

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 РАБОТЕ №2.4
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Кондратенко Даниил Витальевич
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:

Задание 1.

Изучил теоретический материал работы, создал общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавил файл .gitignore с необходимыми правилами.

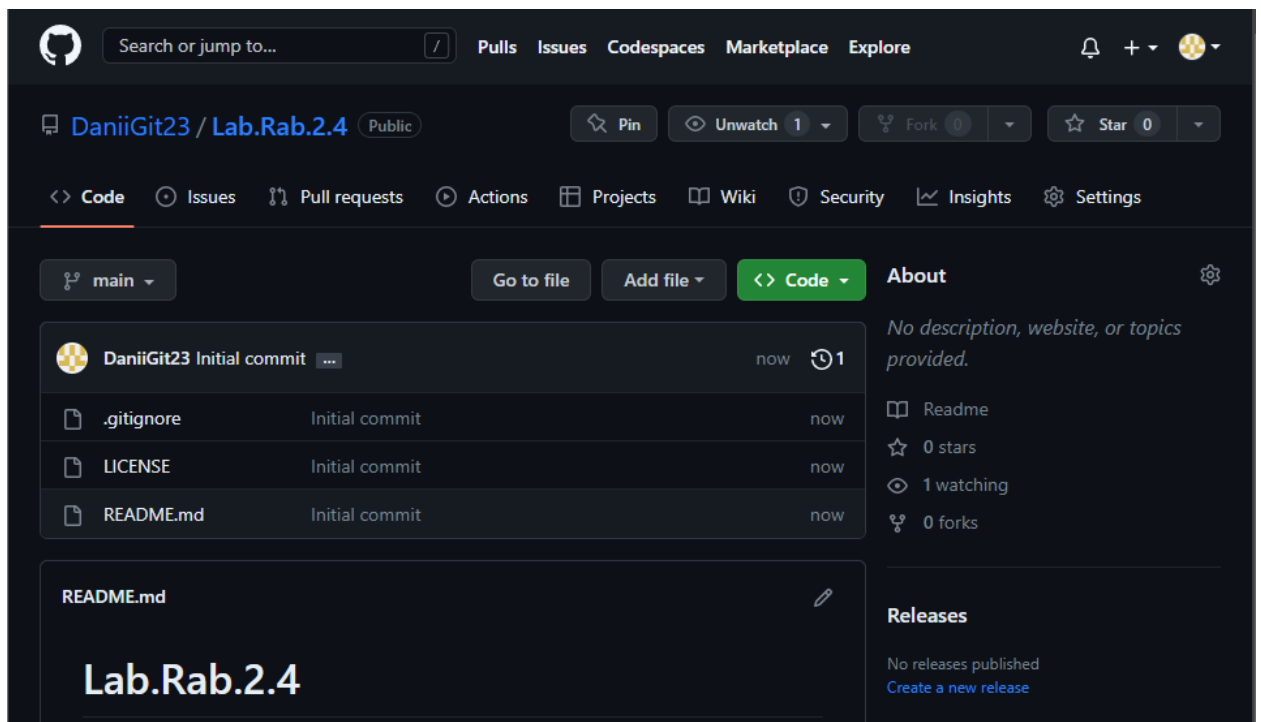
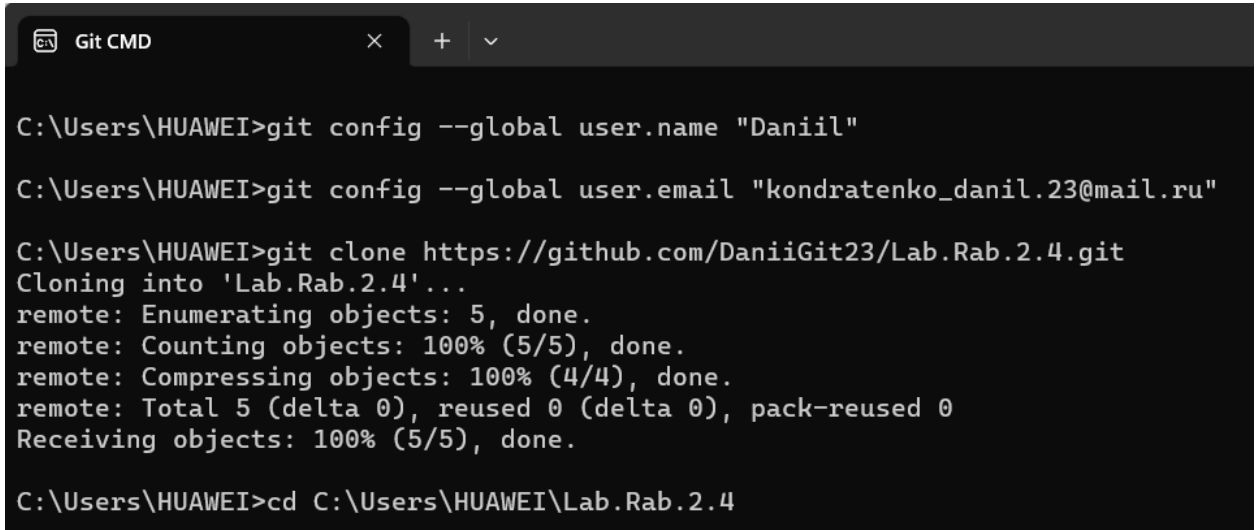


