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

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
дисциплины
«Программирование на Python»
Вариант 19

Выполнил:
Поляков Никита Александрович
2 курс, группа ИВТ-б-о-24-1,
09.03.01 «Информатика и вычислительная техника»,
направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р.А., доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2025 г.

Тема: Условные операторы и циклы в языке Python

Цель: приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3.x if, while, for, break и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Практическая часть:

Перед началом работы был создан новый репозиторий для лабораторной работы №3:

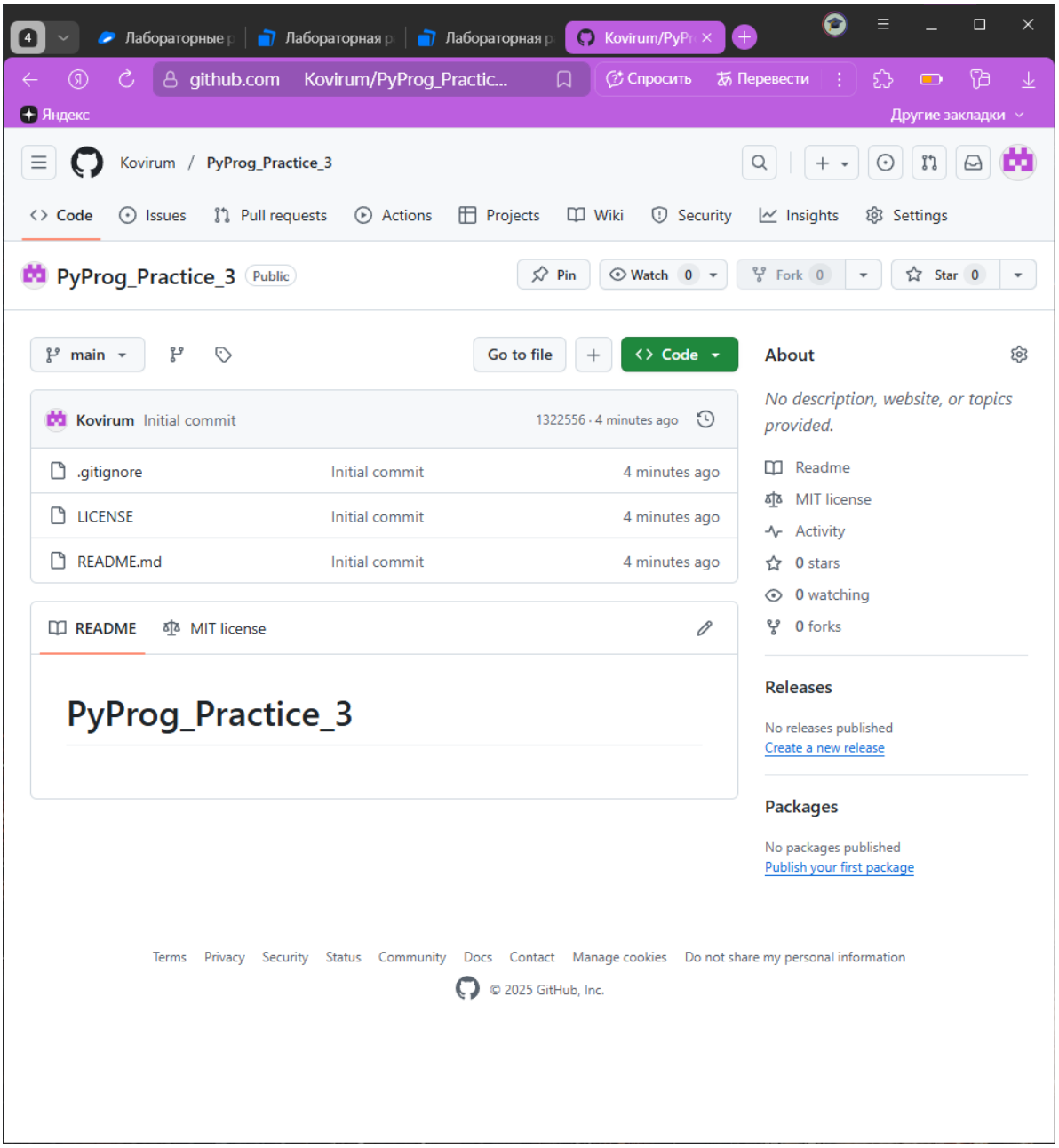

Рисунок 1. Страница созданного репозитория

Ссылка на репозиторий: https://github.com/Kovirum/PyProg_Practice_3

Далее репозиторий был клонирован на компьютер и начата работа над заданиями с соблюдением принципов модели ветвления git-flow, а также правил оформления кода PEP-8:

