

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникации

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №2.4

Дисциплина: «Основы кроссплатформенного программирования»

Тема: «Работа со в языке Python3»

Выполнил: студент 1 курса

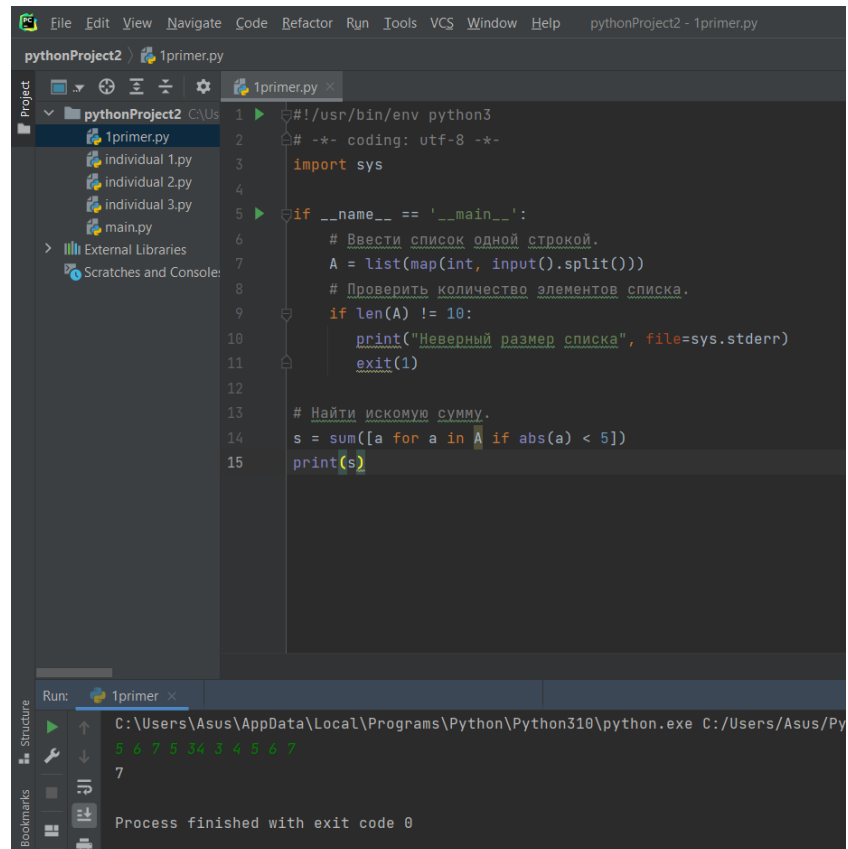
группы ИВТ-б-о-21-1

Толубаев Рамиль Ахметович

Ставрополь 2022

Выполнение работы:

Пример 1. Ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести её на экран.



The screenshot shows an IDE window titled 'pythonProject2 - 1primer.py'. The code in '1primer.py' is as follows:

```
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)
```

The Run window at the bottom shows the command: `C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Asus/Py`. The output is: `5 4 7 8 14 3 4 3 4 7` followed by `7`. The status bar indicates 'Process finished with exit code 0'.

