

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:
Ермолович Иван Денисович
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. Создал новый репозиторий и клонировал его на свой компьютер

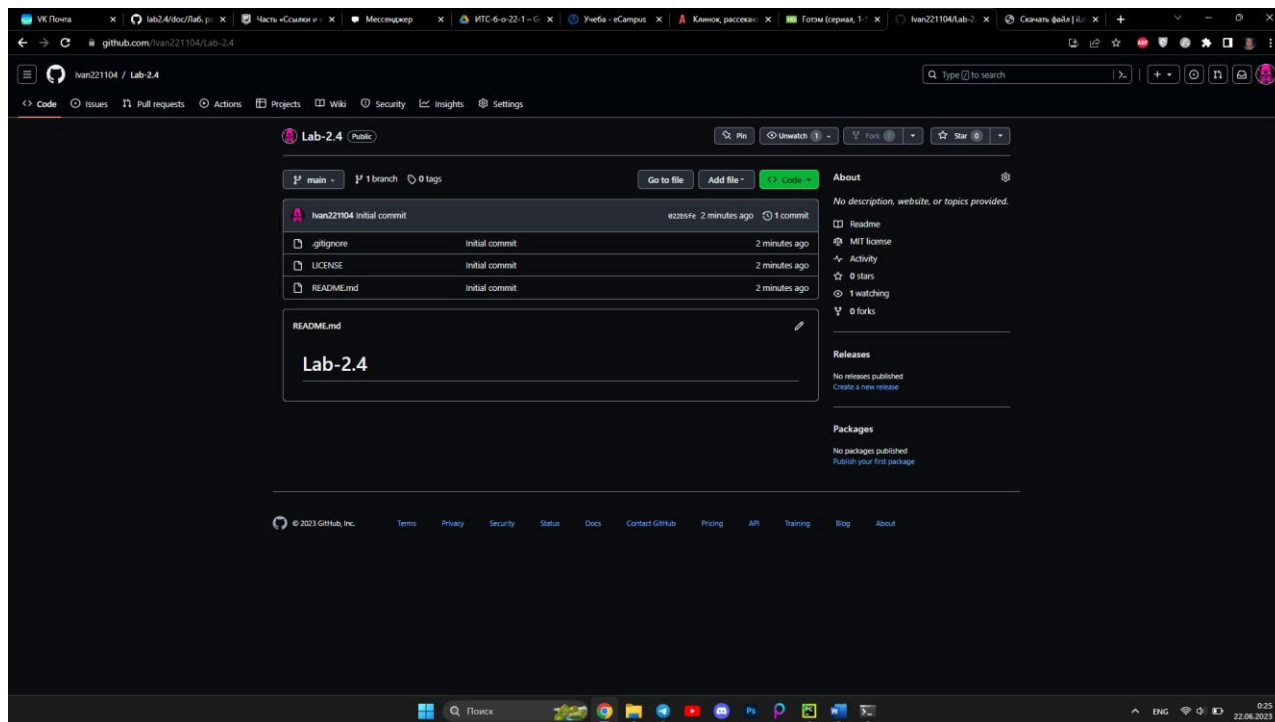
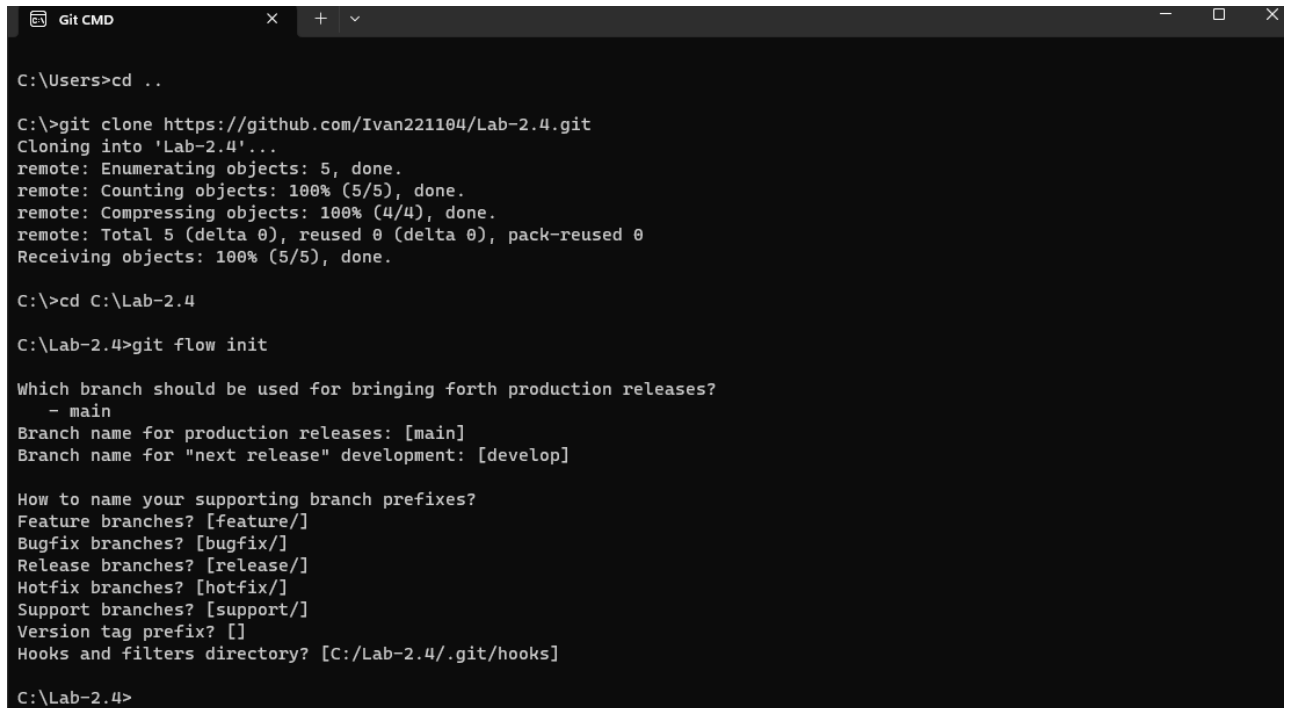