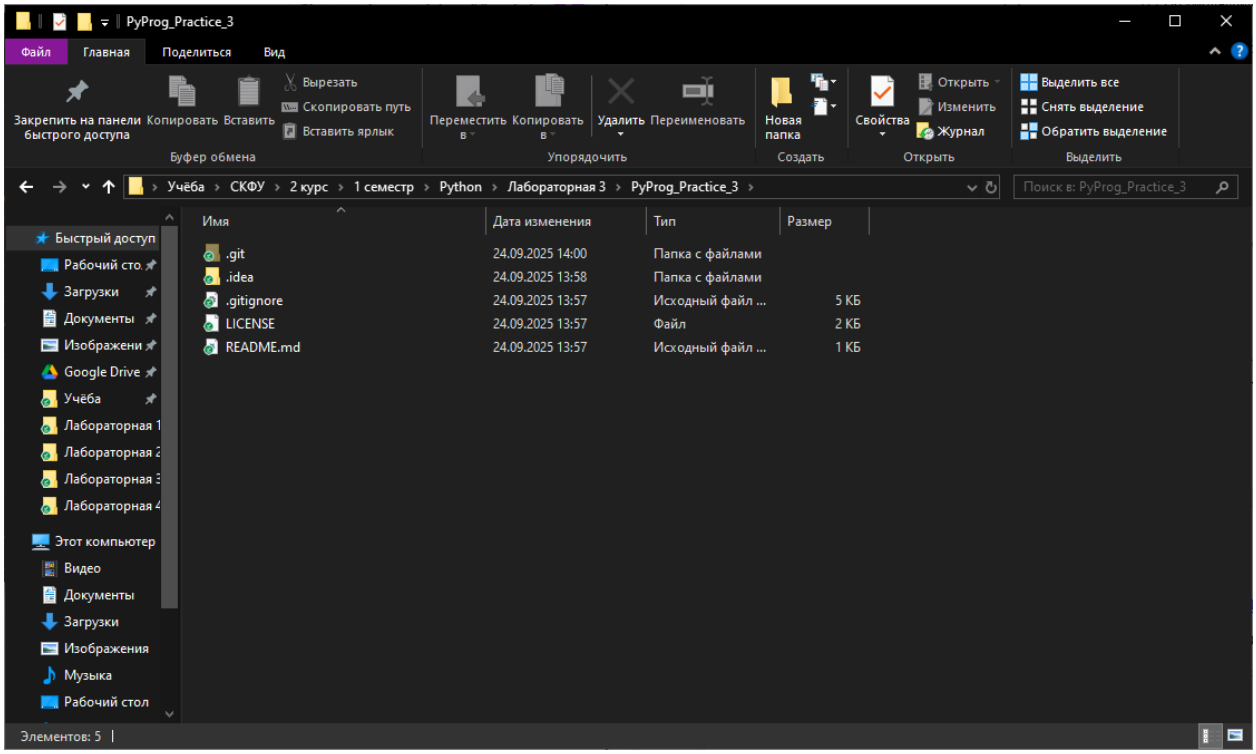

Рисунок 2. Директория локального репозитория

Для выполнения задания проработки всех примеров из методических указаний были организованы отдельные ветки с наименованием «feature/exampleN», где N – номер примера. Каждый отдельный пример реализован в собственном модуле с наименованием «exampleN.py», где N – номер примера.

Пример 1:

