

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»**

Выполнил:  
Хачатрян Владимир Владимирович  
1 курс, группа ИТС-б-о-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.

## Работа со списками в языке Python.

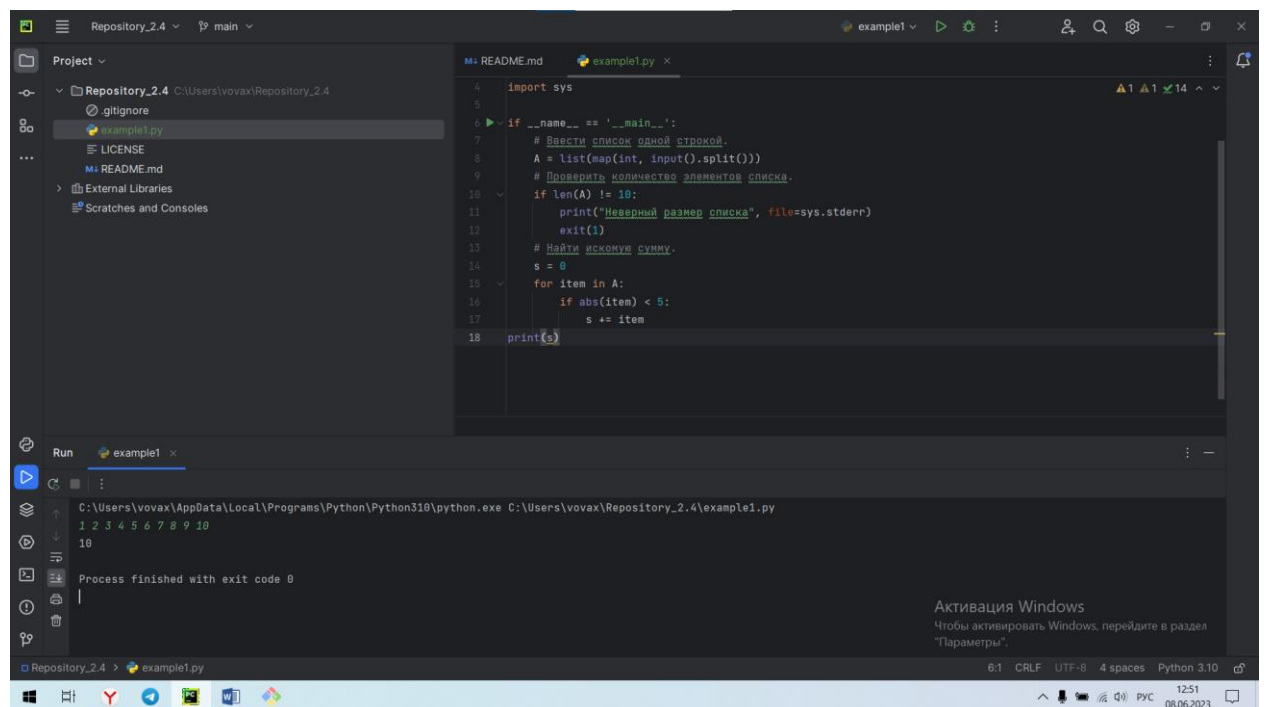
**Цель:** приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x.