Рисунок 1 – Новый репозиторий

Задание 2.

Проклонировал репозиторий на свой компьютер.

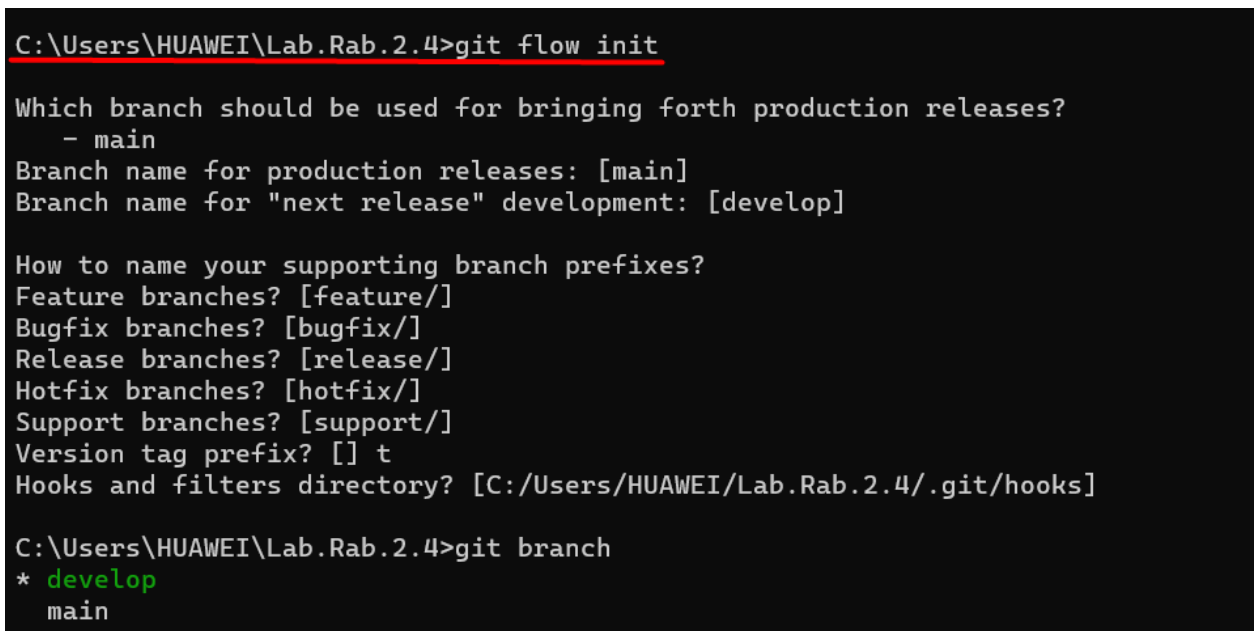


```
Git CMD
C:\Users\HUAWEI>git config --global user.name "Daniil"
C:\Users\HUAWEI>git config --global user.email "kondratenko_danil.23@mail.ru"
C:\Users\HUAWEI>git clone https://github.com/DaniiGit23/Lab.Rab.2.4.git
Cloning into 'Lab.Rab.2.4'...
remote: Enumerating objects: 5, done.
remote: Counting 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.
C:\Users\HUAWEI>cd C:\Users\HUAWEI\Lab.Rab.2.4
```

Рисунок 2 – Клонирование репозитория

Задание 3.