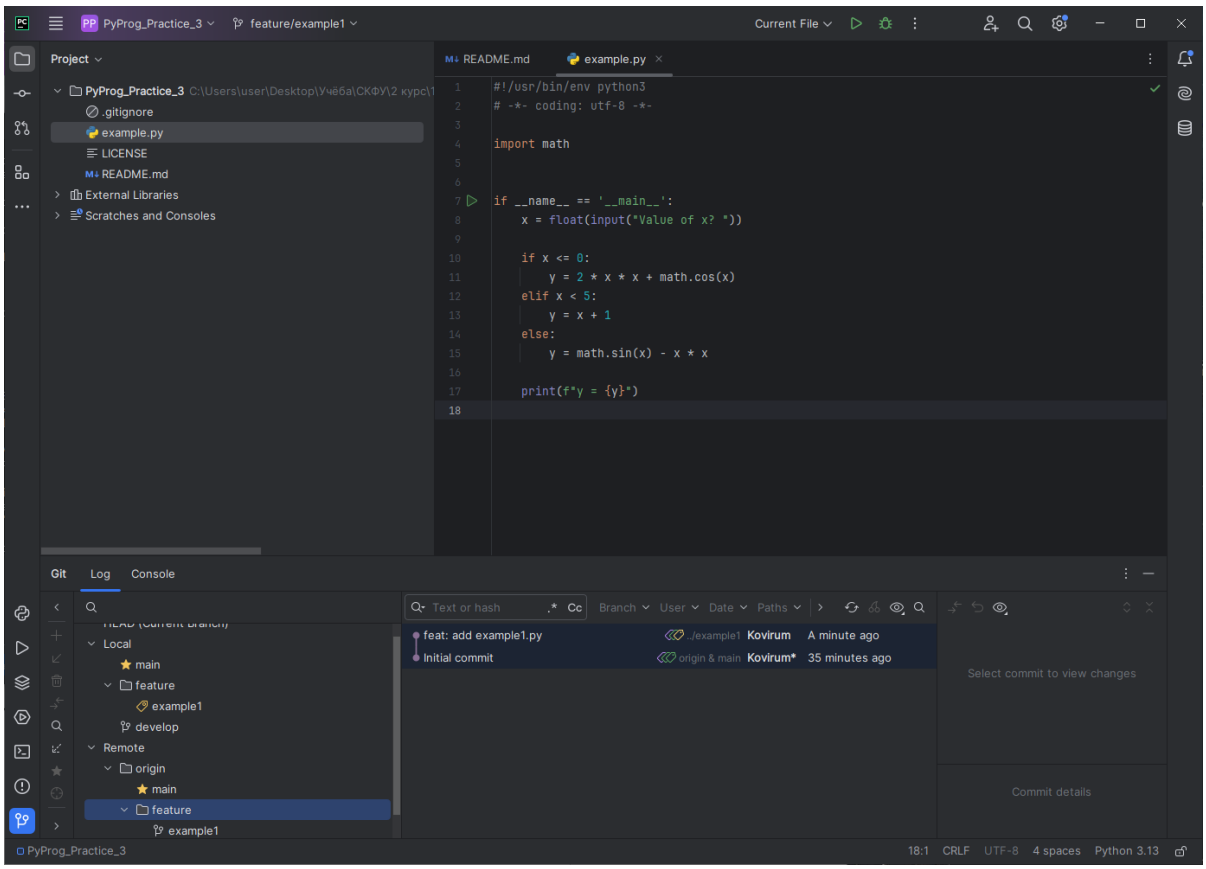

Рисунок 3. Код и git-информация примера 1

Пример 2:

