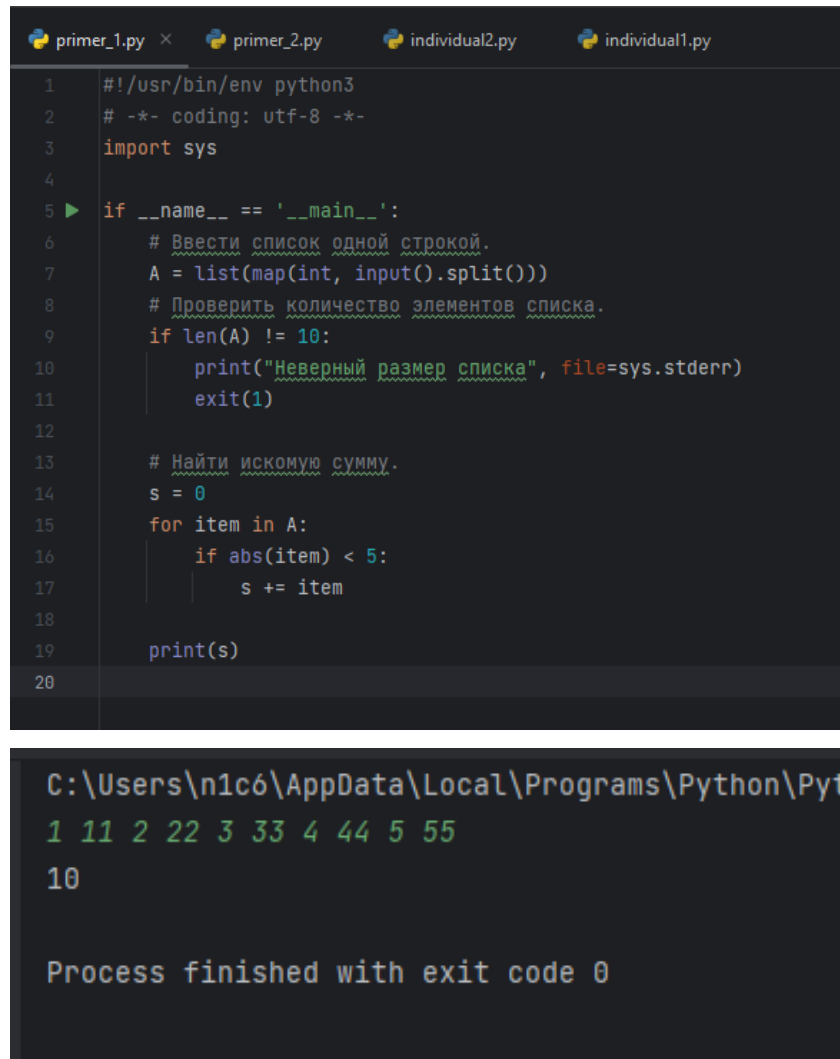
Ставрополь, 2023 г.