### Ход работы:

Создал общедоступный репозиторий на GitHub, добавил файл .gitignore с необходимыми правилами. Клонировал репозиторий на свой компьютер.

```
C:\Users\vovax>git clone https://github.com/GitVolodya/Repository_2.4.git
Cloning into 'Repository_2.4'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

**Пример 1:** ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.



The screenshot shows a code editor with a file named `example1.py`. The code is as follows:

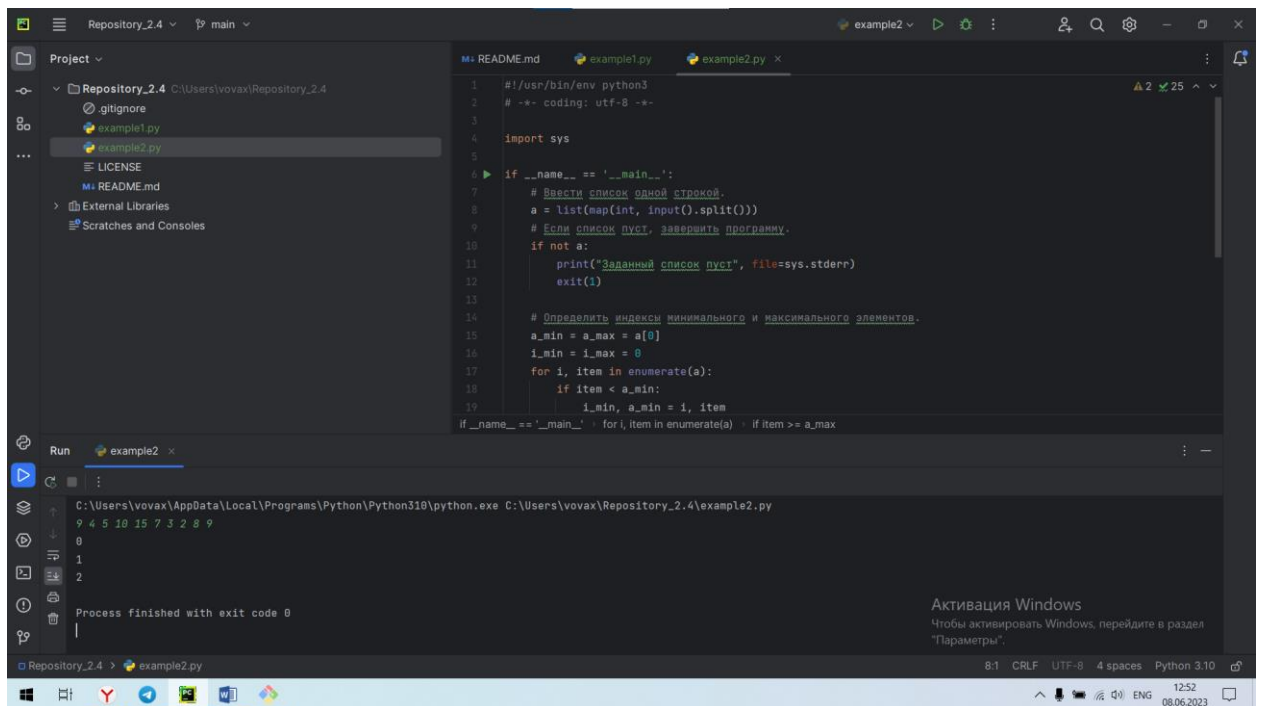
```
1 import sys
2
3 if __name__ == '__main__':
4     # Ввести список одной строкой.
5     A = list(map(int, input().split()))
6     # Проверить количество элементов списка.
7     if len(A) != 10:
8         print("Неверный размер списка", file=sys.stderr)
9         exit(1)
10    # Найти искомую сумму.
11    s = 0
12    for item in A:
13        if abs(item) < 5:
14            s += item
15    print(s)
```

The output of the program is shown in the console:

```
C:\Users\vovax\AppData\Local\Programs\Python\Python310\python.exe C:\Users\vovax\Repository_2.4\example1.py
1 2 3 4 5 6 7 8 9 10
10
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is 6:1 CRLF, UTF-8, 4 spaces, and the Python version is 3.10. The date and time are 12:51 on 08.06.2023.

