

Северо-Кавказский федеральный университет
Кафедра инфокоммуникаций СКФУ

Отчет
По лабораторной работе №5
По предмету: «Основы
кроссплатформенного программирования»

Исполнитель:

Студента группы ИТС-б-з-22-1

Направление подготовки 11.03.02

Инфокоммуникационные

технологии и системы связи

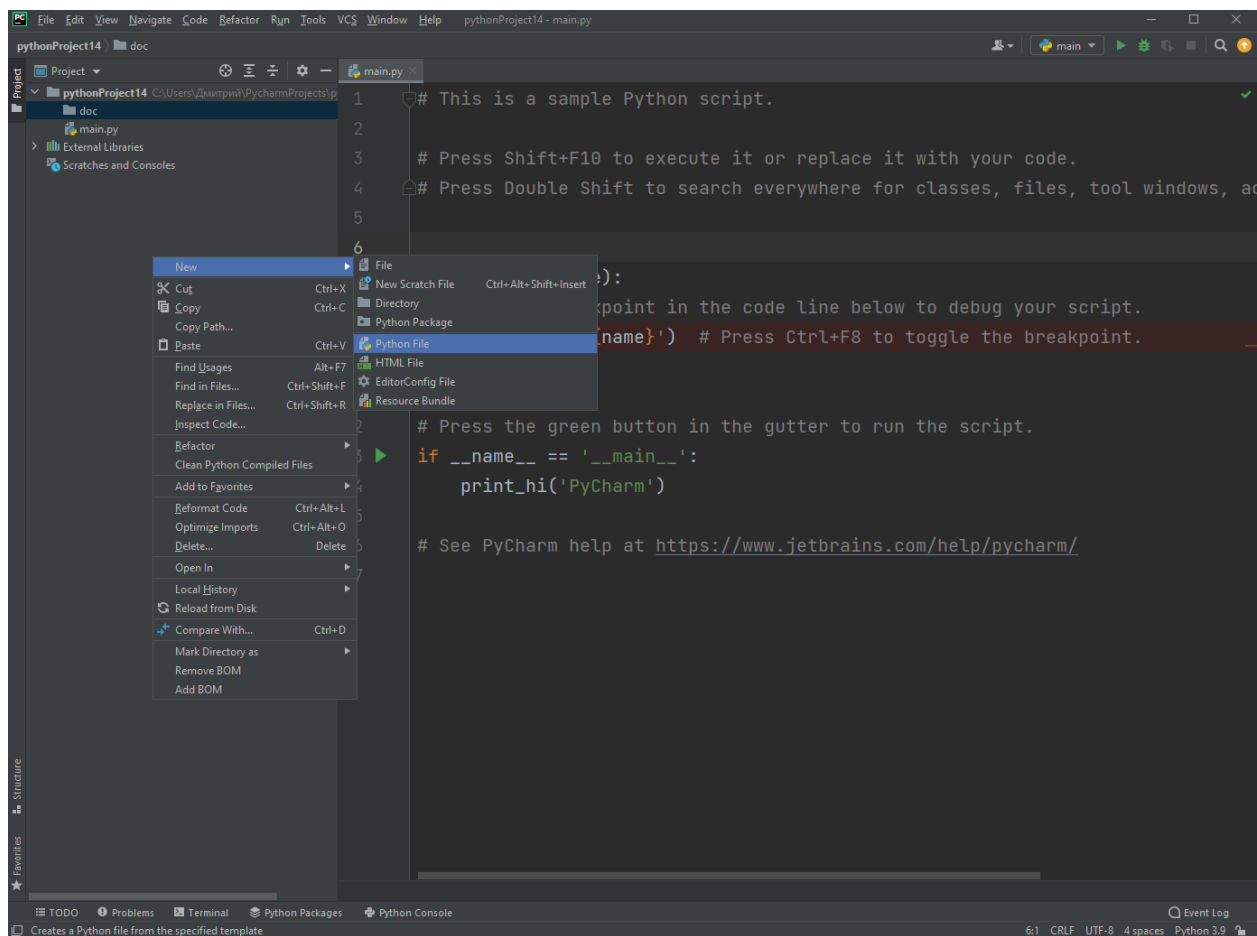
Пальников Станислав Петрович
(Ф.И.О.)

Руководитель дисциплины:

Воронкин Роман Александрович

Ставрополь, 2023

Ход выполнения работы:



```
1 import sys
2
3 if __name__ == '__main__':
4     k=0
5     s=0
6     A = [int(input()) for i in range(10)]
7     for i in range(len(A)):
8         if A[i]<=0 and A[i]%7==0:
9             s+=A[i]
10            k+=1
11    print(s,k)
12
13 if __name__ == '__main__':
```

Run: modul1

C:\Users\Дмитрий\AppData\Local\Programs\Python\Python39\python.exe
C:/Users/Дмитрий/PycharmProjects/pythonProject14/modul1.py

1
-7
7
-7
-14
8
4
6
4
8
-28 3

Run | TODO | Problems | Terminal | Python Packages | Python Console | Event Log

PEP 8: W292 no newline at end of file | 15:1 | CRLF | UTF-8 | 4 spaces | Python 3.9

```
1 import sys
2
3 if __name__ == '__main__':
4     k=0
5     s=0
6     A = [int(input()) for i in range(10)]
7     for i in range(len(A)):
8         if A[i]<=0 and A[i]%7==0:
9             s+=A[i]
10            k+=1
11    print(s,k)
12
13 if __name__ == '__main__':
```

Run: modul1

C:\Users\Дмитрий\AppData\Local\Programs\Python\Python39\python.exe
C:/Users/Дмитрий/PycharmProjects/pythonProject14/modul1.py