Рисунок 1. Результат выполнения программы

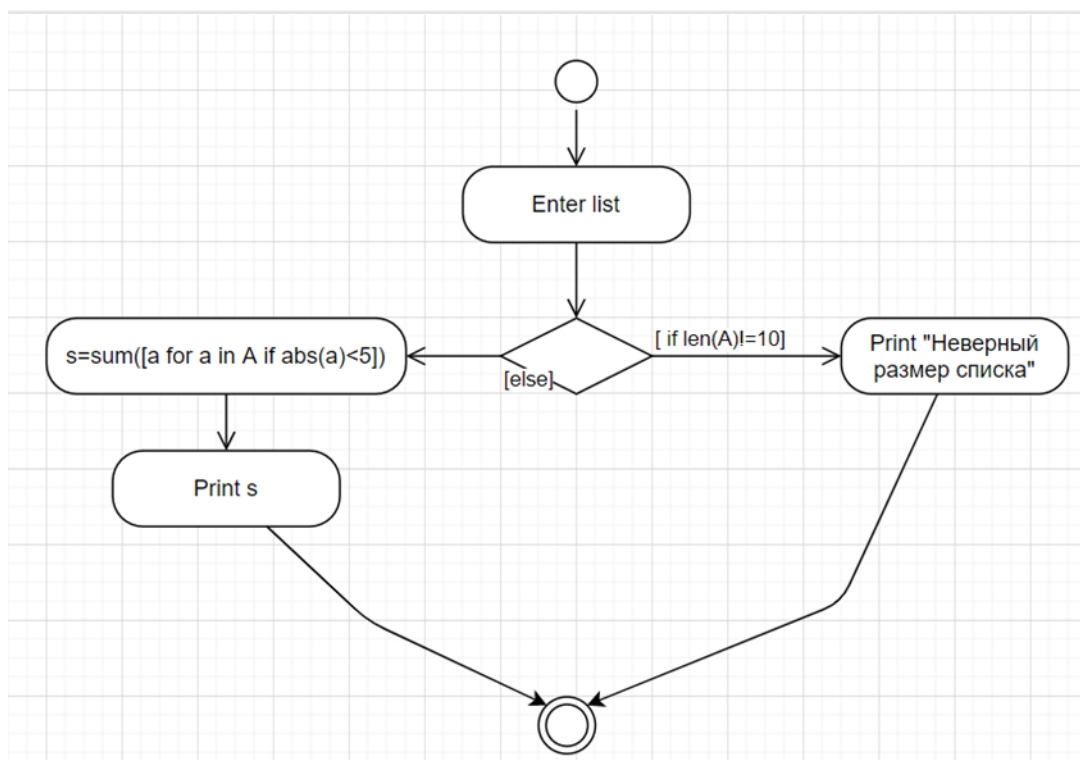
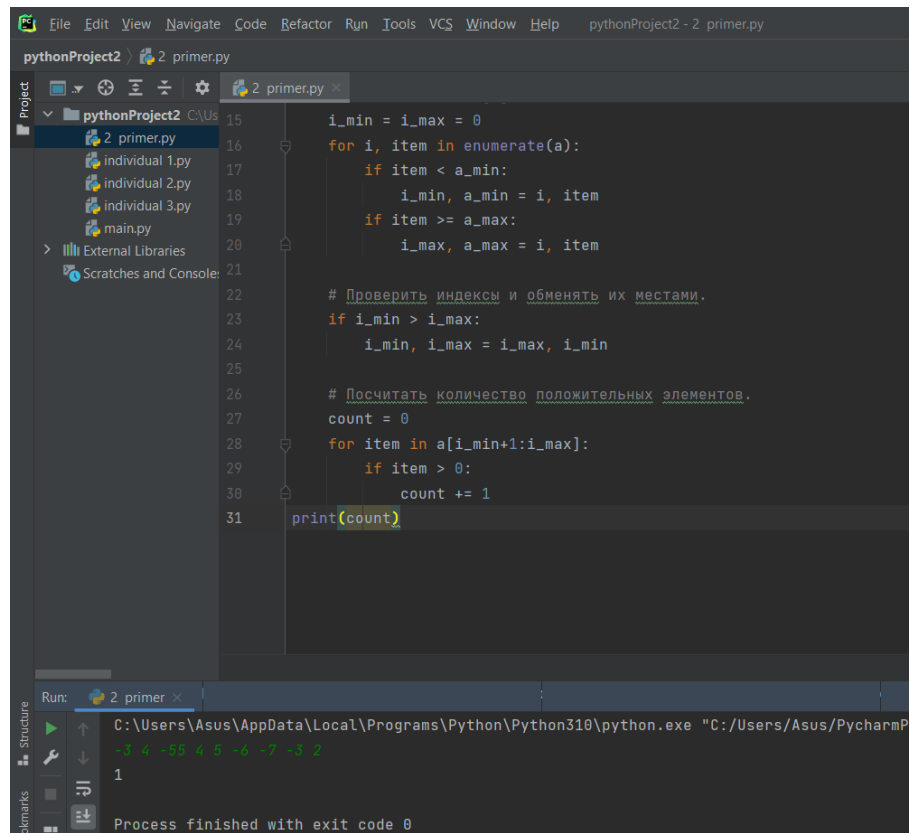


