

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО РАБОТЕ №2.4**  
**дисциплины «Основы кроссплатформенного программирования»**

Выполнила:  
Дудова Мира Сергеевна  
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., канд. тех. наук, доцент,  
доцент кафедры инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** работа со списками в языке Python.

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

Порядок выполнения работы:

1. Создала новый репозиторий и клонировала его на свой компьютер

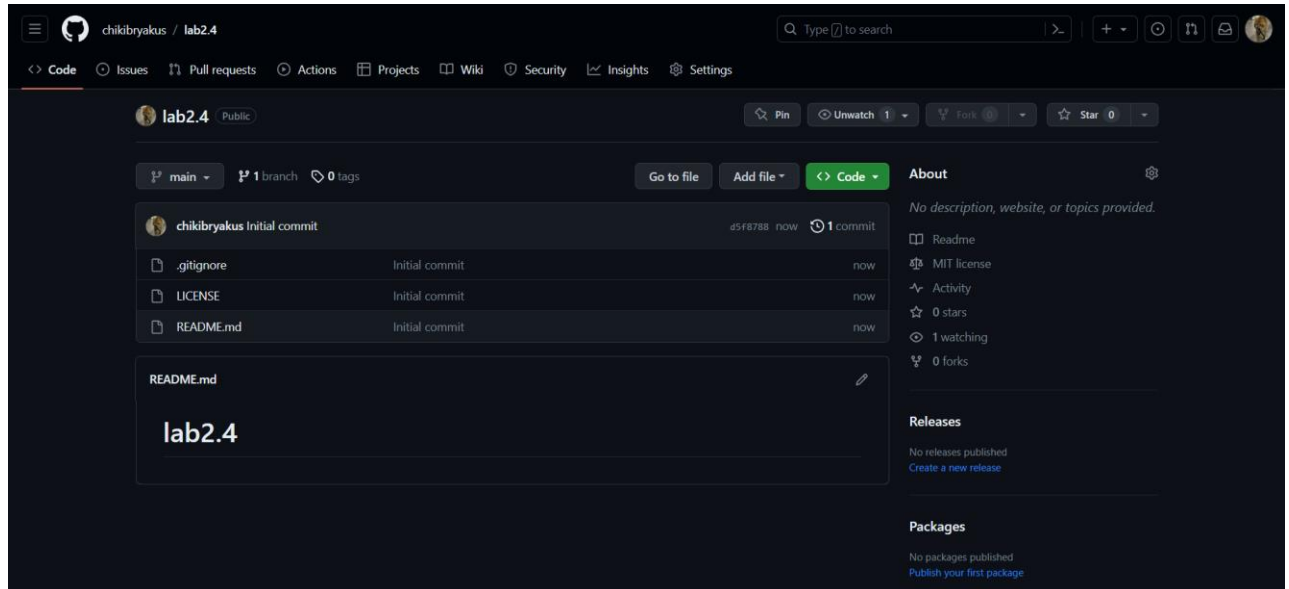


Рисунок 1 – Новый репозиторий

2. Проклонировала репозиторий на свой компьютер. В ходе данной лабораторной работы работала с моделью ветвления git-flow.

```
C:\Users\Hp>git clone https://github.com/chikibryakus/lab2.4.git
Cloning into 'lab2.4'...
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.

C:\Users\Hp>cd C:\Users\Hp\lab2.4
C:\Users\Hp\lab2.4>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

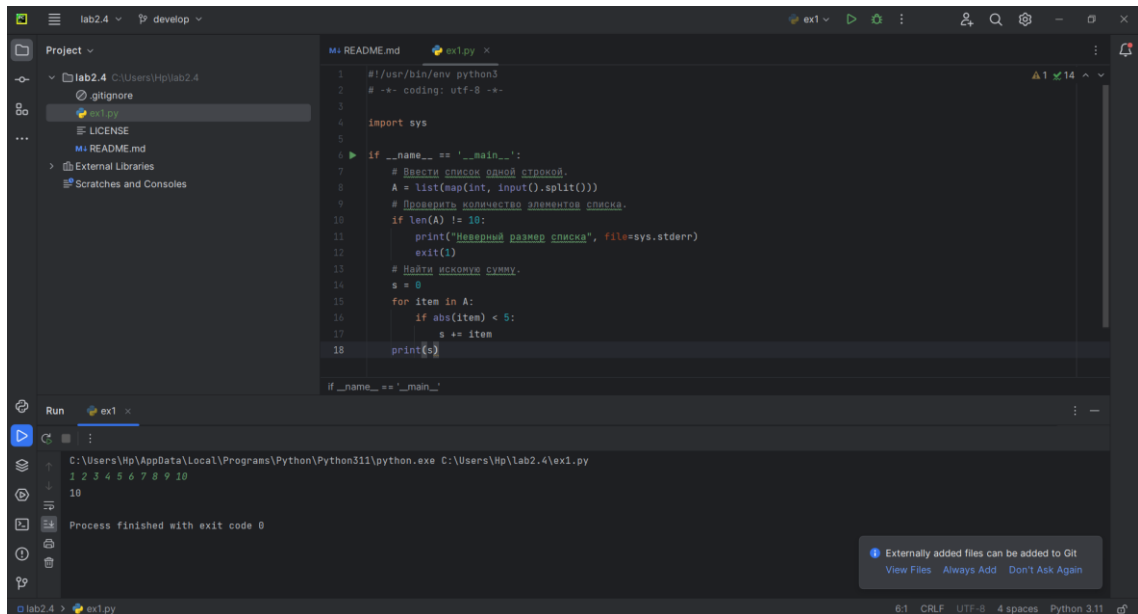
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Hp/lab2.4/.git/hooks]