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



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 if __name__ == '__main__':
7     # Ввести список одной строкой.
8     a = list(map(int, input().split()))
9     # Если список пуст, завершить программу.
10    if not a:
11        print("Заданный список пуст", file=sys.stderr)
12        exit(1)
13
14    # Определить индексы минимального и максимального элементов.
15    a_min = a_max = a[0]
16    i_min = i_max = 0
17    for i, item in enumerate(a):
18        if item < a_min:
19            i_min, a_min = i, item
20        if item >= a_max:
21            i_max, a_max = i, item
22
23    # Подсчитать количество положительных элементов между i_min и i_max.
24    count = 0
25    for i in range(i_min + 1, i_max):
26        if a[i] > 0:
27            count += 1
28
29    print(count)
```

Run example2

C:\Users\vovax\AppData\Local\Programs\Python\Python310\python.exe C:\Users\vovax\Repository\_2.4\example2.py

9 4 5 10 15 7 3 2 8 9

0

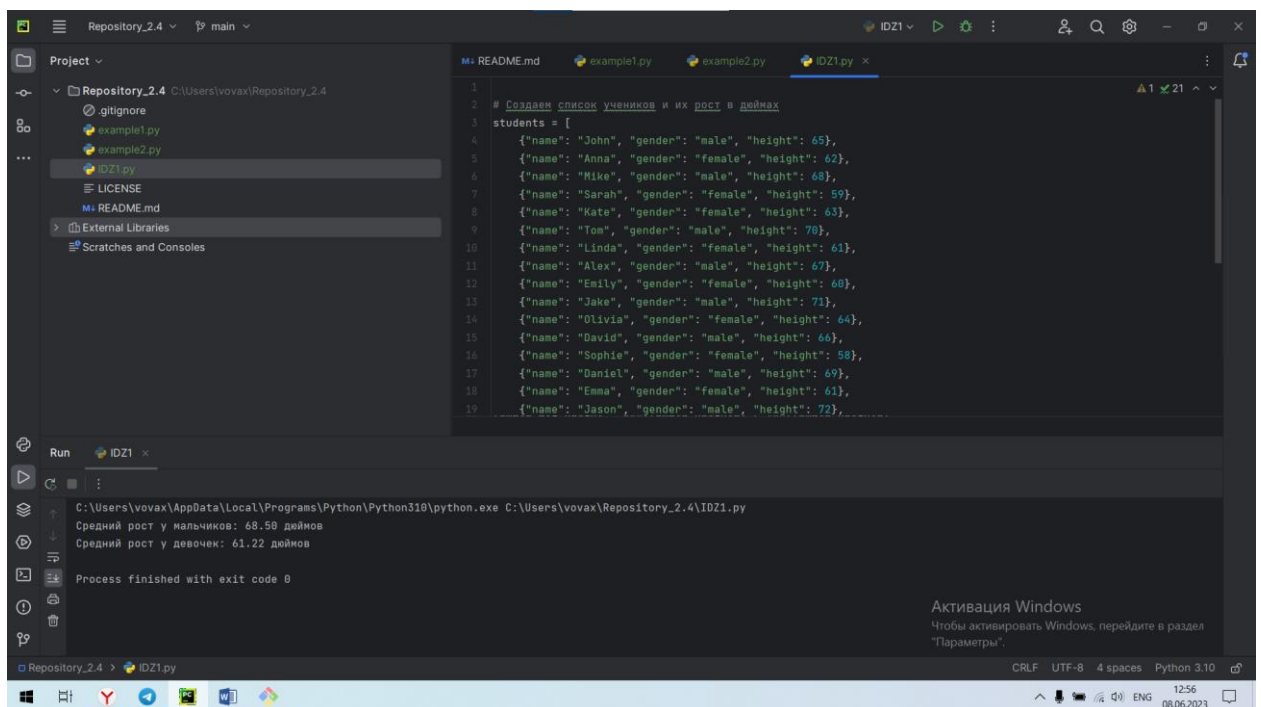
1

2

Process finished with exit code 0

## Индивидуальное задание 1

**Условие:** Определить средний рост девочек и мальчиков одного класса. В классе учится  $n$  учеников ( $n > 15$ ).