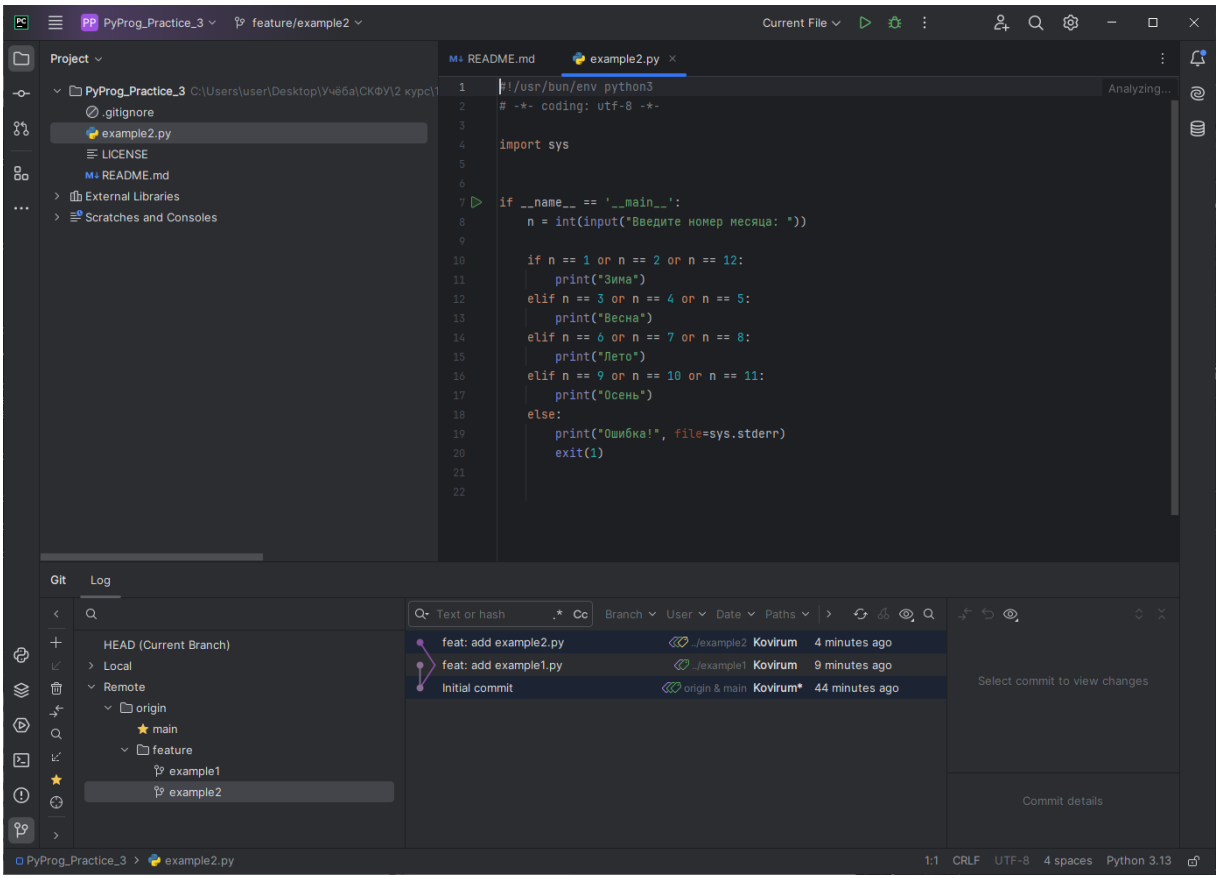

Рисунок 4. Код и git-информация примера 2

Пример 3:

