

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.4
дисциплины «Основы кроссплатформенного программирования»
Вариант ____

Выполнил:
Дмитриченко Марат Заурович
1 курс, группа ИТС-б-0-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети», очная форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

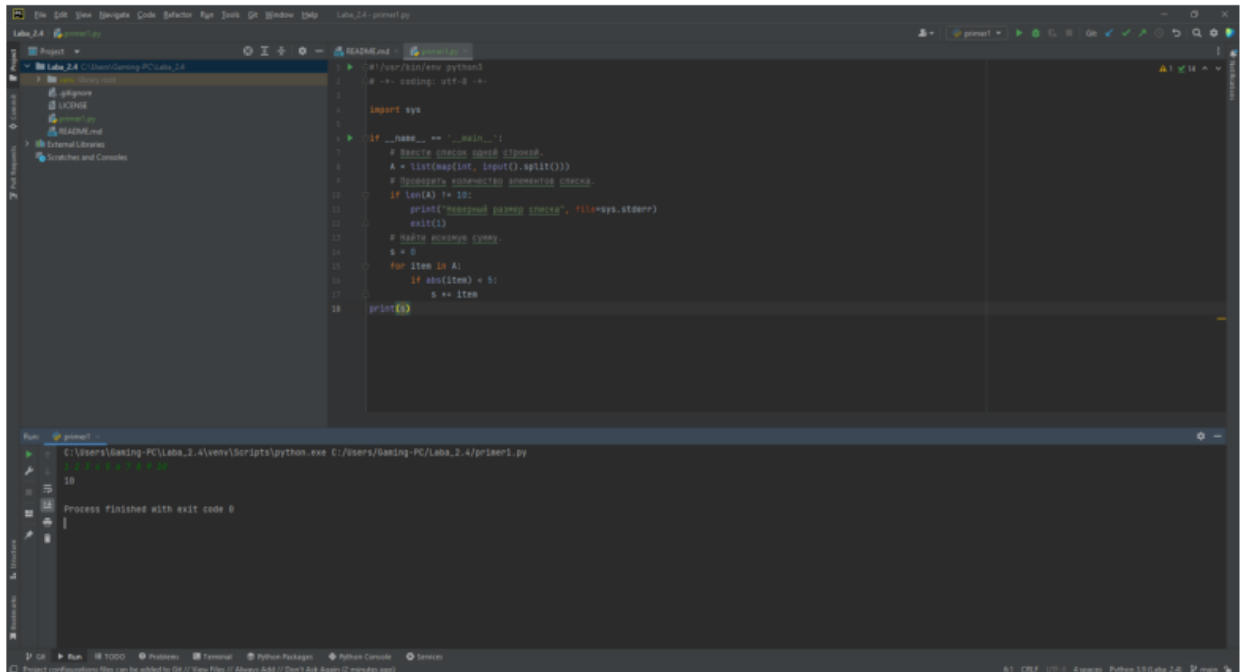
Ставрополь, 2023 г.

