

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:

Сотников Андрей Александрович  
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:

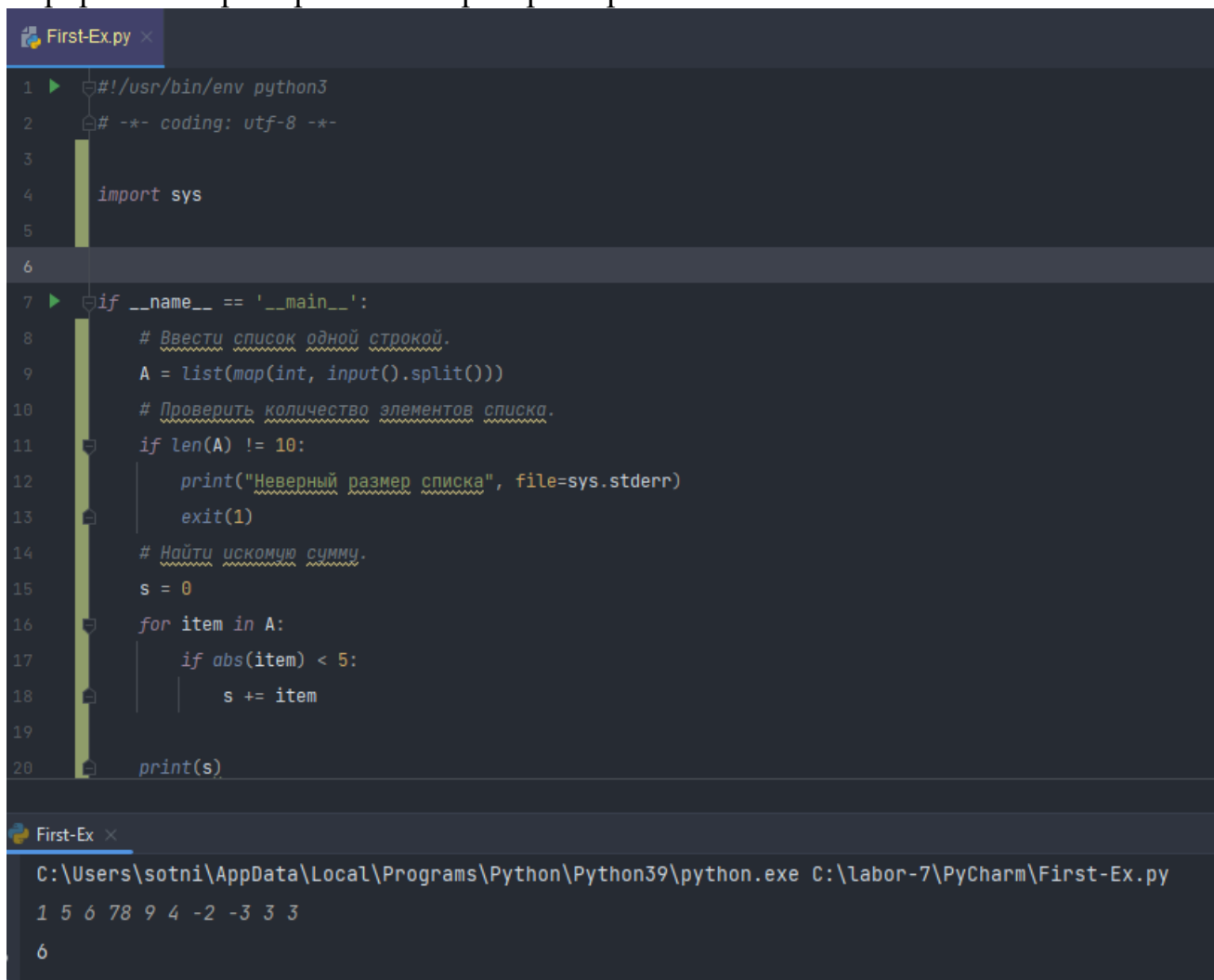
---

(подпись)

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

Ставрополь, 2022 г.

Проработка примеров из лабораторной работы:



The image shows a PyCharm IDE window with a Python script named 'First-Ex.py' and its execution output in a terminal window.