C:\Users\Hp\lab2.4>
```

Рисунок 2 – Клонирование репозитория

3. Создала проект в папке репозитория и добавила новый файл ex1.py

Условие примера: вывести 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     # Ввести список одной строкой.
8     A = list(map(int, input().split()))
9     # Проверить количество элементов списка.
10    if len(A) != 10:
11        print("Неверный размер списка", file=sys.stderr)
12        exit(1)
13    # Найти сумму.
14    s = 0
15    for item in A:
16        if abs(item) < 5:
17            s += item
18    print(s)
19
20 if __name__ == '__main__':
```

Run ex1 x

C:\Users\Hp\AppData\Local\Programs\Python\Python311\python.exe C:\Users\Hp\lab2.4\ex1.py

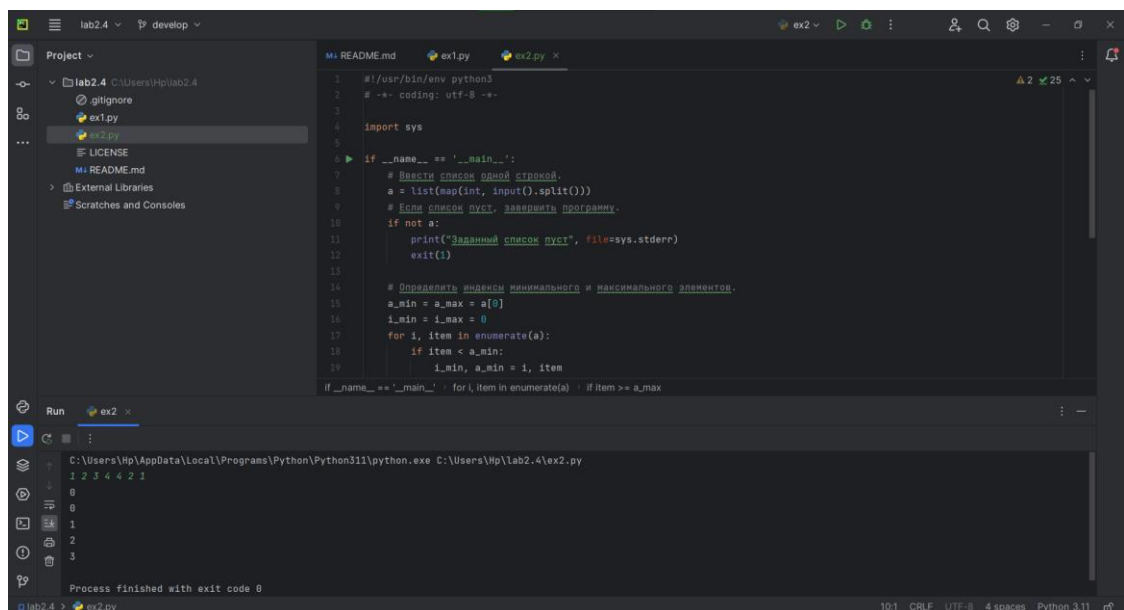
```
1 2 3 4 5 6 7 8 9 10
10
```

Process finished with exit code 0

Рисунок 3 – Программа и ее результат

4. Создала новый файл под названием ex2.py

**Условие примера:** 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    print(count)
29
30 if __name__ == '__main__':
```