Тема: Работа со списками в языке Python

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

Ход работы

Условие примера: ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 if __name__ == '__main__':
7     # Введите список из 10 элементов
8     A = list(map(int, input().split()))
9     # Проверьте соответствие условиям задачи
10     if len(A) != 10:
11         print("Неверное количество элементов")
12         exit(1)
13     # Находим сумму
14     s = 0
15     for item in A:
16         if abs(item) < 5:
17             s += item
18     print(s)
```

Run: python3

C:\Users\Basing-PC\Labo.2.4\env\scripts\python.exe C:/Users/Basing-PC/Labo.2.4/primer1.py

10

Process finished with exit code 0

Рисунок 1. Реализация первого примера

Условие примера: написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

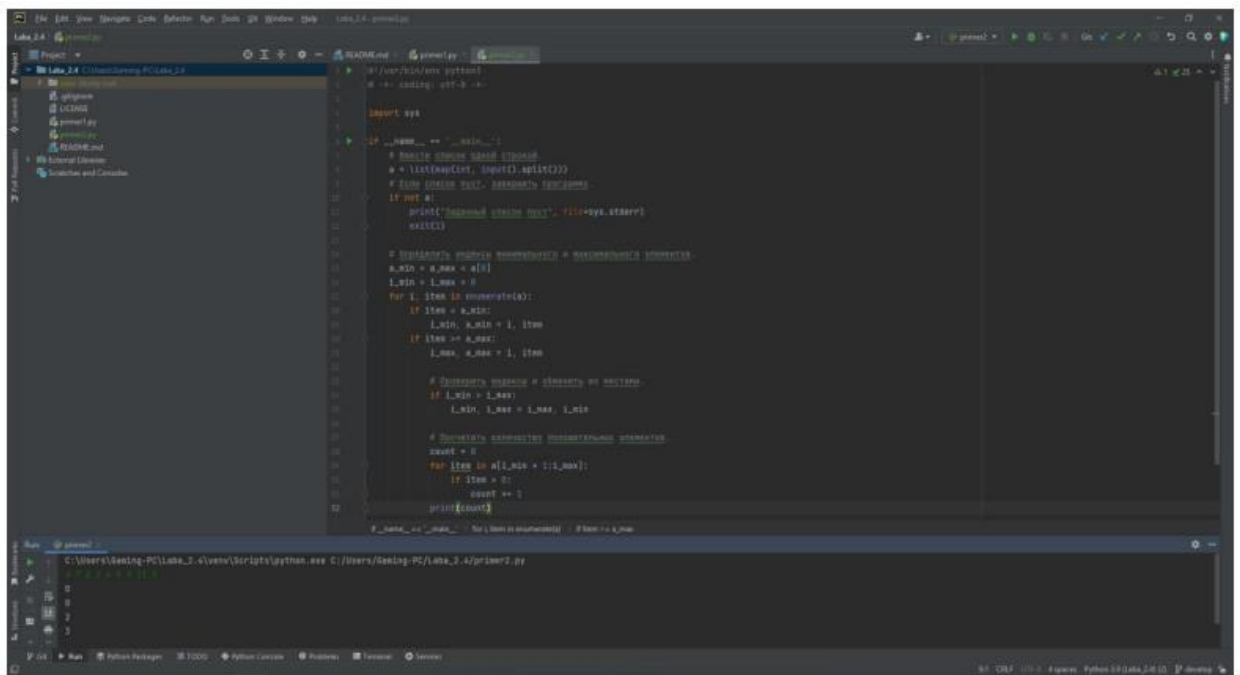


Рисунок 2. Реализация второго примера

Индивидуальное задание

Условие примера: ввести список А из 10 элементов, найти сумму элементов, больших 3 и меньших 8 и вывести ее на экран.

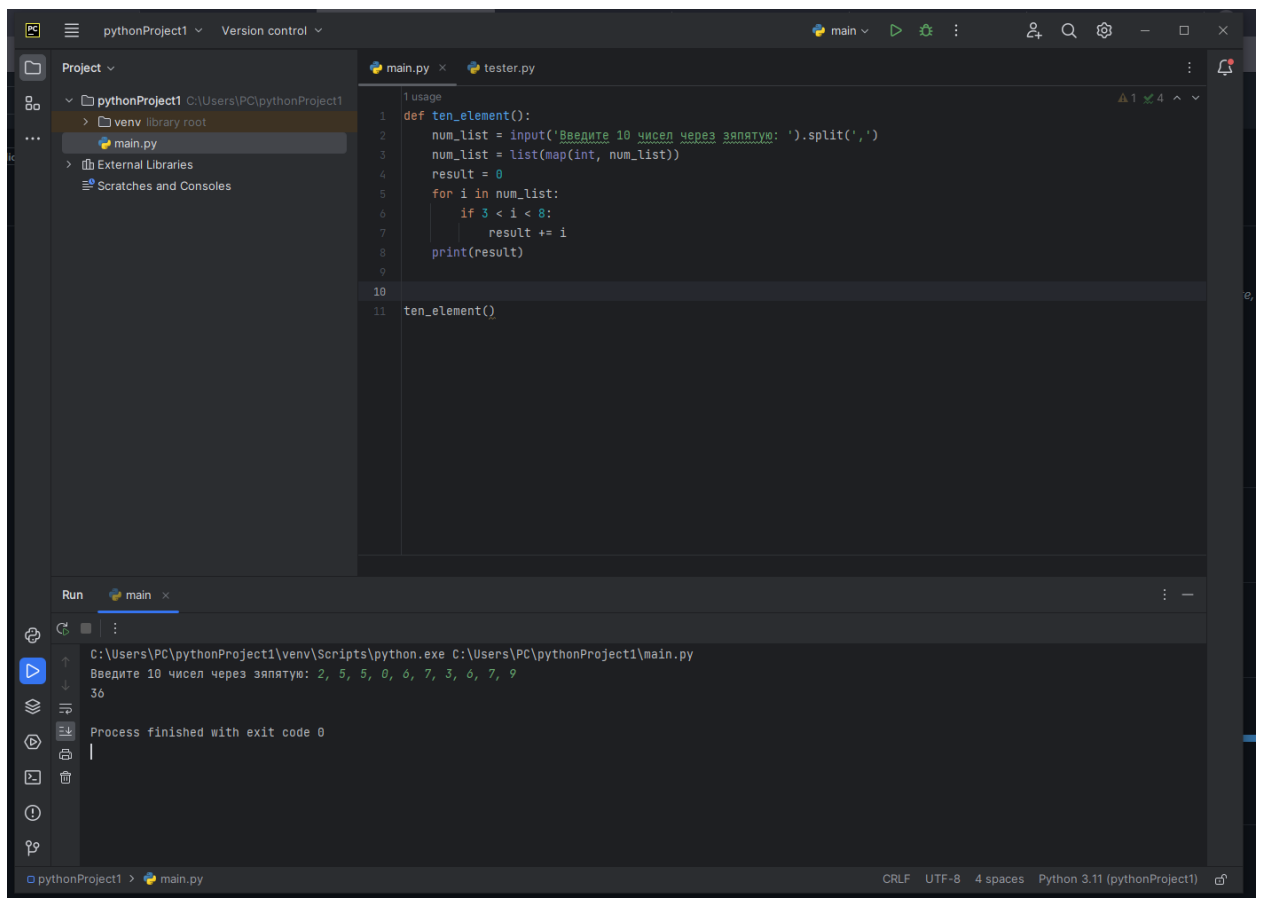


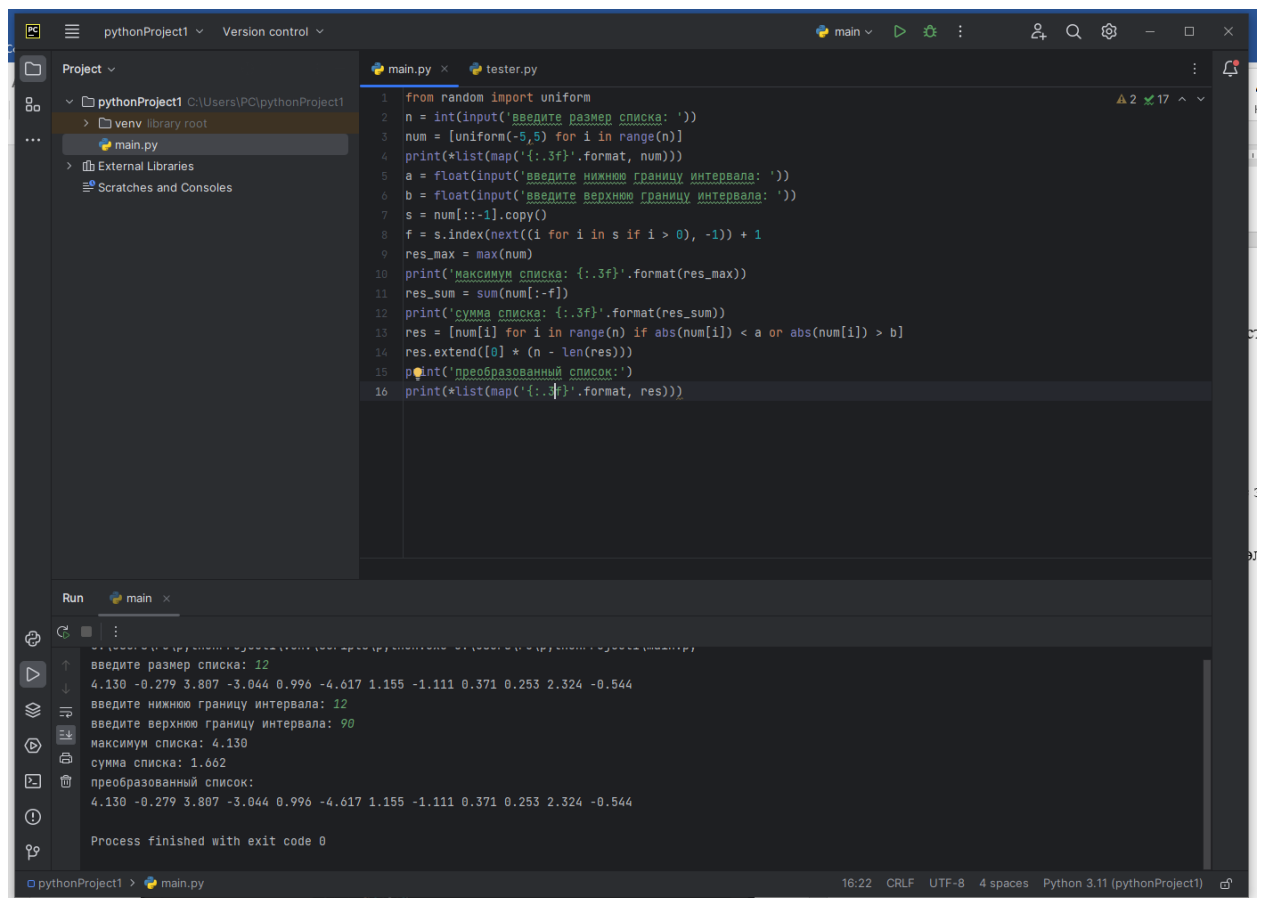
Рисунок 3. Реализация первого примера

Условие примера: в списке, состоящем из вещественных элементов, вычислить:

1. максимальный элемент списка;
2. сумму элементов списка, расположенных до последнего положительного элемента.

Сжать список, удалив из него все элементы, модуль которых находится в интервале $[a, b]$.

Освободившиеся в конце списка элементы заполнить нулями.