Организовал свой репозиторий в соответствии с моделью ветвления git-flow, появилась новая ветка develop.



```
C:\Users\HUAWEI\Lab.Rab.2.4>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? [] t
Hooks and filters directory? [C:/Users/HUAWEI/Lab.Rab.2.4/.git/hooks]

C:\Users\HUAWEI\Lab.Rab.2.4>git branch
* develop
main
```

Рисунок 3 – Модель ветвления git-flow

Реализовывал примеры и индивидуальные задания на основе ветки develop, без создания дополнительной ветки feature/(название ветки) по указанию преподавателя.

Задание 4.

Создал проект PyCharm в папке репозитория.

Работа с примером №1.

Добавил новый файл `primer1.py`.

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

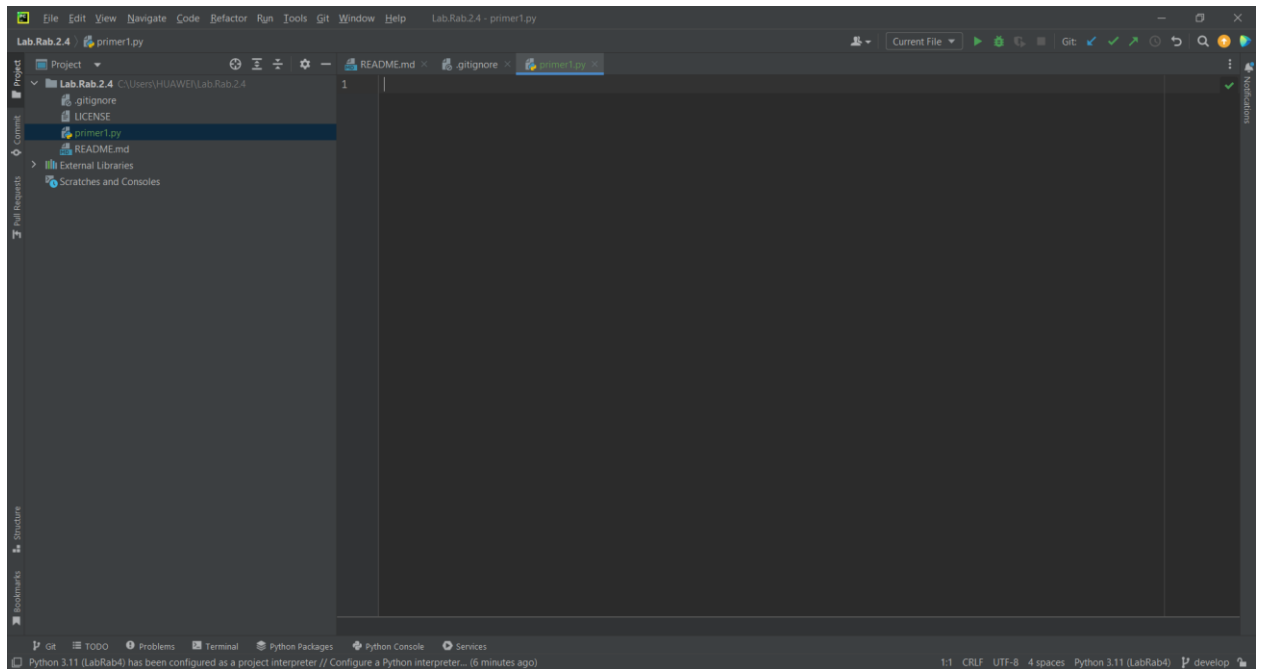


Рисунок 4 – Проект PyCharm и новый файл *primer1.py*

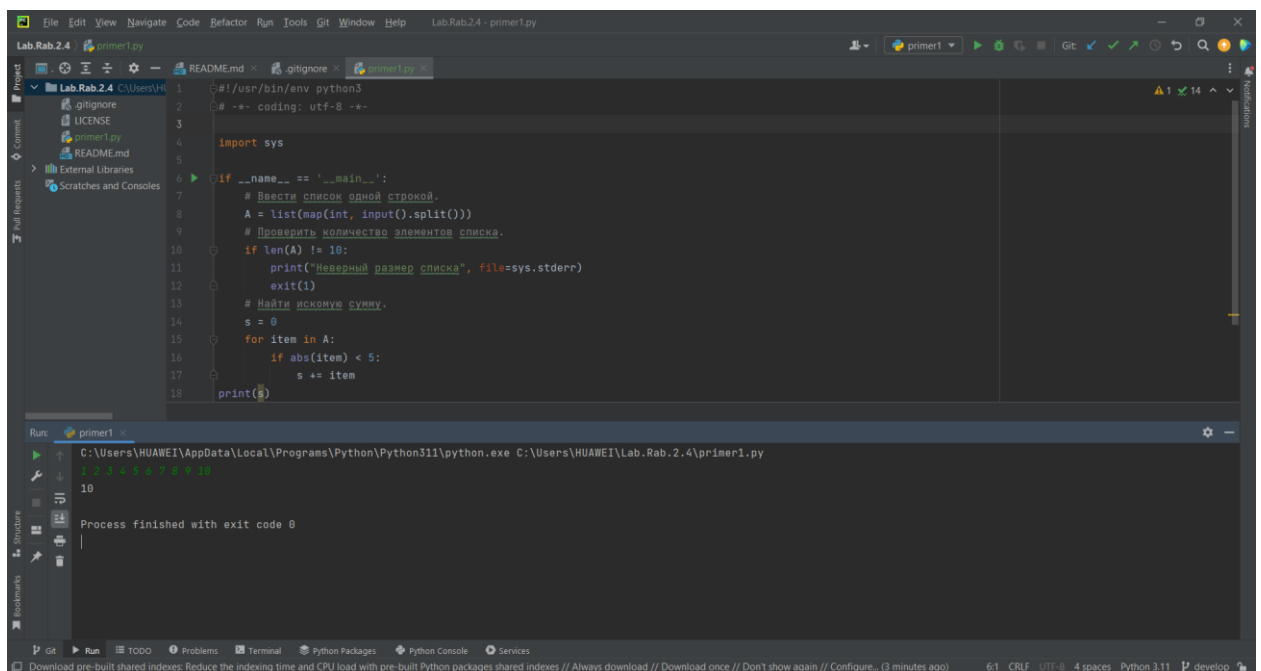


Рисунок 5 – Программа и ее результат

Задание 5.

Создал новый файл по названию *primer2.py*

Работа с примером №2.