Run ex2 x

C:\Users\Hp\AppData\Local\Programs\Python\Python311\python.exe C:\Users\Hp\lab2.4\ex2.py

```
1 2 3 4 4 2 1
0
0
1
2
3
```

Process finished with exit code 0

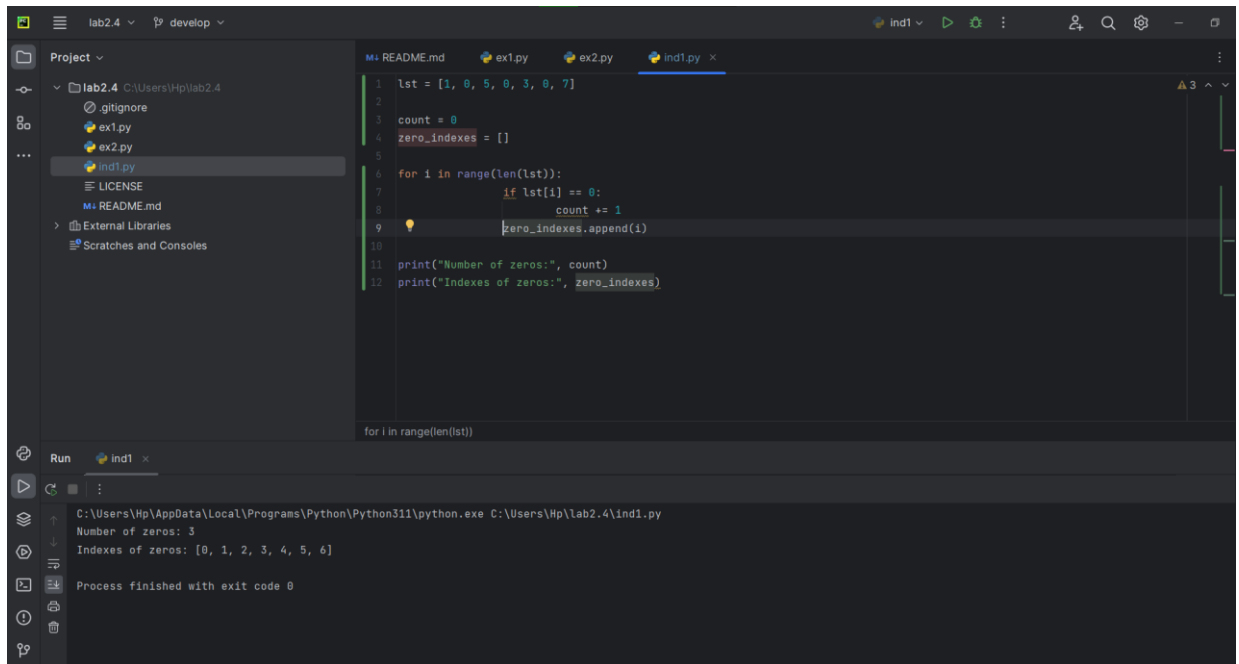
Рисунок 4 – Программа и ее результат

## 5. Индивидуальное задание №1.

Создала новый файл под названием *ind1.py*

Вариант 8.

