

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение 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-о-21-1

Романов Платон Дмитриевич

«    » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Проверил: Доцент, к.т.н, доцент  
кафедры инфокоммуникаций

Воронкин А. В.

Работа защищена с оценкой: \_\_\_\_\_

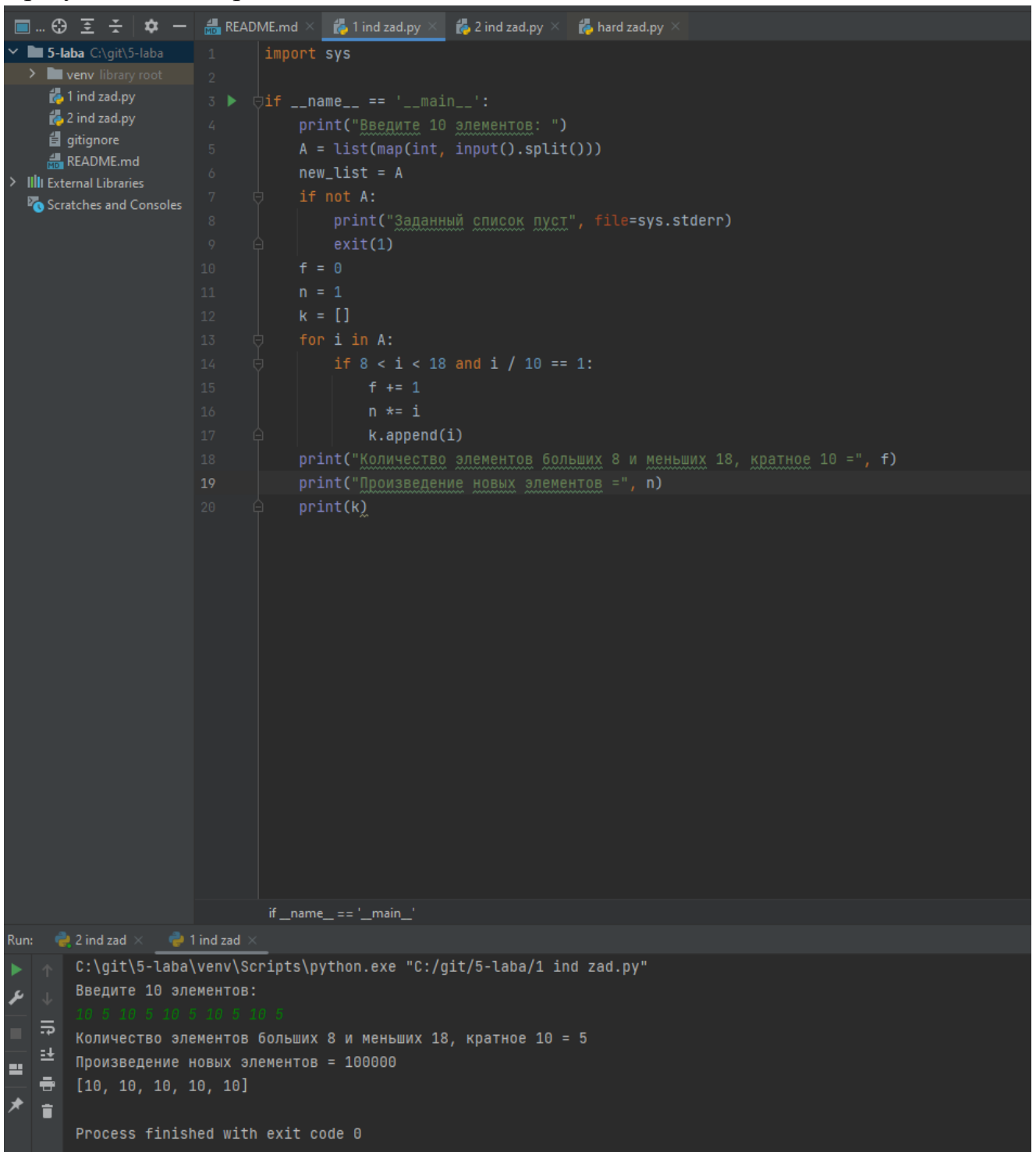
\_\_\_\_\_  
(подпись)

Ставрополь, 2022

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

Ссылка на репозиторий - <https://github.com/lesnaya1shelupon/5-laba>

**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.



```
1 import sys
2
3 if __name__ == '__main__':
4     print("Введите 10 элементов: ")
5     A = list(map(int, input().split()))
6     new_list = A
7     if not A:
8         print("Заданный список пуст", file=sys.stderr)
9         exit(1)
10    f = 0
11    n = 1
12    k = []
13    for i in A:
14        if 8 < i < 18 and i / 10 == 1:
15            f += 1
16            n *= i
17            k.append(i)
18    print("Количество элементов больших 8 и меньших 18, кратное 10 =", f)
19    print("Произведение новых элементов =", n)
20    print(k)
```

Run: 2 ind zad × 1 ind zad ×

```
C:\git\5-laba\venv\Scripts\python.exe "C:/git/5-laba/1 ind zad.py"
Введите 10 элементов:
10 5 10 5 10 5 10 5 10 5
Количество элементов больших 8 и меньших 18, кратное 10 = 5
Произведение новых элементов = 100000
[10, 10, 10, 10, 10]

Process finished with exit code 0
```