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

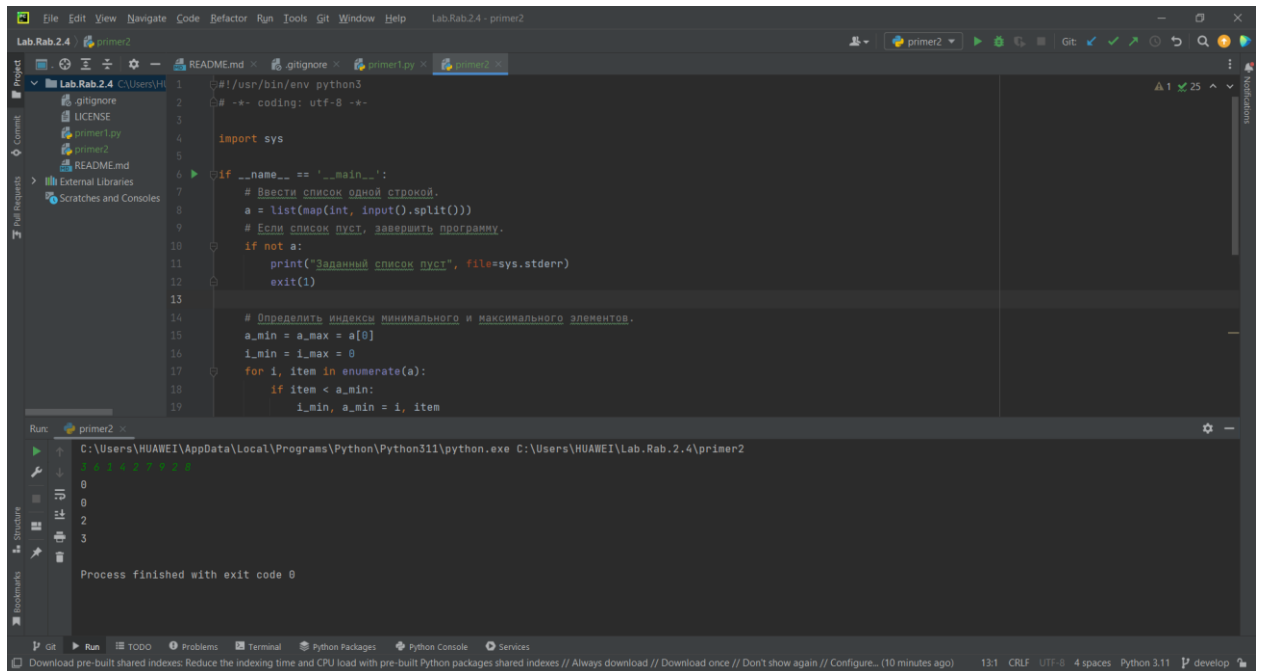


Рисунок 6 – Программа и ее результат

Задание 6.

Выполнение индивидуального задания №1.

Создал новый файл под названием *individual1.py*

Вариант 12 (по списку группы).

Условие задания: ввести список A из 10 элементов, найти сумму элементов, больших 2 и меньших 20 и кратных 8, их количество и вывести результаты на экран.

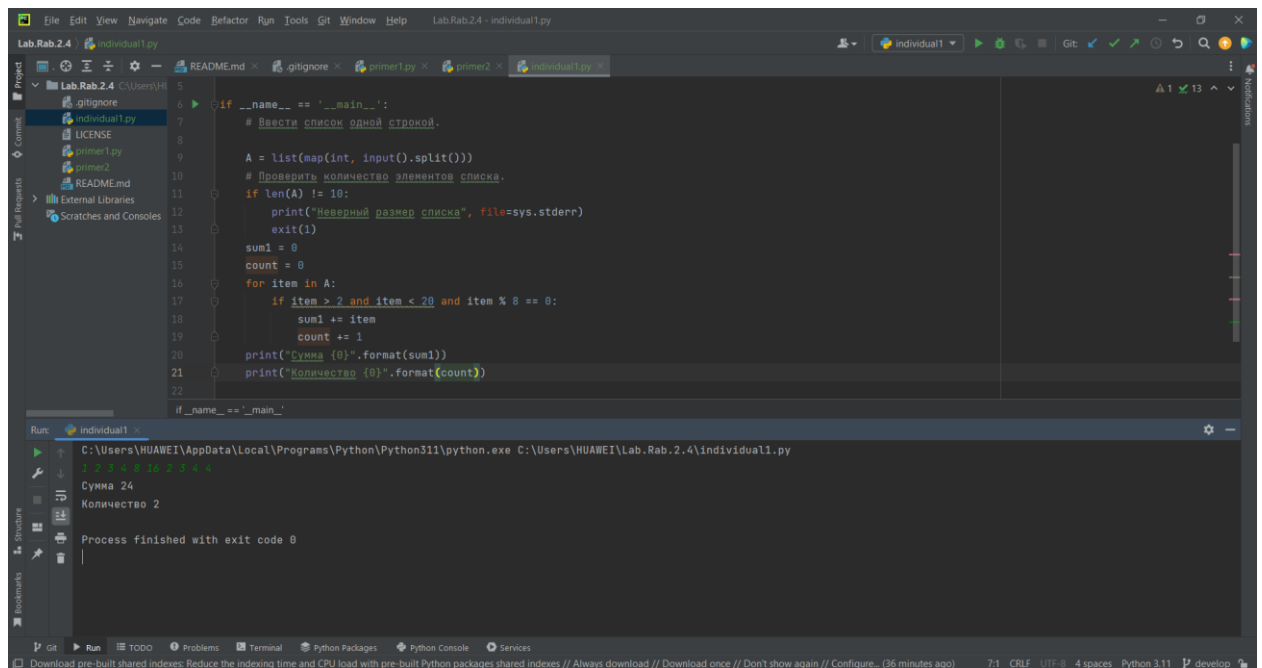


Рисунок 7 – Программа и ее результат

Задание 7.