Рисунок 1 – Новый репозиторий

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



```
C:\Users>cd ..

C:\>git clone https://github.com/Ivan221104/Lab-2.4.git
Cloning into 'Lab-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:\>cd C:\Lab-2.4

C:\Lab-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? []
Hooks and filters directory? [C:/Lab-2.4/.git/hooks]

C:\Lab-2.4>
```

Рисунок 2 – Клонирование репозитория

3. Создал проект PyCharm в папке репозитория.

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

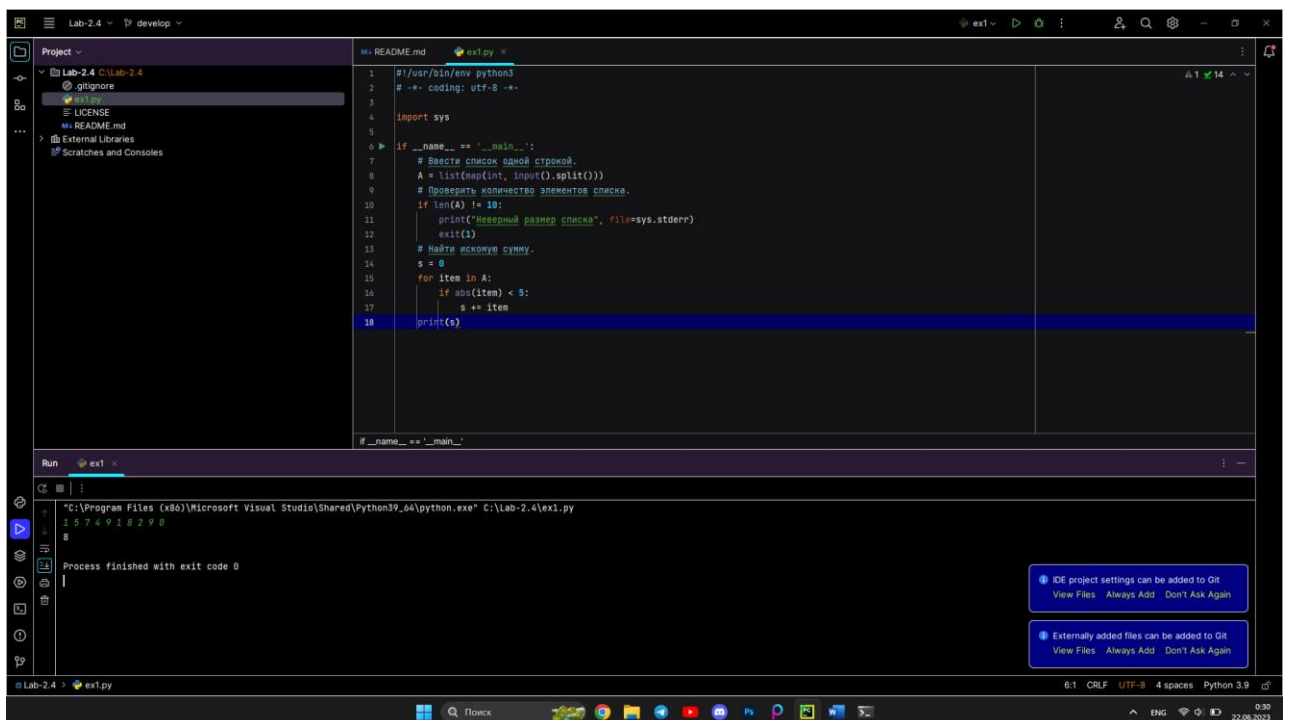
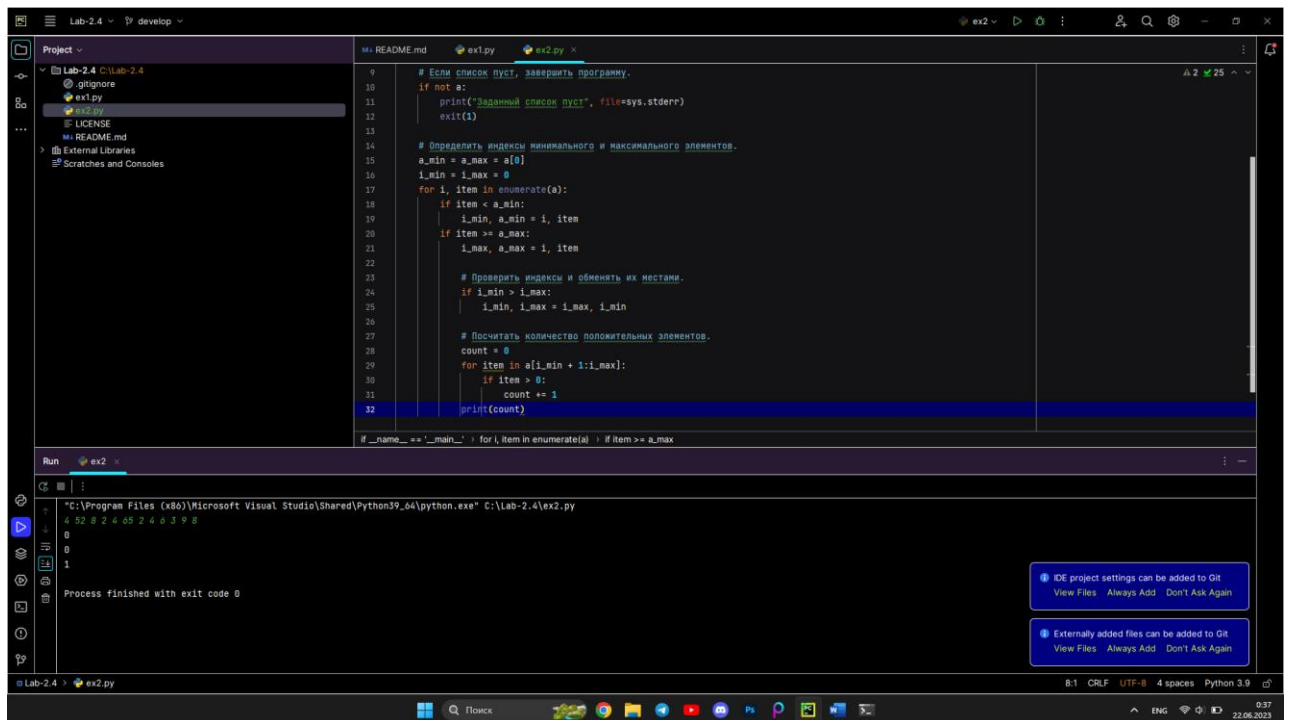


