

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЁТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.4**  
**дисциплины**  
**«Основы кроссплатформенного программирования»**

Выполнил:  
Костукайло Кирилл Николаевич  
1 курс, группа ИВТ-б-о-21-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной техники и  
автоматизированных систем», очная  
форма обучения

---

(подпись)

Проверил:  
Кафедры инфокоммуникаций, старший  
преподаватель  
Воронкин Р.А.

---

(подпись)

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

Ставрополь, 2022 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.

### Ход работы:

1. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.
3. Выполнил клонирование созданного репозитория.
4. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.
5. Организовал свой репозиторий в соответствие с моделью ветвления git-flow.
6. Создал проект PyCharm в папке репозитория.
7. Проработал примеры лабораторной работы. Создал для каждого примера отдельный модуль языка Python. Зафиксировал изменения в репозитории.

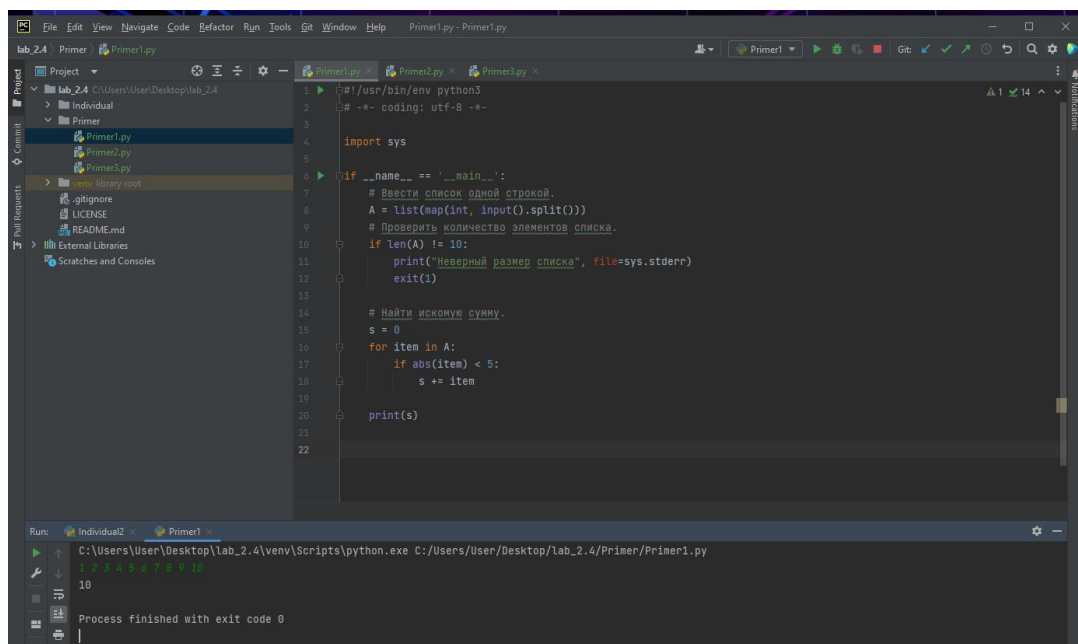


Рисунок 1.1. Пример 1

```
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 = sum([a for a in A if abs(a) < 5])
15     print(s)
16
17 if __name__ == '__main__':
```

Run: Primer2

C:\Users\User\Desktop\lab\_2.4\venv\Scripts\python.exe C:\Users\User\Desktop\lab\_2.4\Primer\Primer2.py

25 1 4 7 2 0 4 4 3 12

15

Process finished with exit code 0

Рисунок 1.2. Пример 2

```
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     # Если список пуст, завершить программу.
9     if not a:
10         print("Заданный список пуст", file=sys.stderr)
11         exit(1)
12
13     # Определить индексы минимального и максимального элементов.
14     a_min = a_max = a[0]
15     i_min = i_max = 0
16
17     for i, item in enumerate(a):
18         if item < a_min:
19             i_min, a_min = i, item
20
21         if item >= a_max:
22             i_max, a_max = i, item
23
24     # Проверить индексы и поменять их местами.
25     if i_min < i_max:
26         for i, item in enumerate(a):
```

Run: Primer3

C:\Users\User\Desktop\lab\_2.4\venv\Scripts\python.exe C:\Users\User\Desktop\lab\_2.4\Primer\Primer3.py

32 2 45 2 7 52 47 14 76 2 4 7 2

0

Process finished with exit code 0

Рисунок 1.3. Пример 3

8. Выполните индивидуальные задания, согласно своего варианта.