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

**Ход работы:**

<https://github.com/OLEG1232155/5> - репозиторий

**Пример 1.** Ввести список *A* из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.



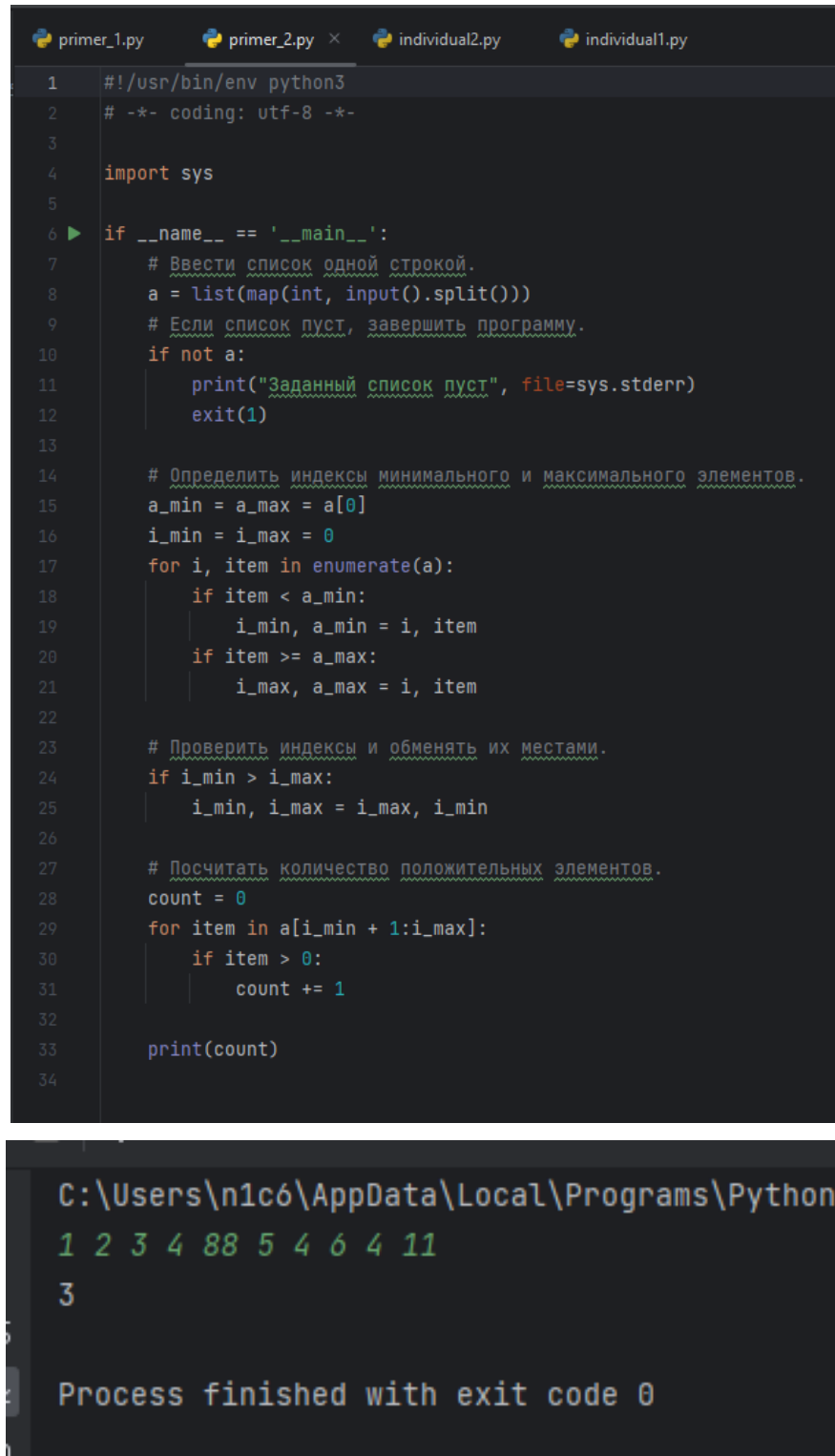
```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import sys
4
5  if __name__ == '__main__':
6      # Ввести список одной строкой.
7      A = list(map(int, input().split()))
8      # Проверить количество элементов списка.
9      if len(A) != 10:
10         print("Неверный размер списка", file=sys.stderr)
11         exit(1)
12
13     # Найти искомую сумму.
14     s = 0
15     for item in A:
16         if abs(item) < 5:
17             s += item
18
19     print(s)
20
```

```
C:\Users\n1c6\AppData\Local\Programs\Python\Pyt
1 11 2 22 3 33 4 44 5 55
10

Process finished with exit code 0
```

Рисунок 1 – Окно программы первого примера

**Пример 2. Написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.**



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      # Ввести список одной строкой.
8      a = list(map(int, input().split()))
9      # Если список пуст, завершить программу.
10     if not a:
11         print("Заданный список пуст", file=sys.stderr)
12         exit(1)
13
14     # Определить индексы минимального и максимального элементов.
15     a_min = a_max = a[0]
16     i_min = i_max = 0
17     for i, item in enumerate(a):
18         if item < a_min:
19             i_min, a_min = i, item
20         if item >= a_max:
21             i_max, a_max = i, item
22
23     # Проверить индексы и обменять их местами.
24     if i_min > i_max:
25         i_min, i_max = i_max, i_min
26
27     # Посчитать количество положительных элементов.
28     count = 0
29     for item in a[i_min + 1:i_max]:
30         if item > 0:
31             count += 1
32
33     print(count)
34
```

```
C:\Users\n1c6\AppData\Local\Programs\Python\
1 2 3 4 88 5 4 6 4 11
3
Process finished with exit code 0
```

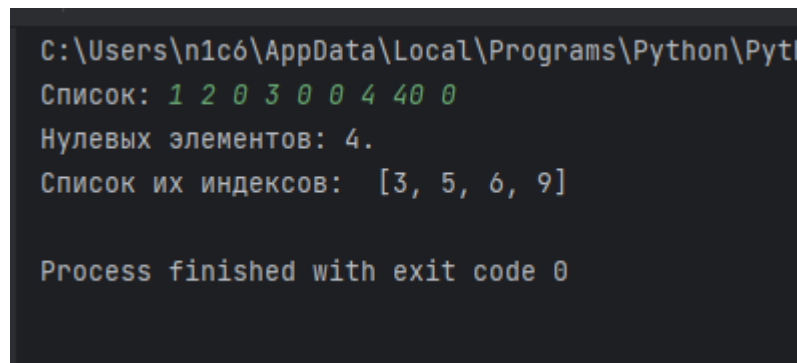
Рисунок 2 – Окно программы второго примера

## Индивидуальное задание 1.

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



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  if __name__ == '__main__':
5      s = list(map(int, input("Список: ").split()))
6      k = 0
7      s1 = []
8
9      for i in range(len(s)):
10         if s[i] == 0:
11             s1.append(i + 1)
12             k += 1
13
14     if k == 0:
15         print("В списке нет нулевых элементов.")
16     else:
17         print("Нулевых элементов: {0}.".format(k))
18         print("Список их индексов: ", s1)
19
```



```
C:\Users\n1c6\AppData\Local\Programs\Python\Python38\python.exe
Список: 1 2 0 3 0 0 4 40 0
Нулевых элементов: 4.
Список их индексов: [3, 5, 6, 9]

Process finished with exit code 0
```

Рисунок 3 – Окно программы для первой задачи и проверка кода на работоспособность.

