

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:

Быковская Стефания Станиславовна  
1 курс, группа ИТС-б-0-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи», направ-  
ленность (профиль) «Инфокоммуника-  
ционные системы и сети», очная форма  
обучения

---

(подпись)

Руководитель практики:

Воронкин Р.А. доцент кафедры инфо-  
коммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

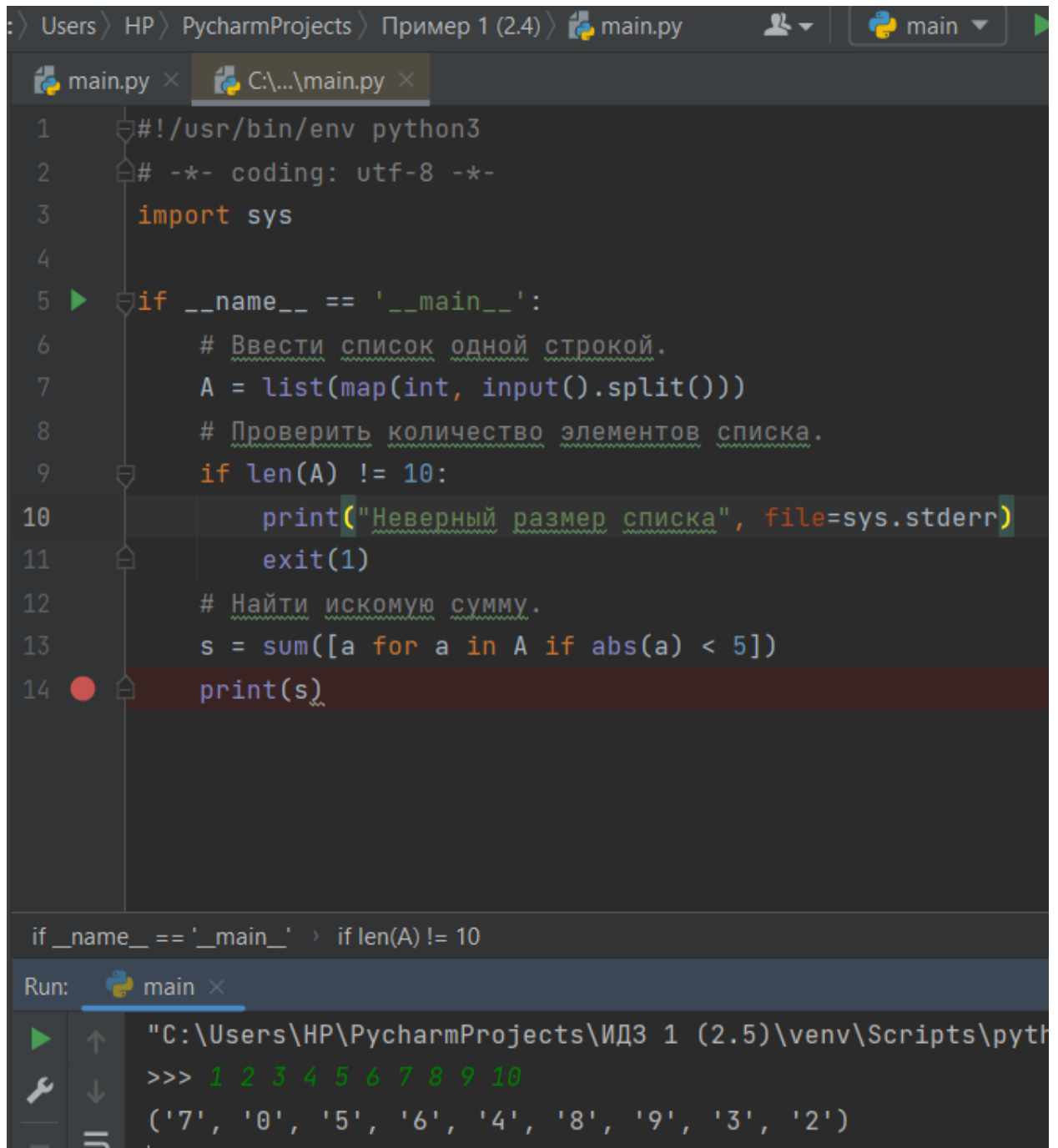
Ставрополь, 2023 г.