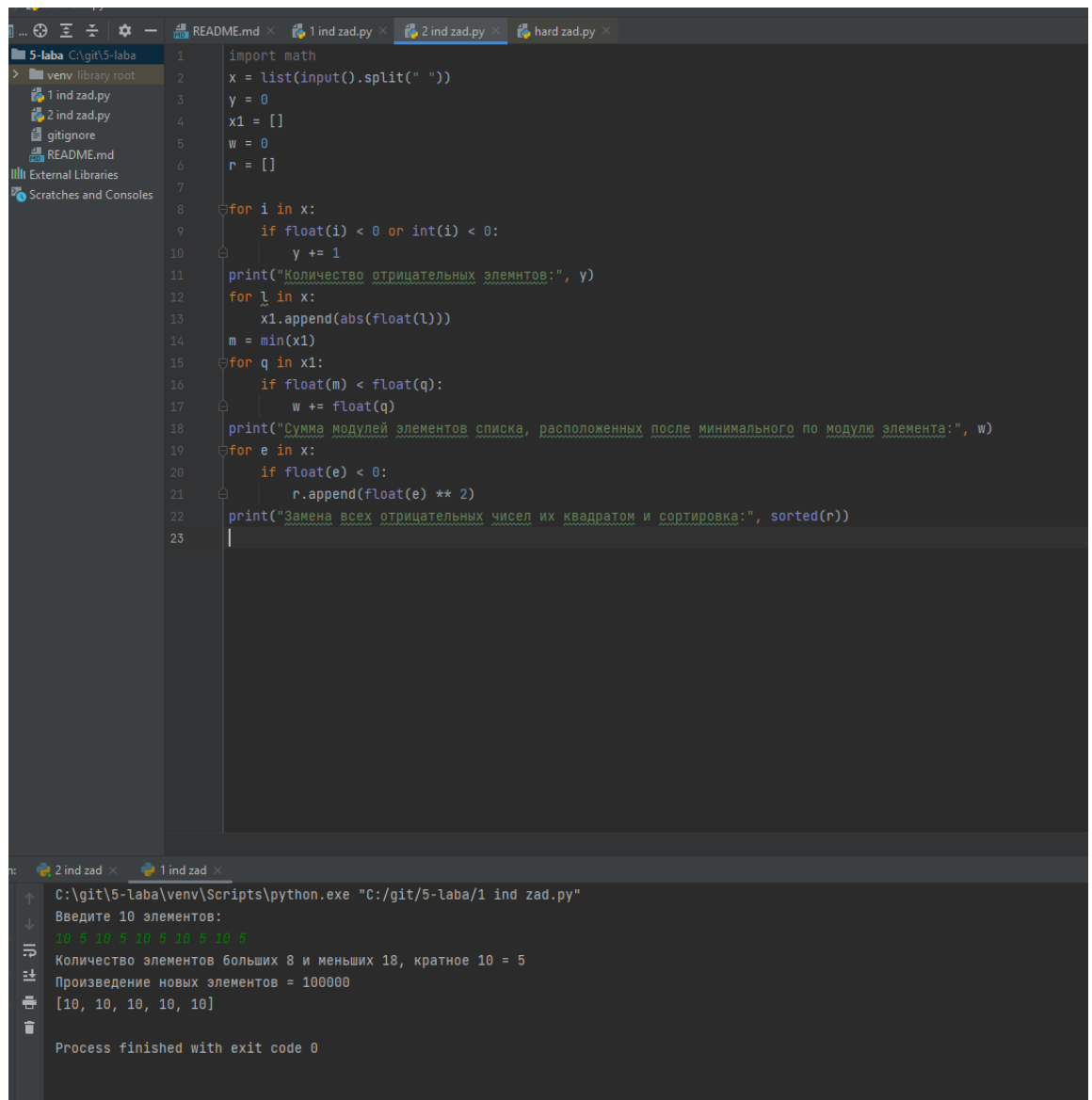
Рисунок 1. Код и окно вывода задачи 1.

**Задание 2.** В списке, состоящем из вещественных элементов, вычислить:

1) Количество отрицательных элементов списка;

му модулей элементов списка, расположенных после минимального по модулю элемента.

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



```
1 import math
2 x = list(input().split(" "))
3 y = 0
4 x1 = []
5 w = 0
6 r = []
7
8 for i in x:
9     if float(i) < 0 or int(i) < 0:
10         y += 1
11 print("Количество отрицательных элементов:", y)
12 for l in x:
13     x1.append(abs(float(l)))
14 m = min(x1)
15 for q in x1:
16     if float(m) < float(q):
17         w += float(q)
18 print("Сумма модулей элементов списка, расположенных после минимального по модулю элемента:", w)
19 for e in x:
20     if float(e) < 0:
21         r.append(float(e) ** 2)
22 print("Замена всех отрицательных чисел их квадратами и сортировка:", sorted(r))
23
```

C:\git\5-laba\venv\Scripts\python.exe "C:/git/5-laba/1 ind zad.py"

Введите 10 элементов:  
10 5 10 5 10 5 10 5 10 5

Количество элементов больших 8 и меньших 18, кратное 10 = 5

Произведение новых элементов = 100000

[10, 10, 10, 10, 10]

Process finished with exit code 0

Рисунок 2. Окно вывода задачи 2.

### 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 другие элементы данных в памяти. В отличие от таких типов данных 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 `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:** приобрели навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.