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 В заданном списке подсчитать число нулевых элементов и вывести на экран их индексы.
5
6
7 import sys
8
9 if __name__ == '__main__':
10     A = list(map(int, input().split()))
11     if not A:
12         print("Заданный список пуст.", file=sys.stderr)
13         exit(1)
14
15     k = 0
16     for index, i in enumerate(A):
17         if i == 0:
18             print(index)
19
20         if i != 0:
21             k = k + 1
22
23     if k > 0:
24         print("Число нулевых элементов равно:", k)
25
26 if __name__ == '__main__': for index, i in enumerate(A): if i == 0
```

Run: Individual1

C:\Users\User\Desktop\lab\_2.4\venv\Scripts\python.exe C:/Users/User/Desktop/lab\_2.4/Individual/Individual1.py

24 0 1 0 44 1 33 0 64 3 0 45 91 0 2

3

7

10

13

Число нулевых элементов равно: 4

Process finished with exit code 0

Рисунок 2.1. Индивидуальное задание 1

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 В списке, состоящем из вещественных элементов, вычислить:
5 1. максимальный по модулю элемент списка;
6 2. сумму элементов списка, расположенных между первым и вторым положительными элементами.
7 Преобразовать список таким образом, чтобы элементы, равные нулю, располагались после всех
8 остальных.
9
10
11 import sys
12
13 if __name__ == '__main__':
14     A = list(map(int, input().split()))
15     if not A:
16         print("Заданный список пуст.", file=sys.stderr)
17         exit(1)
18
19     k = 0
20     for index, i in enumerate(A):
21         if abs(i) > abs(k):
22             k = i
23
24     print(k)
25
26 if __name__ == '__main__':
```

Run: Individual2

C:\Users\User\Desktop\lab\_2.4\venv\Scripts\python.exe C:/Users/User/Desktop/lab\_2.4/Individual/Individual2.py

21 -2 5 26 -43 23 3 0 34 0 10 0 2 -24 2 53 0 43

53

91

[21, -2, 5, 26, -43, 23, 3, 34, 12, 2, -24, 2, 53, 11, 0, 0, 0]

Process finished with exit code 0

Рисунок 2.2. Индивидуальное задание 2

9. Зафиксируйте сделанные изменения в репозитории.

Ответ 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 ссылки на какие-либо другие объекты или структуры. Списки в Python - упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив, но типы могут отличаться).

## 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?

Перебор элементов списка состоит в том, что мы в цикле просматриваем все элементы этого списка. Читать элементы списка можно с помощью следующего цикла: `my_list = ['один', 'два', 'три', 'четыре', 'пять']` `for elem in my_list: print(elem)` Перебор элементов списка состоит в том, что мы в цикле просматриваем все элементы списка и, если нужно, выполняем с каждым из них некоторую операцию. Переменная цикла изменяется от 0 до N-1, где N – количество элементов списка, то есть в диапазоне `range(N)`: `for i in range(N): A[i] += 1` в этом примере все элементы списка A увеличиваются на 1. Если список изменять не нужно, для перебора его элементов удобнее всего использовать такой цикл: `for x in A:`

## 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`. `lst = ['test', 'twest', 'twest', 'trest']` `'test' in lst` # Вывод: True `'toast' in lst` # Вывод: False

## 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()`, в порядке возрастания будет(`list1.sort()`). Для сортировки списка в порядке убывания необходимо вызвать метод `sort` с аргументом `reverse=True`(`list1.reverse()`).

#### 10. Как удалить один или несколько элементов из списка?

Удалить элемент можно, написав его индекс (на основе его индекса) в методе `pop(list.pop(index))`. Если не указывать индекс, то функция удалит последний элемент. Элемент можно удалить с помощью метода `remove` (пишется `my_list.remove(100)`). Оператор `del` можно использовать для тех же целей `del list[index]`. Можно удалить несколько элементов с помощью оператора среза. Можно удалить все элементы из списка с помощью метода `clear` (`list.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. Как осуществляется доступ к элементам списков с помощью срезов?

Со списками, так же как и со строками, можно делать срезы. А именно: `A[i:j]` срез из `j-i` элементов `A[i]`, `A[i+1]`, ..., `A[j-1]`. `A[i:j-1]` срез из `i-j`

элементов  $A[i]$ ,  $A[i-1]$ , ...,  $A[j+1]$  (то есть меняется порядок элементов).  $A[i:j:k]$  срез с шагом  $k$ :  $A[i]$ ,  $A[i+k]$ ,  $A[i+2*k]$ ,... . Если значение  $k < 0$ , то элементы идут в противоположном порядке. Каждое из чисел  $i$  или  $j$  может отсутствовать, что означает “начало строки” или “конец строки” Списки, в отличие от строк, являются изменяемыми объектами: можно отдельному элементу списка присвоить новое значение. Но можно менять и целиком срезы.

### 13. Какие существуют функции агрегации для работы со списками?

Для работы со списками Python предоставляет следующие функции:

$\text{len}(L)$  - получить число элементов в списке  $L$        $\text{min}(L)$  - получить минимальный элемент списка  $L$        $\text{max}(L)$  - получить максимальный элемент списка  $L$        $\text{sum}(L)$  - получить сумму элементов списка  $L$ , если список  $L$  содержит только числовые значения.

### 14. Как создать копию списка?

Операция присваивания не копирует объект, он лишь создаёт ссылку на объект. Для изменяемых коллекций, или для коллекций, содержащих изменяемые элементы, часто необходима такая копия, чтобы её можно было изменить, не изменяя оригинал. Данный модуль предоставляет общие (поверхностная и глубокая) операции копирования. `Spisok = copy.copy(oldspisok)`

### 15. 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()`. Ещё одно отличие заключается в том, что метод `list.sort()` определён только для списков, в то время как `sorted()` работает со всеми итерируемыми объектами.

Вывод: приобрёл навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