Рисунок 3 – Программа и ее результат

4. Создал новый файл под названием *ex2.py*

Пример №2.

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



The screenshot shows a code editor with a project named 'Lab-2.4'. The file 'ex2.py' is open, displaying the following Python code:

```
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 # Проверить индексы и поменять их местами.
24 if i_min > i_max:
25     i_min, i_max = i_max, i_min
26
27 # Посчитать количество положительных элементов.
28 count = 0
29 for item in a[i_min + 1:i_max]:
30     if item > 0:
31         count += 1
32 print(count)
```

The 'Run' panel at the bottom shows the execution output:

```
"C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python9_04\python.exe" C:\Lab-2.4\ex2.py
+ 52 8 2 + 05 2 4 0 3 9 8
0
1
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, 4 spaces, and Python 3.9.

Рисунок 4 – Программа и ее результат

5. Индивидуальное задание №1.

Создал новый файл под названием

Вариант 11

Ввести список A из 10 элементов, найти сумму отрицательных элементов кратных 7, их количество и вывести результаты на экран.

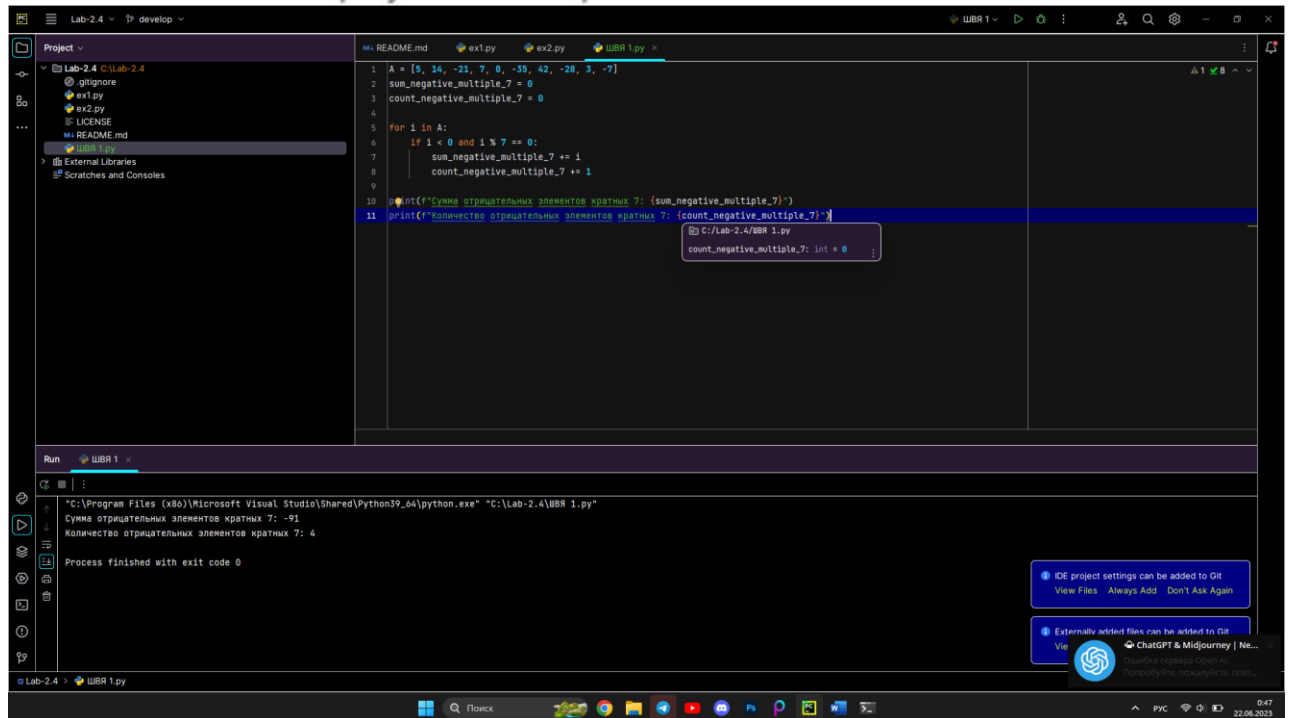


Рисунок 5 – Программа и ее результат

6. Индивидуальное задание №2.

Создал новый файл под названием *ind2.py*