**Python Script (First-Ex.py):**

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

**Execution Output (Terminal):**

```
C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\labor-7\PyCharm\First-Ex.py
1 5 6 78 9 4 -2 -3 3 3
6
```

Рисунок 1 – Пример №1

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

if __name__ == '__main__' > if len(A) != 10
First-Ex x
C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\labor-7\PyCharm\First-Ex.py
1 5 4 78 9 6 -2 -3 3 3
6
```

Рисунок 2 – Пример №1 с использованием List Comprehensions

```
First-Ex.py × Second-Ex.py × Individual-First.py ×
1 # -*- coding: utf-8 -*-
2
3
4 import sys
5
6
7 if __name__ == '__main__':
8     # Ввести список одной строкой
9     a = list(map(int, input().split()))
10    # Если список пуст, завершить программу
11    if not a:
12        print("Заданный список пуст", file=sys.stderr)
13        exit(1)
14    # Определить индексы минимального и максимального элемента
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    if i_min > i_max:
24        i_min, i_max = i_max, i_min
25    count = 0
26    for item in a[i_min + 1:i_max]:
27        if item > 0:
28            count += 1
29    print(count)

```

if \_\_name\_\_ == '\_\_main\_\_' > for item in a[i\_min + 1:i\_max] > if item > 0

Second-Ex ×

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe

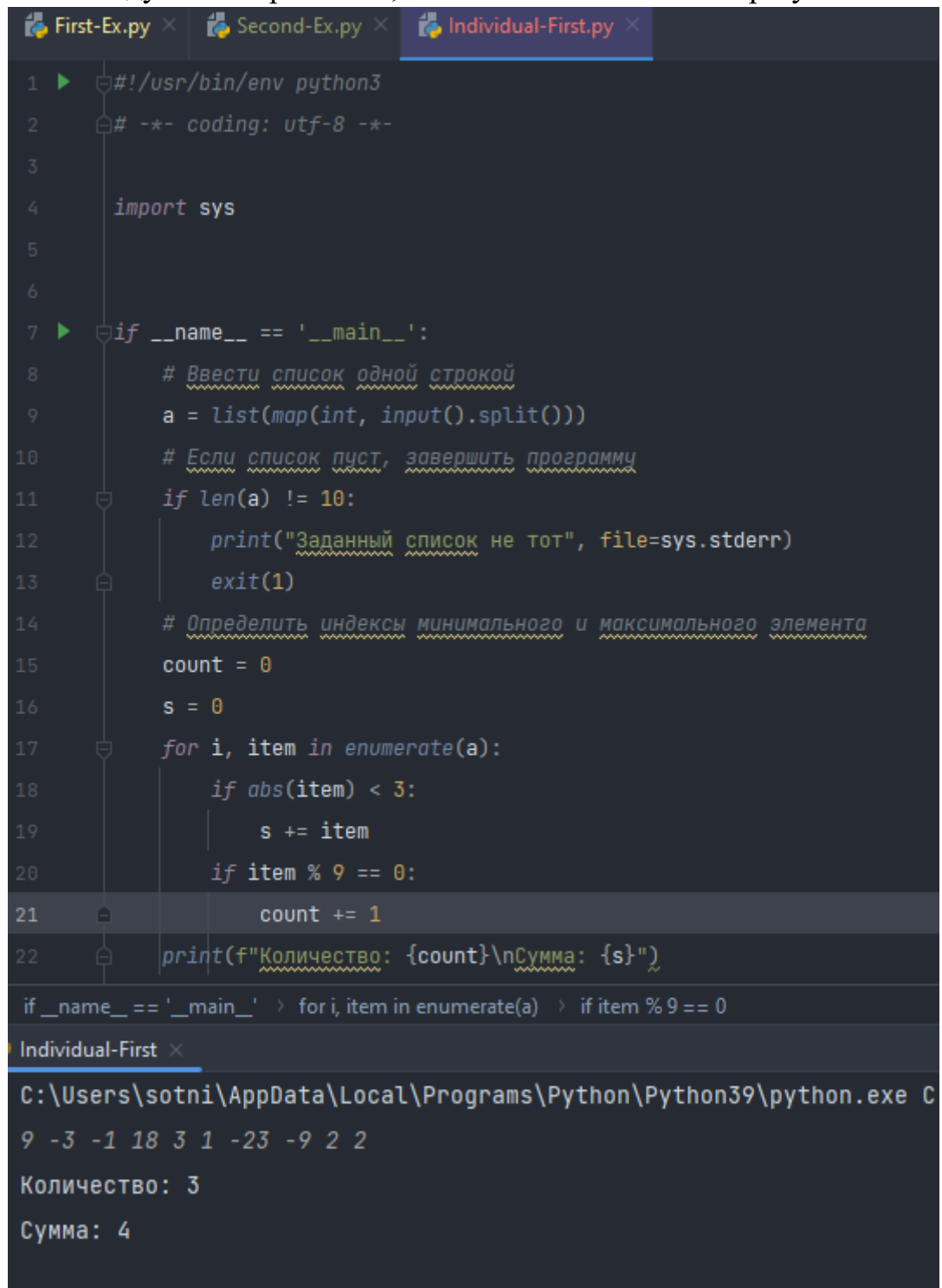
1 5 4 8 7 -4 -5 1 2 3 4 5 789 12 3 4 5

5

Рисунок 3 – Пример №2

## Индивидуальное задание:

Задание №1: Ввести список А из 10 элементов, найти сумму элементов, меньших по модулю 3 и кратных 9, их количество и вывести результаты на экран.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == '__main__':
8      # Ввести список одной строкой
9      a = list(map(int, input().split()))
10     # Если список пуст, завершить программу
11     if len(a) != 10:
12         print("Заданный список не тот", file=sys.stderr)
13         exit(1)
14     # Определить индексы минимального и максимального элемента
15     count = 0
16     s = 0
17     for i, item in enumerate(a):
18         if abs(item) < 3:
19             s += item
20         if item % 9 == 0:
21             count += 1
22     print(f"Количество: {count}\nСумма: {s}")
23
24 if __name__ == '__main__': > for i, item in enumerate(a) > if item % 9 == 0
```