**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.

Порядок выполнения работы:

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



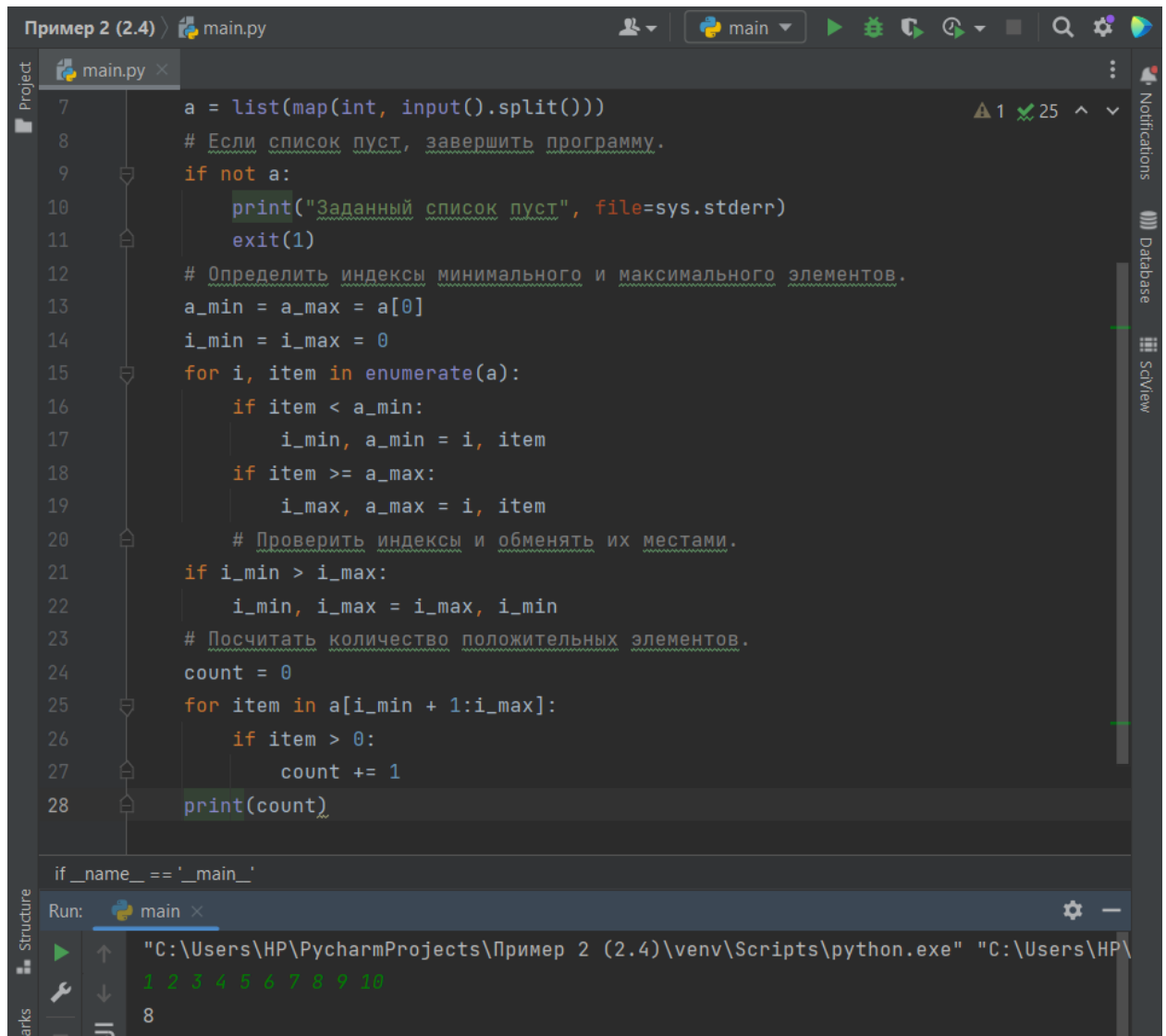
The screenshot shows the PyCharm IDE with a Python script named `main.py` open. The script is as follows:

```
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      # Найти искомую сумму.
13      s = sum([a for a in A if abs(a) < 5])
14      print(s)
```