```
1 # Создаем список учеников и их рост в дюймах
2 students = [
3     {"name": "John", "gender": "male", "height": 65},
4     {"name": "Anna", "gender": "female", "height": 62},
5     {"name": "Mike", "gender": "male", "height": 68},
6     {"name": "Sarah", "gender": "female", "height": 59},
7     {"name": "Kate", "gender": "female", "height": 63},
8     {"name": "Tom", "gender": "male", "height": 70},
9     {"name": "Linda", "gender": "female", "height": 61},
10    {"name": "Alex", "gender": "male", "height": 67},
11    {"name": "Emily", "gender": "female", "height": 60},
12    {"name": "Jake", "gender": "male", "height": 71},
13    {"name": "Olivia", "gender": "female", "height": 64},
14    {"name": "David", "gender": "male", "height": 66},
15    {"name": "Sophie", "gender": "female", "height": 58},
16    {"name": "Daniel", "gender": "male", "height": 69},
17    {"name": "Emma", "gender": "female", "height": 61},
18    {"name": "Jason", "gender": "male", "height": 72},
19 ]
```

Run IDZ1

C:\Users\vovax\AppData\Local\Programs\Python\Python310\python.exe C:\Users\vovax\Repository\_2.4\IDZ1.py

Средний рост у мальчиков: 68.50 дюймов

Средний рост у девочек: 61.22 дюймов

Process finished with exit code 0

## Индивидуальное задание 2

**Условие задания:** в списке, состоящем из целых элементов, вычислить:

1. минимальный по модулю элемент списка;
2. сумму модулей элементов списка, расположенных после первого элемента, равного нулю.

Преобразовать список таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине - элементы, стоявшие в нечетных позициях.

```
12 print(f"Минимальный по модулю элемент списка: {min_elem}")
13
14 # Вычисляем сумму модулей элементов после первого нуля
15 if 0 in lst:
16     idx = lst.index(0)
17     sum_elems = sum(abs(x) for x in lst[idx + 1:])
18     print(f"Сумма модулей элементов списка, расположенных после первого элемента, равного нулю: {sum_elems}")
19 else:
20     print("В списке нет элементов, равных нулю.")
21
22 # Преобразуем список
23 new_lst = lst[::2] + lst[1::2]
24 print(f"Преобразованный список: {new_lst}")

if __name__ == "__main__":
```

Run IDZ2 x

C:\Users\vovax\AppData\Local\Programs\Python\Python310\python.exe C:\Users\vovax\Repository\_2.4\IDZ2.py

Введите список целых чисел через пробел: 1 2 5 10 17

Минимальный по модулю элемент списка: 1

В списке нет элементов, равных нулю.

Преобразованный список: [1, 5, 17, 2, 10]

Process finished with exit code 0

```
12 print(f"Минимальный по модулю элемент списка: {min_elem}")
13
14 # Вычисляем сумму модулей элементов после первого нуля
15 if 0 in lst:
16     idx = lst.index(0)
17     sum_elems = sum(abs(x) for x in lst[idx + 1:])
18     print(f"Сумма модулей элементов списка, расположенных после первого элемента, равного нулю: {sum_elems}")
19 else:
20     print("В списке нет элементов, равных нулю.")
21
22 # Преобразуем список
23 new_lst = lst[::2] + lst[1::2]
24 print(f"Преобразованный список: {new_lst}")

if __name__ == "__main__":
```

Run IDZ2 x

C:\Users\vovax\AppData\Local\Programs\Python\Python310\python.exe C:\Users\vovax\Repository\_2.4\IDZ2.py

Введите список целых чисел через пробел: 10 5 6 0 1

Минимальный по модулю элемент списка: 0

Сумма модулей элементов списка, расположенных после первого элемента, равного нулю: 1

Преобразованный список: [10, 6, 1, 5, 0]

Process finished with exit code 0

Ссылка на репозиторий: [https://github.com/GitVolodya/Repository\\_2.4.git](https://github.com/GitVolodya/Repository_2.4.git)

## 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:** в ходе работы я приобрел навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.