```
1 from random import uniform
2 n = int(input('введите размер списка: '))
3 num = [uniform(-5,5) for i in range(n)]
4 print(*list(map('{:.3f}'.format, num)))
5 a = float(input('введите нижнюю границу интервала: '))
6 b = float(input('введите верхнюю границу интервала: '))
7 s = num[::-1].copy()
8 f = s.index(next((i for i in s if i > 0), -1)) + 1
9 res_max = max(num)
10 print('максимум списка: {:.3f}'.format(res_max))
11 res_sum = sum(num[:f])
12 print('сумма списка: {:.3f}'.format(res_sum))
13 res = [num[i] for i in range(n) if abs(num[i]) < a or abs(num[i]) > b]
14 res.extend([0] * (n - len(res)))
15 print('преобразованный список:')
16 print(*list(map('{:.3f}'.format, res)))
```

Run main

Process finished with exit code 0

Рисунок 4. Реализация второго примера

Ответы на контрольные вопросы

1. Что такое списки в языке Python?

Список - это один из встроенных типов данных в языке Python, представляющий собой упорядоченный изменяемый набор объектов произвольных типов. В нем можно хранить объекты различных типов. 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?

Списки в Python хранятся в оперативной памяти в виде последовательности элементов, каждый из которых может быть любого типа. При создании списка выделяется некоторое количество памяти, которое может быть увеличено или уменьшено в зависимости от изменения размера списка.

4. Каким образом можно перебрать все элементы списка?

Для перебора всех элементов списка можно использовать цикл `for` или метод `for in`:

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?

Чтобы проверить, содержится ли элемент в списке, можно использовать оператор `in`.

7. Как определить число вхождений заданного элемента в списке?

Чтобы определить число вхождений заданного элемента в списке, можно использовать метод `count()`

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

Чтобы добавить элемент в список, можно использовать метод `append()`
Чтобы вставить элемент в список на заданную позицию, можно использовать метод `insert()`

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?

Для удаления одного элемента из списка можно использовать метод `remove()`, указав в скобках значение элемента, который нужно удалить. Удалить элемент можно, написав его индекс в методе `pop`. Если не указывать индекс, то функция удалит последний элемент.

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?

Доступ к элементам списка с помощью срезов осуществляется с помощью квадратных скобок `[]`. Срезы задаются в виде `start:stop:step`, где `start` - индекс первого элемента в срезе (включительно), `stop` - индекс последнего элемента в срезе (не включительно), а `step` - шаг, с которым нужно выбирать элементы из списка.

13. Какие существуют функции агрегации для работы со списками?

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.

14. Как создать копию списка?

`copy.copy(x)`

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

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