Рисунок 2. UML-диаграмма

Пример 2. Написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.



```
15     i_min = i_max = 0
16     for i, item in enumerate(a):
17         if item < a_min:
18             i_min, a_min = i, item
19         if item >= a_max:
20             i_max, a_max = i, item
21
22     # Проверить индексы и обменять их местами.
23     if i_min > i_max:
24         i_min, i_max = i_max, i_min
25
26     # Посчитать количество положительных элементов.
27     count = 0
28     for item in a[i_min+1:i_max]:
29         if item > 0:
30             count += 1
31     print(count)
```

Run: 2 primer ×

C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe "C:/Users/Asus/PycharmP
-3 4 -65 4 9 -6 -7 -3 2
1
Process finished with exit code 0

Рисунок 3. Результат выполнения программы

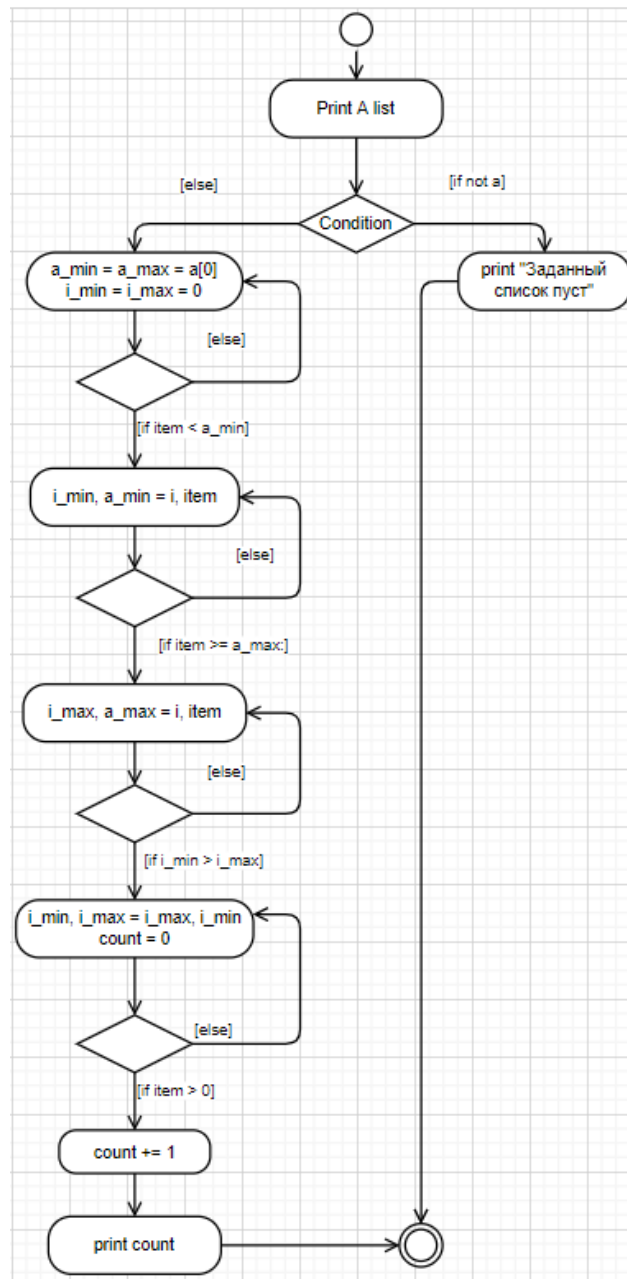


Рисунок 4. UML-диаграмма

Индивидуальное 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  #!/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      F = []
9      R = []
10     # Если список пуст, завершить программу.
11     if not A:
12         print("Заданный список пуст", file=sys.stderr)
13         exit(1)
14     for x in A:
15         if 8 < x and x < 18:
16             F.append(x)
17         t=0
18         m = 10
19     for i in F:
20         if i % 10 == 0:

```

Run: individual 3 × ind 1 ×