1
2
3
4
5
7
8
9
10
1
0 0

Run | TODO | Problems | Terminal | Python Packages | Python Console | Event Log

15:1 | CRLF | UTF-8 | 4 spaces | Python 3.9

```
1 from random import uniform
2 import sys
3
4 if __name__ == '__main__':
5     maxx = 0
6     maxx_i = 0
7     poli = 0
8     s = 0
9     A = [round(uniform(-10, 10), 2) for i in range(10)]
10    a = int(input())
11    b = int(input())
12
13    for i in range(len(A)):
14        if abs(A[i]) > maxx:
15            maxx = abs(A[i])
16            maxx_i = i
17        if A[i] > 0:
18            poli = i
19    for i in range(poli, len(A)):
20        s += A[i]
21    print('максимальное по модулю ', maxx, 'сумма после первого', s)
22    print('массив', A)
23    print('отсортированный массив: ', sorted(A, key=lambda x: int(x) not in range(a, b + 1)))
```

Run

C:\Users\Дмитрий\AppData\Local\Programs\Python\Python39\python.exe C:/Users/Дмитрий/PycharmProjects/pythonProject14/modul2.py

максимальное по модулю 8.81 сумма после первого 8.81
массив [-8.2, 8.57, 7.72, -2.82, -1.86, -0.82, -4.56, -1.2, -8.96, 8.81]
отсортированный массив: [-8.2, 8.57, 7.72, -2.82, -1.86, -0.82, -4.56, -1.2, -8.96, 8.81]

Process finished with exit code 0

```
1 from random import uniform
2 import sys
3
4 if __name__ == '__main__':
5     maxx = 0
6     maxx_i = 0
7     poli = 0
8     s = 0
9     A = [round(uniform(-10, 10), 2) for i in range(10)]
10    a = int(input())
11    b = int(input())
12
13    for i in range(len(A)):
14        if abs(A[i]) > maxx:
15            maxx = abs(A[i])
16            maxx_i = i
17        if A[i] > 0:
18            poli = i
19    for i in range(poli, len(A)):
20        s += A[i]
21    print('максимальное по модулю ', maxx, 'сумма после первого', s)
22    print('массив', A)
23    print('отсортированный массив: ', sorted(A, key=lambda x: int(x) not in range(a, b + 1)))
```

Run

C:\Users\Дмитрий\AppData\Local\Programs\Python\Python39\python.exe C:/Users/Дмитрий/PycharmProjects/pythonProject14/modul2.py

максимальное по модулю 0 сумма после первого -2.3999999999999995
массив [9.99, 0.87, -5.21, -6.03, 5.19, 6.6, 7.67, 6.85, 5.49, -7.89]
отсортированный массив: [9.99, 6.6, 7.67, 6.85, 0.87, -5.21, -6.03, 5.19, 5.49, -7.89]

Process finished with exit code 0

Ответы на 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]
```

Список может выглядеть так:

```
my_list = ['один', 'два', 'три', 'четыре', 'пять']
```

Можно смешивать типы содержимого:

```
my_list = ['один', 10, 2.25, [5, 15], 'пять']
```

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?

Функция `len()` используется для возврата количества элементов.

Функция `enumerate()` применяется для итерируемых коллекций (строки, списки, словари и т. д.) и создает объект, который генерирует кортежи, состоящие из двух элементов - индекса элемента и самого элемента.

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

```
if 3 in lst:
```

```
    print("Список содержит число 3")
```

```
else:
```

```
    print("Список не содержит число 3")
```

Результат:

Список содержит число 3

7. Как определить число вхождений заданного элемента в списке?

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

```
lst = [1, 2, 2, 3, 3]
print(lst.count(2))
```

Результат 2 .

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)
```

Результат:

```
[1, 'Привет', 2, 3, 4, 5]
```

Метод append можно использовать для добавления элемента в список:

```
my_list = ['один', 'два', 'три', 'четыре', 'пять']
my_list.append('ещё один')
print(my_list)
```

Результат:

```
['один', 'два', 'три', 'четыре', 'пять', 'ещё один']
```

9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод sort.

```
my_list = ['cde', 'fgh', 'abc', 'klm', 'opq']
list_2 = [3, 5, 2, 4, 1]
my_list.sort()
list_2.sort()
```

```
print(my_list)
```

```
print(list_2)
```

Вывод:

```
['abc', 'cde', 'fgh', 'klm', 'opq']
```

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

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

```
list_2 = [3, 5, 2, 4, 1]
```

```
list_2.sort(reverse=True)
```

Вывод:

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

10. Как удалить один или несколько элементов из списка?

Можно удалить несколько элементов с помощью оператора среза:

```
my_list = ['один', 'два', 'три', 'четыре', 'пять']
```

```
del my_list[1:3]
```

```
print(my_list)
```

Результат:

```
['один', 'четыре', 'пять']
```

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

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

```
>>> a = [i for i in range(10)]
```

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?

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?

Для создания копии списка необходимо использовать либо метод copy, либо использовать оператор среза.

```
>>> a = [1, 2, 3, 4, 5]
```

```
>>> b = a.copy()
```

```
>>> b [1, 2, 3, 4, 5]
```

```
>>> a == b
```

```
True
```

```
>>> a is b
```

```
False
```

```
>>> a is not b
```

```
True
```

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

Функция sorted () - это встроенная функция в Python, которая возвращает отсортированную последовательность (список, кортеж, строку) или отсортированную коллекцию (наборы, словарь) в виде

списка. Функция `sorted()` не влияет на исходную итеративную последовательность, поскольку обеспечивает новый отсортированный вывод.

Разница только в том, что метод `sort` не создает новой коллекции, а меняет уже существующую. Функция же `sorted` не меняет исходную коллекцию, а создает новую с отсортированными элементами.