

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

Институт цифрового развития

Кафедра инфокоммуникаций

Дисциплина: «основы кроссплатформенного программирования»

**ОТЧЕТ**

**по лабораторной работе №5**

Выполнил: студент 1 курса группы ИТС-21-1

Снадный Михаил Сергеевич

Проверил: к.ф.-м.н., доцент кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой: \_\_\_\_\_

Ставрополь, 2021

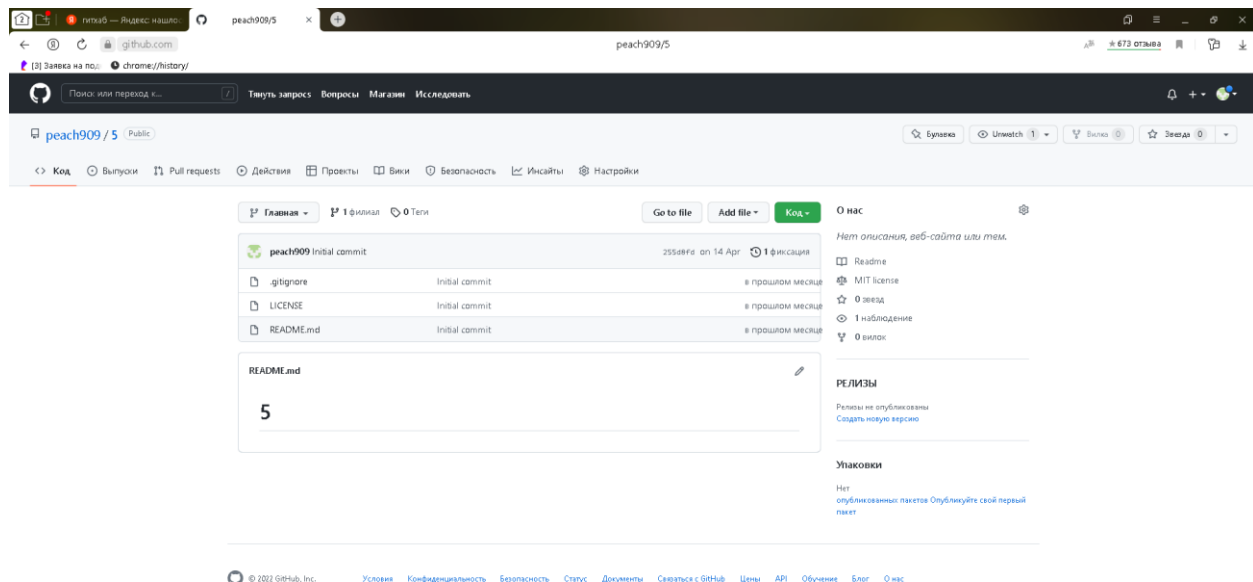
## Тема:

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.

Создал общедоступный репозиторий



написал код для примера 1

```
*IDLE Shell 3.10.4*
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = [float(i) for i in input().split()]
... c = [1, 4, 5, 8, 9, 3]
... for i in a:
...     if i > 0:
...         c.append(i)
... n = c[0]
... for i in range(1, len(c)):
...     n -= c[i]
... print(n)
```

Написал код для примера 2

```
*IDLE Shell 3.10.4*
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from random import randint
... cnt = 0
... values = [randint(-50, 50) for _ in range(10)]
... min_index = values.index(min(values))
... max_index = values.index(max(values))
... if max_index < min_index: min_index, max_index = max_index, min_index
... for i in range(min_index+1, max_index):
...     if values[i] > 0: cnt += 1
... print(values)
... print('Количество положительных значений между минимальным и максимальным значениями равно:', cnt)
```

Написал код для индивидуального задания 1

```
*IDLE Shell 3.10.4*
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> S = [(u + d + v) / 3 for u, d, v in zip(U, D, V)]
... print(S, sum(S) / len(S))
... |
```

Написал код для индивидуального задания 2

```
*IDLE Shell 3.10.4*
File Edit Shell Debug Options Window Help
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> if __name__ == '__main__':
...     a = []
...     s = 0
...     p = 1
...     n = int(input())
...     for i in range(n):
...         element = float(input())
...         a.append(element)
...         a.sort(reverse=True)
...         for el in a:
...             p *= el
...             if el < 0:
...                 s += el
...     print("Сумма отрицательных элементов списка = ", s)
...     print("Произведение элементов списка, расположенных между максимальным и минимальным элементами = ", p)
...     print("Отсортированный по возрастанию список: ", a)
```

## Контрольные вопросы

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

Ответ: Список (*list*) – это структура данных для хранения объектов 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.

2. Как осуществляется создание списка в Python?

Ответ: Для создания списка нужно заключить элементы в квадратные скобки:

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

3. Как организовано хранение списков в оперативной памяти?

Ответ: 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?

Ответ: 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?

Ответ: Для объединения списков можно использовать оператор сложения (+).

Список можно повторить с помощью оператора умножения (\*):

6. Как проверить есть ли элемент в списке?

Ответ: Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор `in` :

7. Как определить число вхождений заданного элемента в списке?

Ответ: Метод `count` можно использовать для определения числа сколько раз данный элемент встречается в списке:

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

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

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

9. Как выполнить сортировку списка?

Ответ: 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?

Ответ: 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:

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

Ответ: *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?

Ответ: Слайсы (срезы) являются очень мощной составляющей *Python*, которая позволяет быстро и лаконично решать задачи выборки элементов из списка.

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?

Ответ: 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.

14. Как создать копию списка?

Ответ: Воспользоваться командой `copy.copy(x)`

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

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

### **Вывод:**

В ходе работы мы изучили и приобрели навыки по работе со списками при написании программ на языке Python (3).