```

C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe "C:/Users/Asu
12 33 4 5 6 7 3 4 5 6 7
[12]
0
10
Process finished with exit code 0

```

Рисунок 5. Результат выполнения программы

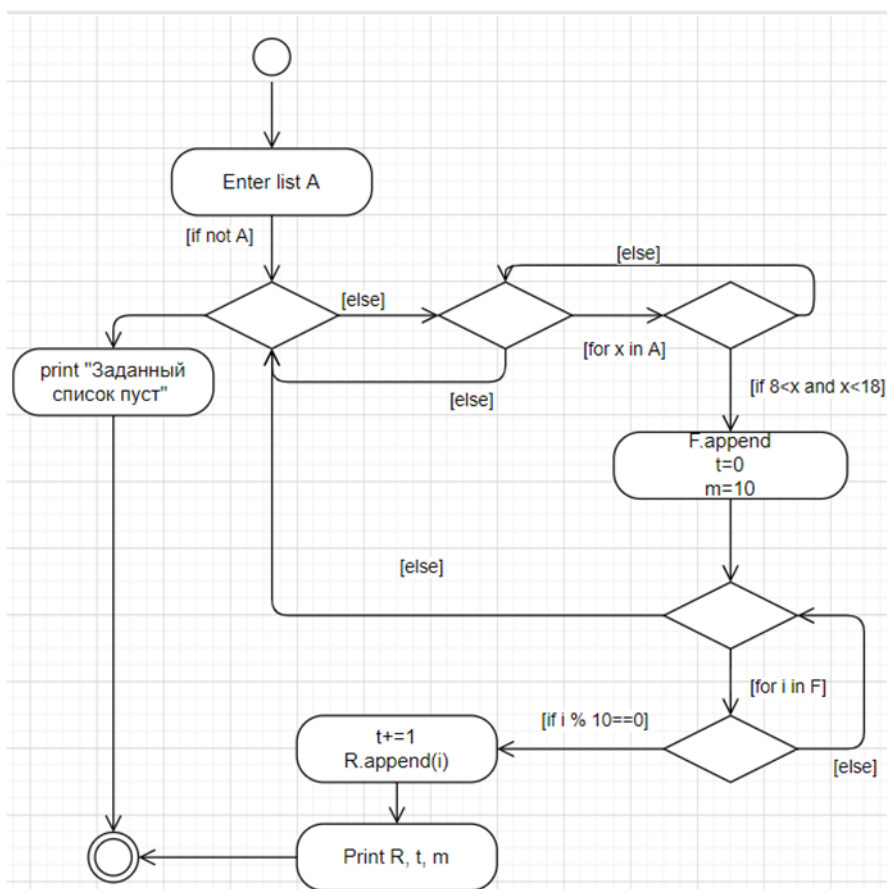
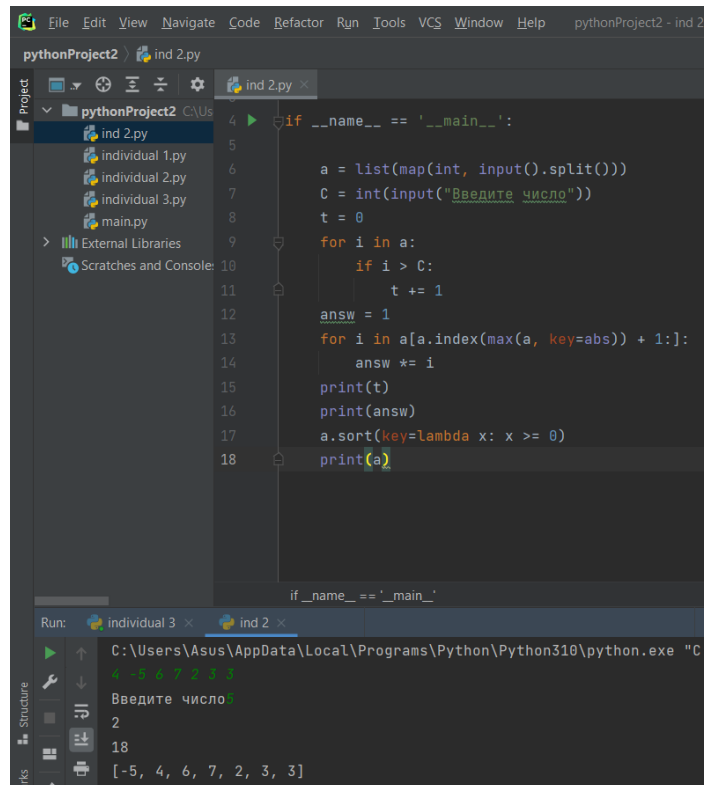


Рисунок 6. UML-диаграмма

Индивидуальное задание 2. В списке, состоящем из вещественных элементов, вычислить:

- 1) количество элементов списка, больших C;
- 2) произведение элементов списка, расположенных после максимального по модулю элемента.

Преобразовать список таким образом, чтобы сначала располагались все отрицательные элементы, а потом - все положительные (элементы, равные 0, считать положительными).



The screenshot shows an IDE with a project named 'pythonProject2'. The file explorer on the left shows files: 'ind 2.py', 'individual 1.py', 'individual 2.py', 'individual 3.py', and 'main.py'. The main editor displays the code in 'ind 2.py'.