Выполнение индивидуального задание №1.

Создал новый файл под названием *individual2.py*

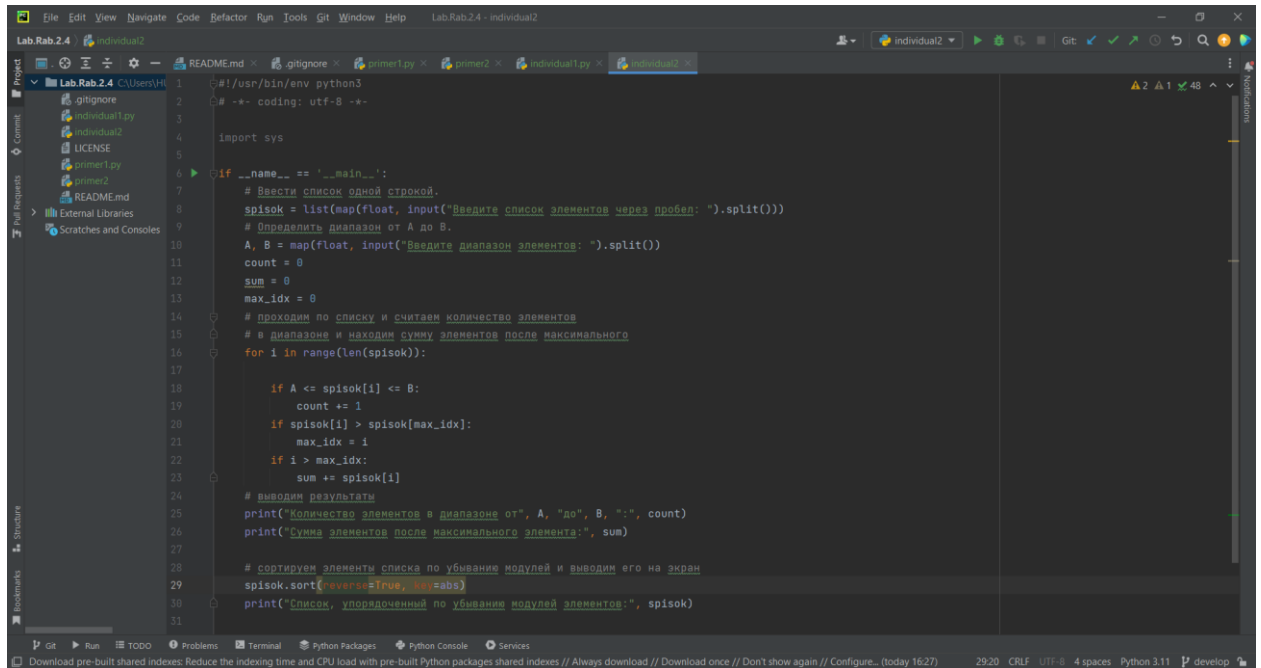
Вариант 12 (по списку группы).

Условие задания: в списке, состоящем из вещественных элементов, ВЫЧИСЛИТЬ:

1. количество элементов списка, лежащих в диапазоне от A до B;
2. сумму элементов списка, расположенных после максимального элемента.

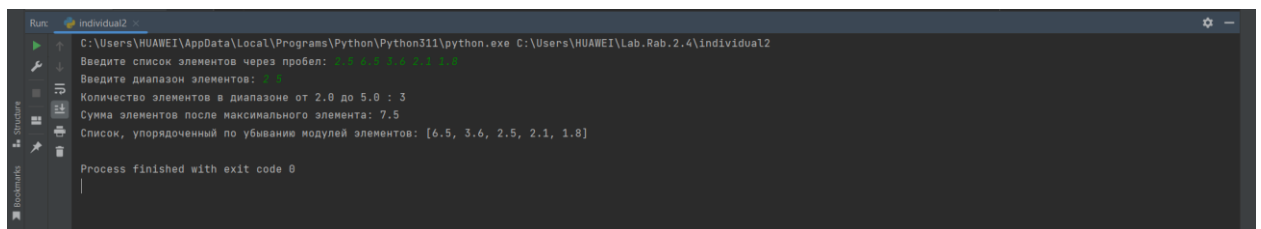
Упорядочить элементы списка по убыванию модулей элементов.