Вариант 11.

Условие задания:

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

1. номер максимального по модулю элемента списка;
2. сумму элементов списка, расположенных после первого положительного элемента.

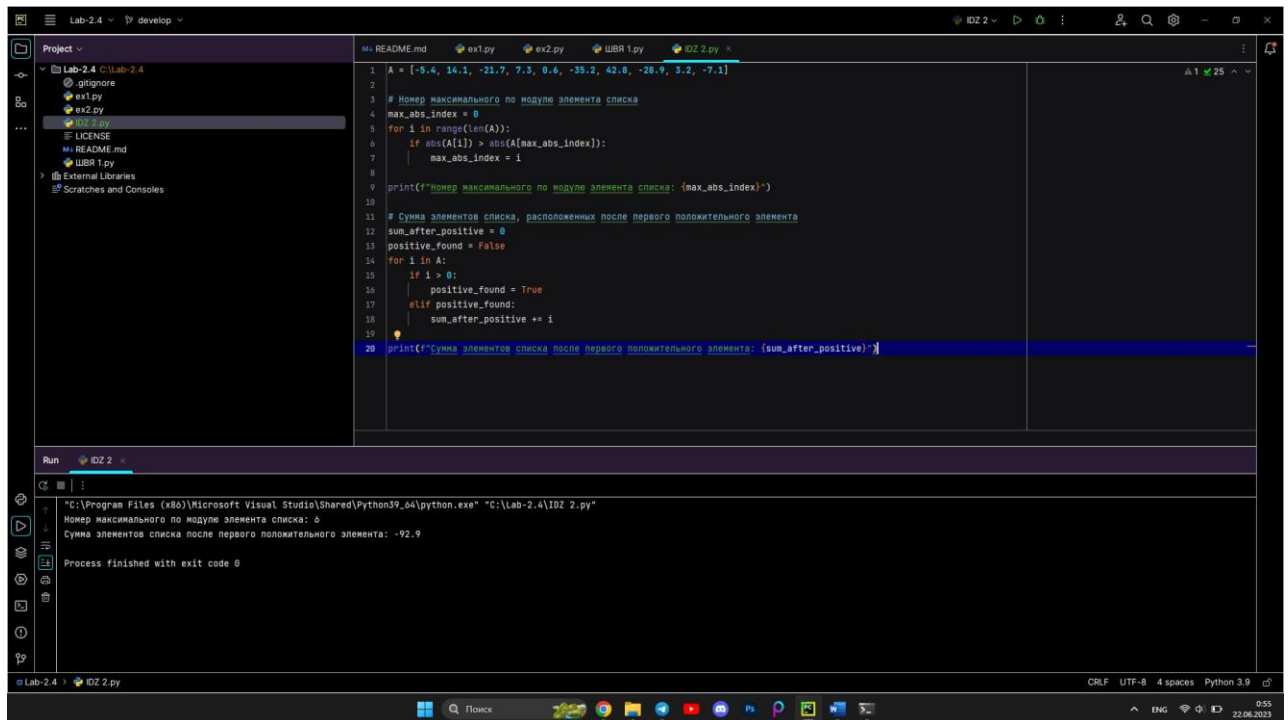


Рисунок 6 – Результат программы

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

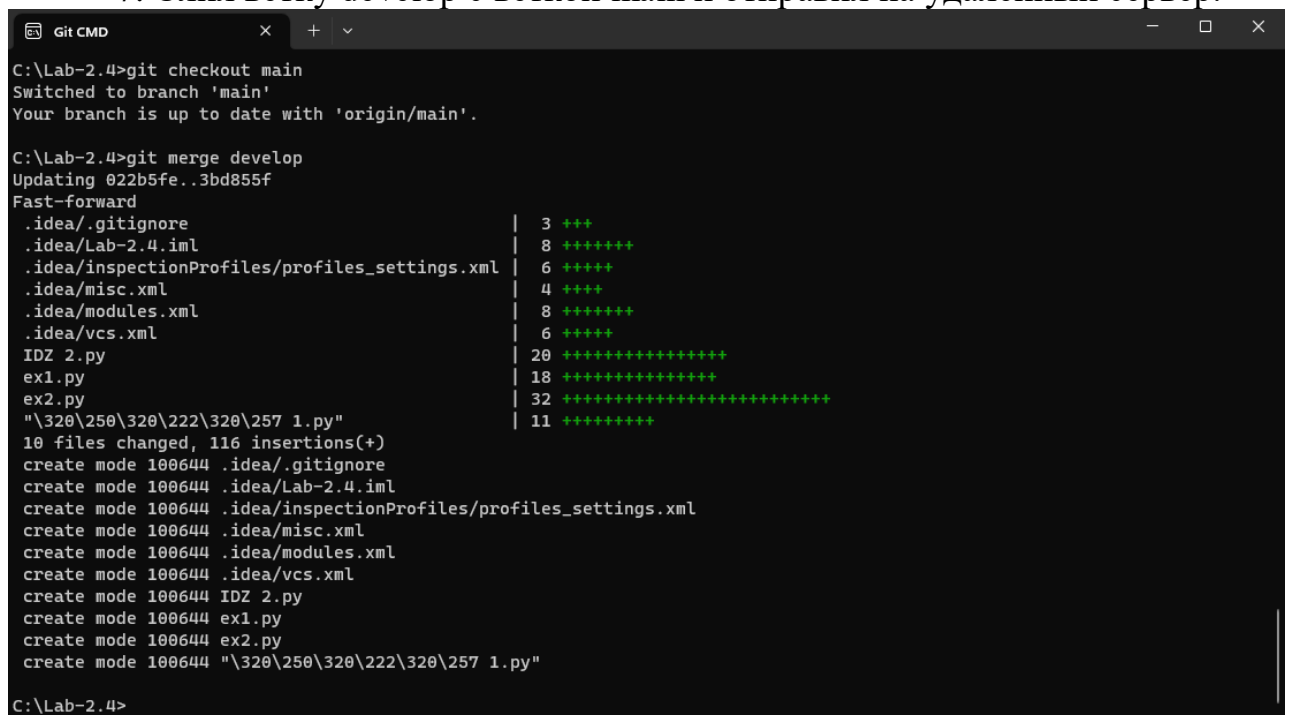


Рисунок 7 – Слияние ветки develop с основной веткой main

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 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:

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.