```
4 if __name__ == '__main__':  
5  
6     a = list(map(int, input().split()))  
7     C = int(input("Введите число"))  
8     t = 0  
9  
10    for i in a:  
11        if i > C:  
12            t += 1  
13  
14    ans = 1  
15    for i in a[a.index(max(a, key=abs)) + 1:]:  
16        ans *= i  
17    print(t)  
18    print(ans)  
19    a.sort(key=lambda x: x >= 0)  
20    print(a)
```

The Run console at the bottom shows the execution of 'individual 3.py' and 'ind 2.py'. The output is as follows:

```
Run: individual 3 x ind 2 x  
C:\Users\Asus\AppData\Local\Programs\Python\Python310\python.exe C:  
-5 4 6 7 2 3 3  
Введите число  
2  
18  
[-5, 4, 6, 7, 2, 3, 3]
```

Рисунок 7. Результат выполнения программы

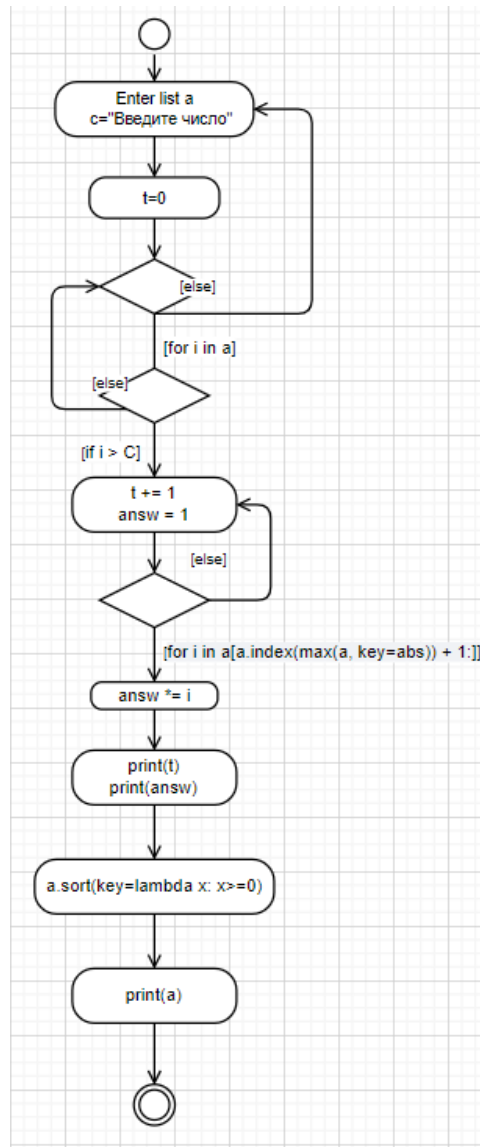


Рисунок 8. UML-диаграмма

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.

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?

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:

- `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.

13. 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.

```
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()`.