Код программы:



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import sys
5
6 if __name__ == '__main__':
7     # Ввести список одной строкой.
8     spisok = list(map(float, input("Введите список элементов через пробел: ").split()))
9     # Определить диапазон от A до B.
10    A, B = map(float, input("Введите диапазон элементов: ").split())
11    count = 0
12    sum = 0
13    max_idx = 0
14
15    # Проходим по списку и считаем количество элементов
16    # в диапазоне и находим сумму элементов после максимального
17    for i in range(len(spisok)):
18
19        if A <= spisok[i] <= B:
20            count += 1
21        if spisok[i] > spisok[max_idx]:
22            max_idx = i
23        if i > max_idx:
24            sum += spisok[i]
25
26    # выводим результаты
27    print("Количество элементов в диапазоне от", A, "до", B, ":", count)
28    print("Сумма элементов после максимального элемента:", sum)
29
30    # сортируем элементы списка по убыванию модулей и выводим его на экран
31    spisok.sort(reverse=True, key=abs)
32    print("Список, упорядоченный по убыванию модулей элементов:", spisok)
```

Рисунок 8 – Код

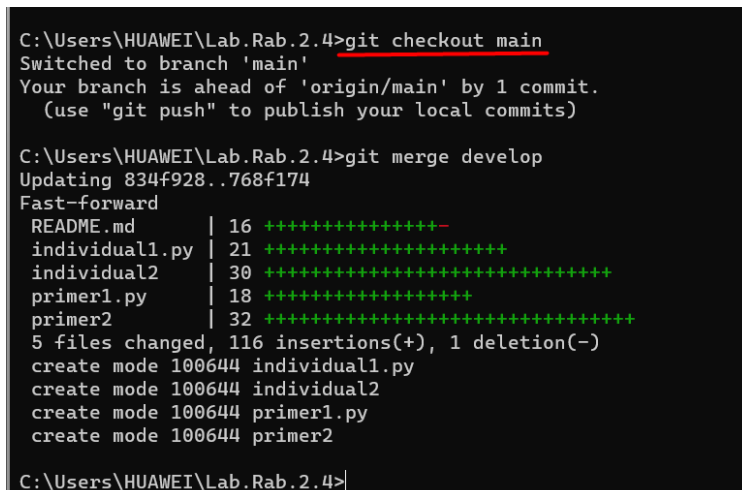


```
C:\Users\HUAWEI\AppData\Local\Programs\Python\Python311\python.exe C:\Users\HUAWEI\Lab.Rab.2.4\individual2
Введите список элементов через пробел: 1.5 3.6 2.5 2.1 1.8
Введите диапазон элементов: 2.0 5.0
Количество элементов в диапазоне от 2.0 до 5.0 : 3
Сумма элементов после максимального элемента: 7.5
Список, упорядоченный по убыванию модулей элементов: [6.5, 3.6, 2.5, 2.1, 1.8]
Process finished with exit code 0
```

Рисунок 9 – Результат программы

Задание 8.

Слил ветку develop с веткой main и отправил на удаленный сервер.



```
C:\Users\HUAWEI\Lab.Rab.2.4>git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

C:\Users\HUAWEI\Lab.Rab.2.4>git merge develop
Updating 834f928..768f174
Fast-forward
 README.md      | 16 ++++++++
 individual1.py | 21 ++++++++
 individual2    | 30 ++++++++
 primer1.py     | 18 ++++++++
 primer2       | 32 ++++++++
 5 files changed, 116 insertions(+), 1 deletion(-)
 create mode 100644 individual1.py
 create mode 100644 individual2
 create mode 100644 primer1.py
 create mode 100644 primer2

C:\Users\HUAWEI\Lab.Rab.2.4>
```

Рисунок 10 – Слияние ветки develop с основной веткой main

Ссылка на репозиторий: <https://github.com/DaniiGit23/Lab.Rab.2.4.git>

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? Чтение элементов 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.