The script is being executed, and the output is shown in the Run console at the bottom. The console shows the command prompt prompt `>>>` followed by the input `1 2 3 4 5 6 7 8 9 10` and the resulting list `('1', '2', '3', '4', '5', '6', '7', '8', '9', '10')`. The output of the script is `10`.

Рисунок 1 – Выполнение примера №1.

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



The screenshot shows the PyCharm IDE with a file named 'main.py' open. The code in the editor is as follows:

```
7 a = list(map(int, input().split()))
8 # Если список пуст, завершить программу.
9 if not a:
10     print("Заданный список пуст", file=sys.stderr)
11     exit(1)
12 # Определить индексы минимального и максимального элементов.
13 a_min = a_max = a[0]
14 i_min = i_max = 0
15 for i, item in enumerate(a):
16     if item < a_min:
17         i_min, a_min = i, item
18     if item >= a_max:
19         i_max, a_max = i, item
20     # Проверить индексы и обменять их местами.
21     if i_min > i_max:
22         i_min, i_max = i_max, i_min
23 # Посчитать количество положительных элементов.
24 count = 0
25 for item in a[i_min + 1:i_max]:
26     if item > 0:
27         count += 1
28 print(count)
```

Below the editor, the 'Run' window shows the execution command and output:

```
Run: main x
"C:\Users\HP\PycharmProjects\Пример 2 (2.4)\venv\Scripts\python.exe" "C:\Users\HP\
1 2 3 4 5 6 7 8 9 10
8
```

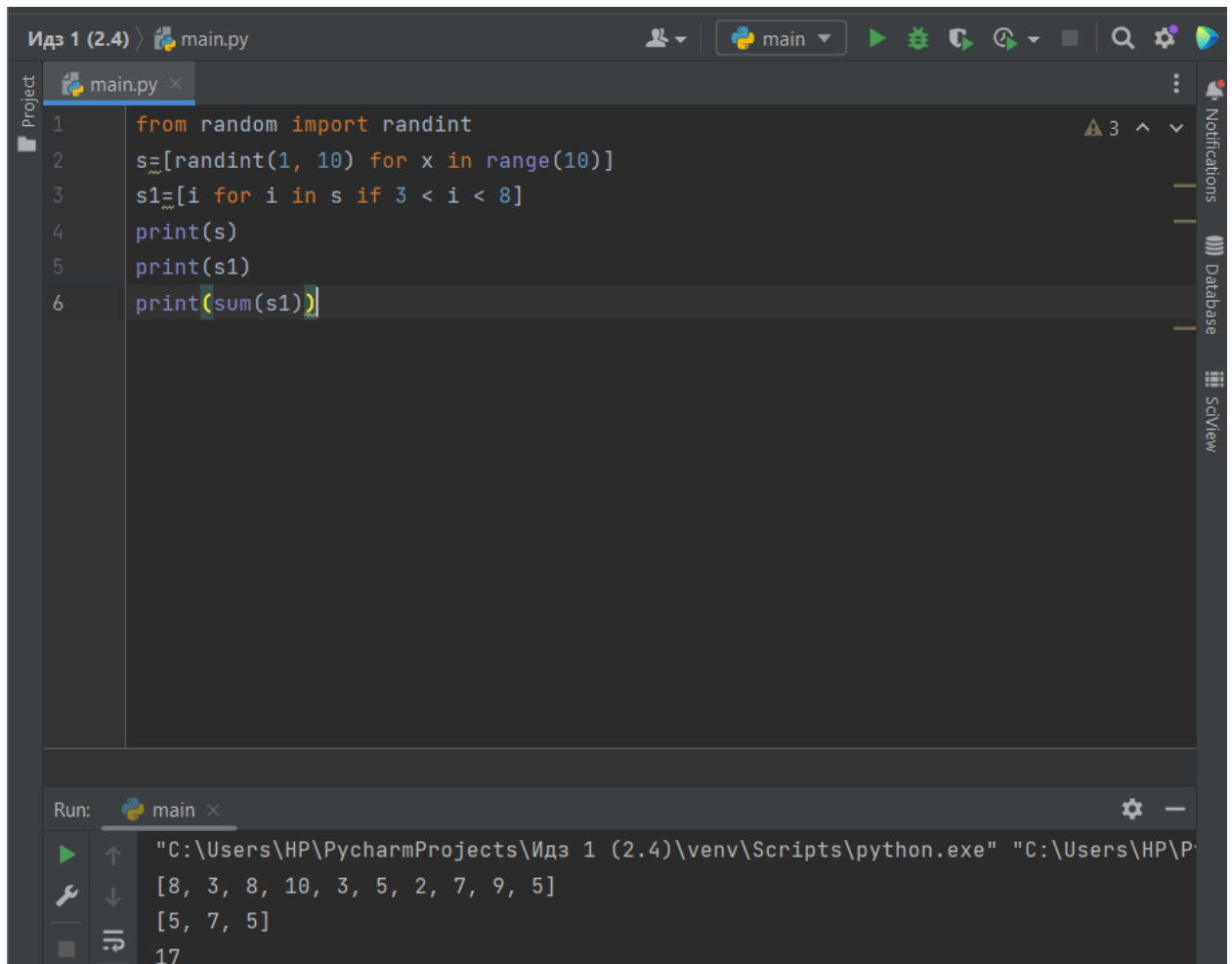
Рисунок 2 – Выполнение примера №2.