Individual-First ×

C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C  
9 -3 -1 18 3 1 -23 -9 2 2  
Количество: 3  
Сумма: 4

Рисунок 4 – Индивидуальное задание №1

```
First-Ex.py × Second-Ex.py × Individual-First.py ×
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  ▶  if __name__ == '__main__':
8      # Ввести список одной строкой
9      a = list(map(int, input().split()))
10     # Если список пуст, завершить программу
11     if len(a) != 10:
12         print("Заданный список не тот", file=sys.stderr)
13         exit(1)
14     # Определить индексы минимального и максимального элемента
15     count = 0
16     s = sum([num for ind, num in enumerate(a) if abs(num) < 3])
17     count = len([num for ind, num in enumerate(a) if num % 9 == 0])
18     print(f"Количество: {count}\nСумма: {s}")
19
20 if __name__ == '__main__':
21     Individual-First ×
22     C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\l
23     0 5 3 2 -1 1 9 18 -18 2
24     Количество: 4
25     Сумма: 4
```

Рисунок 5 – Индивидуальное задание №1 с использованием List Comprehensions

Задание №2 В списке, состоящем из вещественных элементов, вычислить:

1. количество элементов списка, равных 0;
2. сумму элементов списка, расположенных после минимального элемента.

Упорядочить по возрастанию модулей элементов.

```
First-Ex.py × Second-Ex.py × Individual-First.py × Individual-Second.py ×

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

import sys

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

    # Если список пуст, завершить программу
    a_min = a[0]
    i_min = 0

    for i, item in enumerate(a):
        if item < a_min:
            i_min, a_min = i, item

    count = 0
    count = len([num for ind, num in enumerate(a) if num == 0])
    s = sum([num for ind, num in enumerate(a) if ind > i_min])

    print(f"Количество: {count}\nСумма: {s}\n")
    print(f"Осорттированный массив: \n{sorted(a, key=abs)}")

__name__ == '__main__'
```

Individual-Second ×

↑ C:\Users\sotni\AppData\Local\Programs\Python\Python39\python.exe C:\labor-7\PyChar

↓ 6 5 4 -1 0 -10 5 78 -5 3 0 41 0 -1

☰ Количество: 3

☒ Сумма: 121.0

☐ Осорттированный массив:

[0.0, 0.0, 0.0, -1.0, -1.0, 3.0, 4.0, 5.0, 5.0, -5.0, 6.0, -10.0, 41.0, 78.0]

Рисунок 6 – Индивидуальное задание №2





## 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:

```
my_list = [1, 2, 3, 4, 5]
```

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

Как уже было сказано выше, список является изменяемым типом данных. При его создании 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 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?

List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения списков. Проще всего работу list comprehensions показать на примере. Допустим вам

необходимо создать список целых чисел от 0 до n, где n предварительно задается. Классический способ решения данной задачи выглядел бы так:

```
>>> n = int(input())
7
>>> a=[]
>>> for i in range(n):
    a.append(i)

>>> print(a)
[0, 1, 2, 3, 4, 5, 6]
```

Использование *list comprehensions* позволяет сделать это значительно проще:

```
>>> n = int(input())
7
>>> a = [i for i in range(n)]
>>> print(a)
[0, 1, 2, 3, 4, 5, 6]
```

или вообще вот так, в случае если вам не нужно больше использовать *n*:

```
>>> a = [i for i in range(int(input()))]
7
>>> print(a)
[0, 1, 2, 3, 4, 5, 6]
```

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

Функция `sorted()` в Python, выполняет сортировку. Выполняет сортировку последовательности по возрастанию/убыванию.

Метод `sort()` работает только со списками и сортирует уже имеющийся список. Данный метод ничего не возвращает. А метод `sorted()` работает с любыми итерируемыми объектами и возвращает новый отсортированный список. В качестве итерируемых объектов могут выступать списки, строки, кортежи и другие.