**Условие задания:** в заданном списке подсчитать число нулевых элементов и вывести на экран их индексы.



The screenshot shows a code editor with a file named *ind1.py* open. The code in the editor is as follows:

```
1 lst = [1, 0, 5, 0, 3, 0, 7]
2
3 count = 0
4 zero_indexes = []
5
6 for i in range(len(lst)):
7     if lst[i] == 0:
8         count += 1
9         zero_indexes.append(i)
10
11 print("Number of zeros:", count)
12 print("Indexes of zeros:", zero_indexes)
```

Below the code editor, the output of the program is displayed:

```
Number of zeros: 3
Indexes of zeros: [0, 1, 2, 3, 4, 5, 6]
```

The output shows that there are 3 zeros in the list, and their indices are 0, 1, 2, 3, 4, 5, and 6. The process finished with exit code 0.

Рисунок 5 – Программа и ее результат

## 6. Индивидуальное задание №2.

Создала новый файл под названием *ind2.py*

Вариант 8.

**Условие задания:**

8. В списке, состоящем из вещественных элементов, вычислить:

1. максимальный по модулю элемент списка;
2. сумму элементов списка, расположенных между первым и вторым положительными элементами.

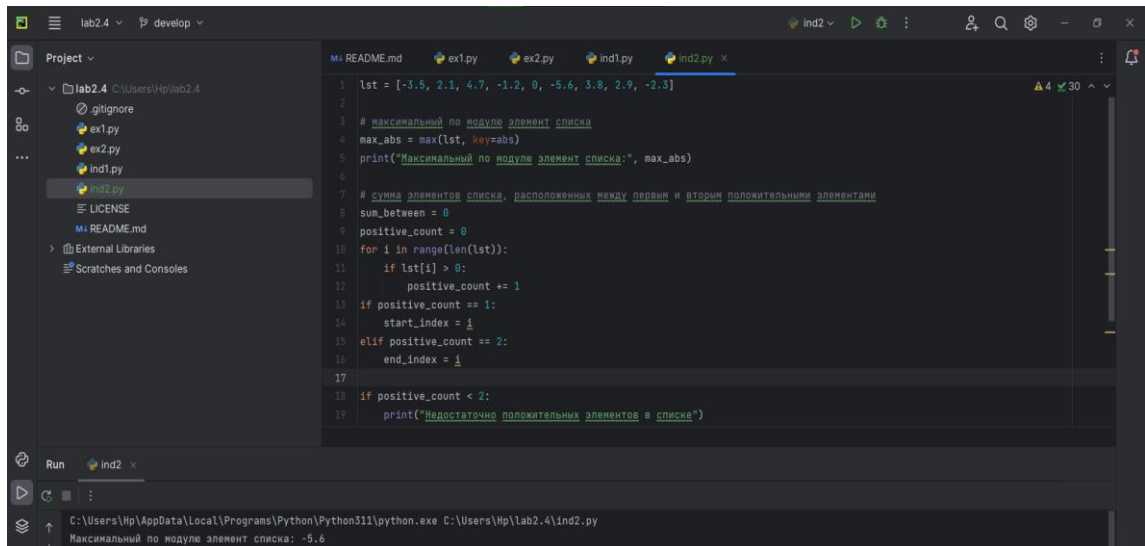


Рисунок 6 – Результат программы

7. Слила ветку develop с веткой main и отправила на удаленный сервер.

```
C:\Users\Hp\lab2.4>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\Hp\lab2.4>git merge develop
Updating d5f8788..73d587a
Fast-forward
 ex1.py | 18 +++++
 ex2.py | 32 +++++
 ind1.py | 12 +++++
 ind2.py | 22 +++++
 4 files changed, 84 insertions(+)
 create mode 100644 ex1.py
 create mode 100644 ex2.py
 create mode 100644 ind1.py
 create mode 100644 ind2.py

C:\Users\Hp\lab2.4>git push
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 2.34 KiB | 2.34 MiB/s, done.
Total 13 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/chikibryakus/lab2.4.git
 d5f8788..73d587a main -> main
```

Рисунок 7 – Слияние ветки develop с основной веткой main

Ссылка: <https://github.com/chikibryakus/lab2.4>

**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 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 список?

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:

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.

#### 13. Как создать копию списка?

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().

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. Синтаксис: List\_name.sort(key, reverse=False) Параметры: ключ: Функция, которая служит ключом для сравнения сортировки. реверс: Если true, то список сортируется в порядке убывания.

**Вывод:** в ходе выполнения лабораторной работы приобретены навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x.