## Индивидуальное 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.

Преобразовать список таким образом, чтобы элементы, равные нулю, располагались после всех остальных.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 if __name__ == '__main__':
7     s = list(map(float, input("Список: ").split()))
8     if not s:
9         print("Заданный список пуст", file=sys.stderr)
10        exit(1)
11
12    m_max = s[0]
13    s_first = s_second = s[0]
14    i_first = i_second = 0
15    k_first = k_second = 0
16    for i, item in enumerate(s):
17        if m_max < item:
18            m_max = item
19        if (0 < item) and (k_first == 1) and (k_second == 0):
20            k_second = 1
21            i_second, s_second = i, item
22        if (0 < item) and (k_first == 0):
23            k_first = 1
24            i_first, s_first = i, item
25
26    summ = 0
27    for item in s[i_first + 1:i_second]:
28        summ += item
29
30    s.sort(key=lambda x: x == 0)
31
32    print("Упорядоченный список(элементы = 0, после всех остальных): ", s)
33    print("Максимальный по модулю элемент списка = ", m_max)
34    if (k_first == 0) or (k_second == 0):
35        print("В списке нет двух положительных элементов.")
36    else:
37        print("Сумма элементов списка, расположенных между первым и вторым положительными элементами = ", summ)
```

```
C:\Users\n1c6\AppData\Local\Programs\Python\Python39\python.exe "C:\Users\n1c6\Desktop\Олег (кросс)\pycharm\5\ind
Список: 1 0 0 -1 -4 9 0 1 2
Упорядоченный список(элементы = 0, после всех остальных): [1.0, -1.0, -4.0, 9.0, 1.0, 2.0, 0.0, 0.0, 0.0]
Максимальный по модулю элемент списка = 9.0
Сумма элементов списка, расположенных между первым и вторым положительными элементами = -5.0

Process finished with exit code 0
```

Рисунок 4 – Окно программы для второй задачи и проверка кода на работоспособность.

## Ответы на контрольные вопросы:

1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов. В нем можно хранить объекты различных типов. Размер списка не статичен, его можно изменять. Список по своей природе является изменяемым типом данных. Переменная, определяемая как список, содержит ссылку на

структуру в памяти, которая в свою очередь хранит на какие-либо другие объекты или структуры.

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? 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 список?

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

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

Метод `append` можно использовать для добавления элемента в список.

Метод `insert` можно использовать, чтобы вставить элемент в список.

9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort`. Для сортировки списка в порядке убывания необходимо вызвать метод `sort` с аргументом `reverse=True`.

10. Как удалить один или несколько элементов из списка?

Удалить элемент можно, написав его индекс в методе `pop`. Если не указывать индекс, то функция удалит последний элемент. Элемент можно удалить с помощью метода `remove`. Оператор `del` можно использовать для тех же целей.

Можно удалить несколько элементов с помощью оператора среза.

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.

12. Какие существуют функции агрегации для работы со списками?

Для работы со списками Python предоставляет следующие функции:

1. `len(L)` - получить число элементов в списке `L`
2. `min(L)` - получить минимальный элемент списка `L`
3. `max(L)` - получить максимальный элемент списка `L`
4. `sum(L)` - получить сумму элементов списка `L`, если список `L`

содержит только числовые значения.

13. Как создать копию списка? `copy.copy(x)`

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

Функция `sorted()` в Python возвращает отсортированный список из элементов в итерируемом объекте. `list.sort()` на 13% быстрее, чем `sorted()`.

15. Самостоятельно изучите функцию `sorted` языка Python. В чем ее

отличие от метода `sort`

списков?

**Функция `sort()`** очень похожа на **`sorted ()`**, но в отличие от **`sorted`** она ничего не возвращает и не вносит изменений в исходную последовательность. Более того, **`sort()`** является **методом** класса **`list`** и может использоваться только со **списками**. Синтаксис: **`List_name.sort(key, reverse=False)`** Параметры: ключ: **Функция**, которая служит ключом для сравнения **сортировки**. реверс: Если **`true`**, то **список** сортируется в порядке убывания.

**Вывод:** приобрел навык по работе со списками при написании программ с помощью языка программирования Python.