## Индивидуальные задания

### Вариант 5

### Задание 1

5. Ввести список А из 10 элементов, найти сумму элементов, больших 3 и меньших 8 и вывести ее на экран.



The screenshot shows the PyCharm IDE interface. The top toolbar includes icons for running, debugging, and other development tools. The main editor window displays a Python script named `main.py` with the following code:

```
1 from random import randint
2 s=[randint(1, 10) for x in range(10)]
3 s1=[i for i in s if 3 < i < 8]
4 print(s)
5 print(s1)
6 print(sum(s1))
```

The bottom panel shows the output of the script execution:

```
Run: main x
"C:\Users\HP\PycharmProjects\Идз 1 (2.4)\venv\Scripts\python.exe" "C:\Users\HP\P
[8, 3, 8, 10, 3, 5, 2, 7, 9, 5]
[5, 7, 5]
17
```

*Рисунок 3 – Выполнение ИДЗ 1.*

## **Задание 2**

Составить программу с использованием одномерных массивов для решения задачи на переупорядочивание элементов массива. Для сортировки допускается использовать метод `sort` с заданным параметром `key` (<https://docs.python.org/3/howto/sorting.html>) и объединение нескольких списков.

The screenshot shows the PyCharm IDE interface. The top pane displays a Python script named `main.py` with the following code:

```
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)))
```

The bottom pane shows the execution output for the script:

```
Run: main x
"C:\Users\HP\PycharmProjects\Идз 2 (2.4)\venv\Scripts\python.exe" "C:\Users\HP\
введите размер списка: 10
1.989 1.611 4.620 1.116 -4.108 -1.261 0.203 4.344 3.001 -1.636
введите нижнюю границу интервала: -4
введите верхнюю границу интервала: 4
максимум списка: 4.620
сумма списка: 8.513
преобразованный список:
4.620 -4.108 4.344 0.000 0.000 0.000 0.000 0.000 0.000 0.000
Process finished with exit code 0
```

Рисунок 4 - Выполнение ИДЗ №2.

## 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()`, 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);

`step` - шаг, с которым нужно выбирать элементы из списка.

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?

`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.

**Выводы:** приобрела навыки по работе со списками при написании программ с помощью языка программирования Python 3.x.