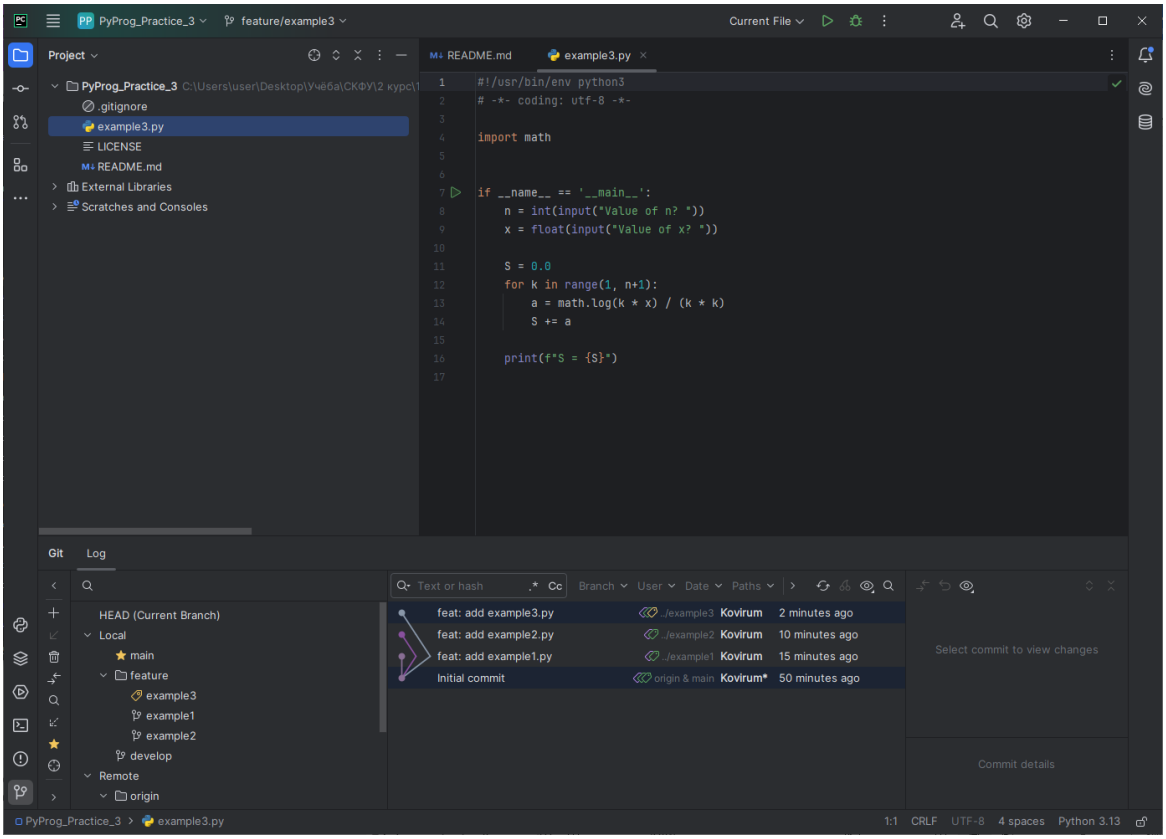

Рисунок 5. Код и git-информация примера 3

Пример 4:

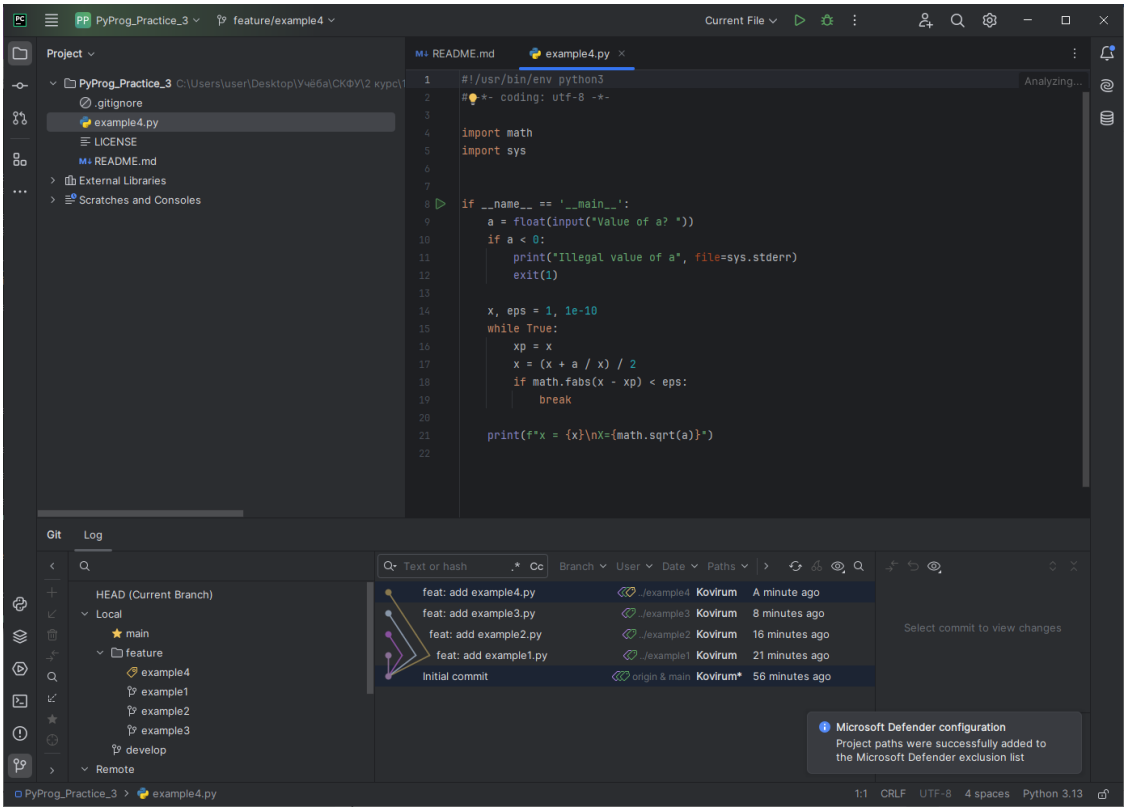

Рисунок 6. Код и git-информация примера 4

Пример 5:

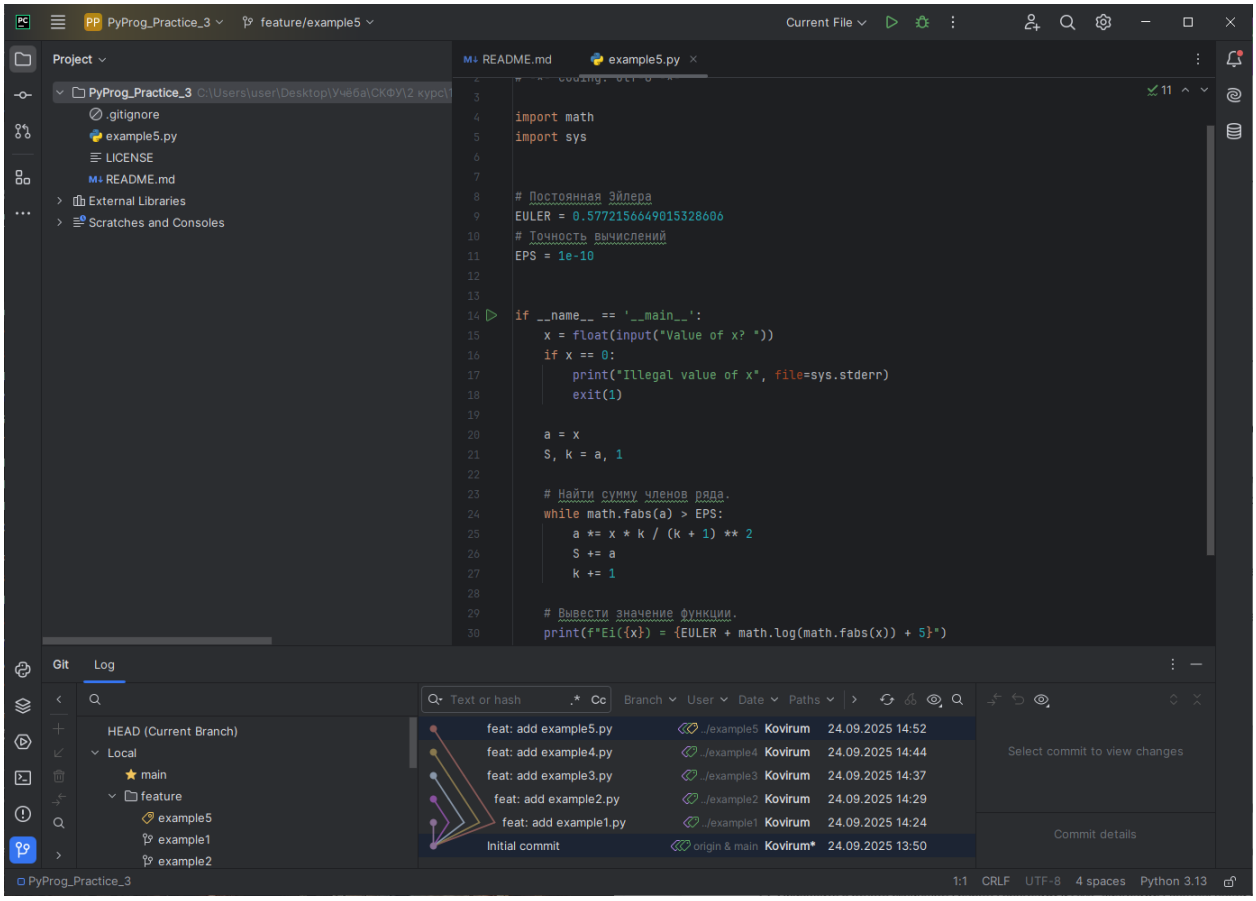

Рисунок 7. Код и git-информация примера 5

Далее были выполнены индивидуальные задания для варианта 19:

Задание 1. Дано целое число C, такое что |C| < 9. Вывести это число в словесной форме, учитывая его знак.

Код программы:

```
#!usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    c = int(input("Введите число C такое, что |C| < 9: "))

    if not abs(c) < 9:
        print("Число C должно соответствовать условию: |C| < 9",
file=sys.stderr)
        exit(1)

    result_str = ""

    if c < 0:
        result_str += "минус "
```

```
    match abs(c):
        case 0:
            result_str += "ноль"
        case 1:
            result_str += "один"
        case 2:
            result_str += "два"
        case 3:
            result_str += "три"
        case 4:
            result_str += "четыре"
        case 5:
            result_str += "пять"
        case 6:
            result_str += "шесть"
        case 7:
            result_str += "семь"
        case 8:
            result_str += "восемь"
        case 9:
            result_str += "девять"

    print(result_str)
)
```

Демонстрация работы программы:

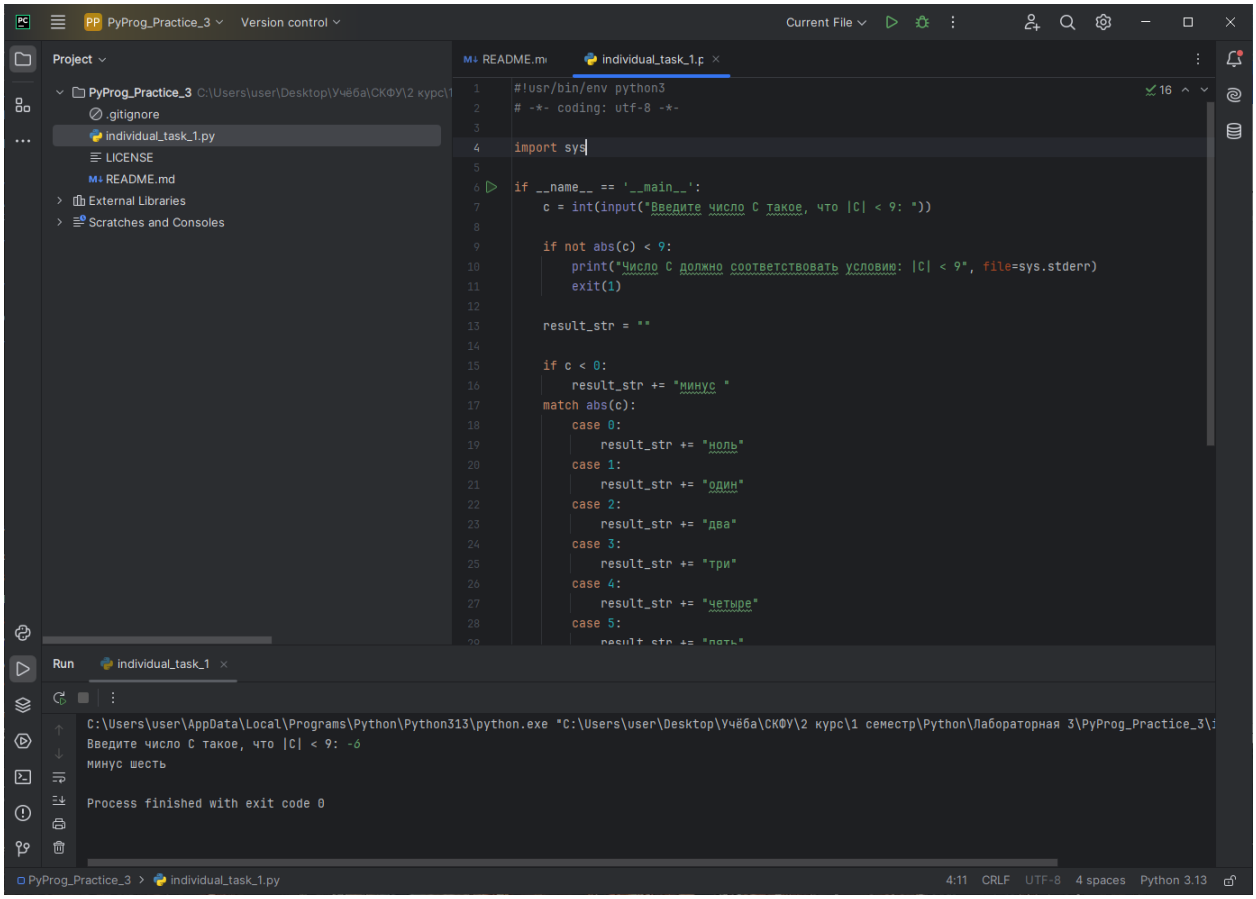

Рисунок 8. Демонстрация работы программы индивидуального задания 1

UML-диаграмма деятельности:

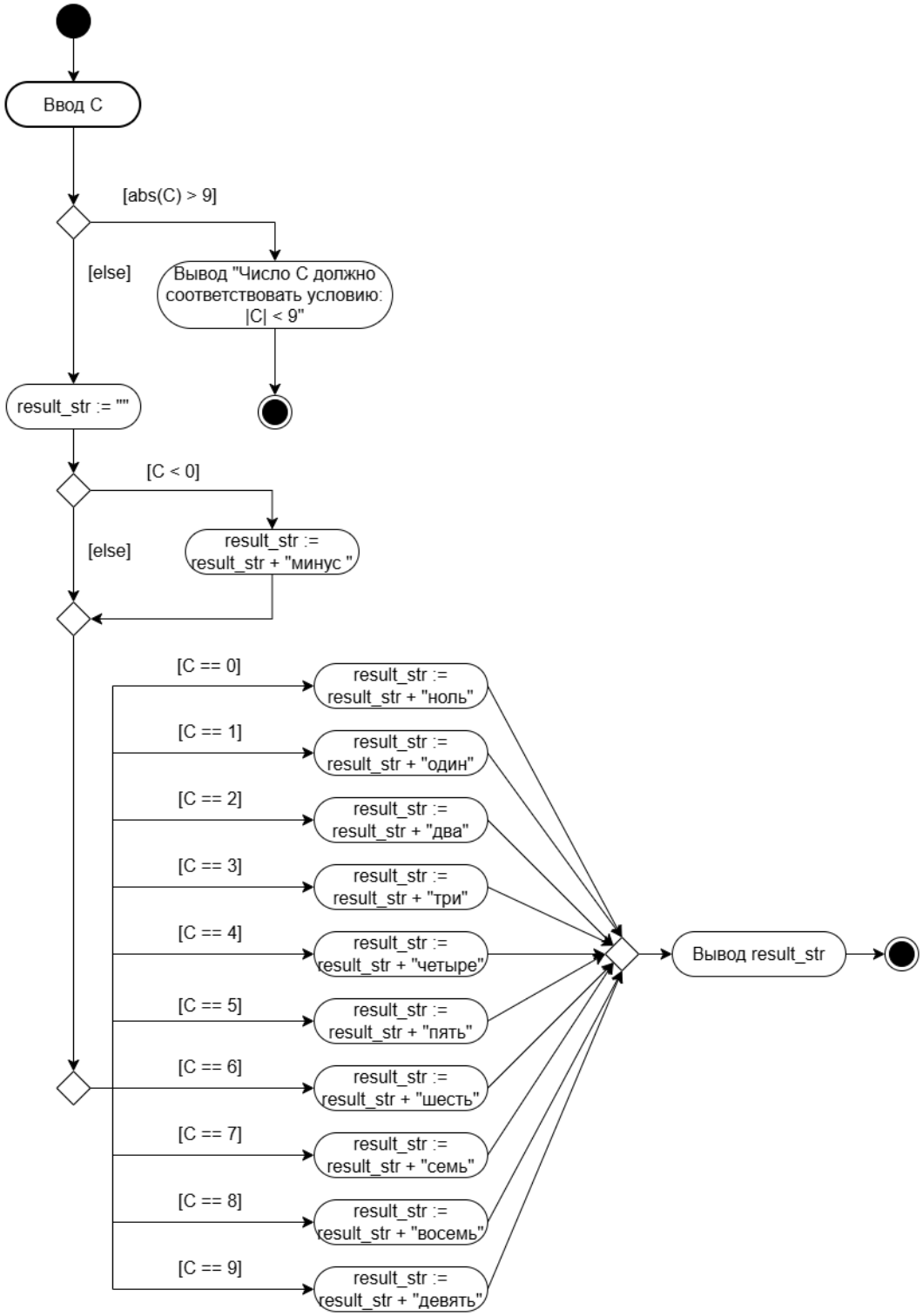

Рисунок 9. UML-диаграмма для индивидуального задания 1

Задание 2. Какая из точек A(a1, a2) или B(b1, b2) находится дальше от центра координат?

Код программы:

```
#!usr/bin/env python3
# -*- coding: utf-8 -*-
```

```
import math

if __name__ == '__main__':
    a1, a2 = map(int, input("Укажите координаты точки A (a1,
a2): ").split())
    b1, b2 = map(int, input("Укажите координаты точки B (b1,
b2): ").split())

    OA_distance = math.sqrt(math.pow(a1 - 0, 2) + math.pow(a2 -
0, 2))
    OB_distance = math.sqrt(math.pow(b1 - 0, 2) + math.pow(b2 -
0, 2))

    if OA_distance > OB_distance:
        print("Точка A находится дальше от начала координат")
    elif OA_distance == OB_distance:
        print("Точки A и B находятся на одинаковом удалении от
начала координат")
    else:
        print("Точка B находится дальше от начала координат")
```

Демонстрация работы программы:

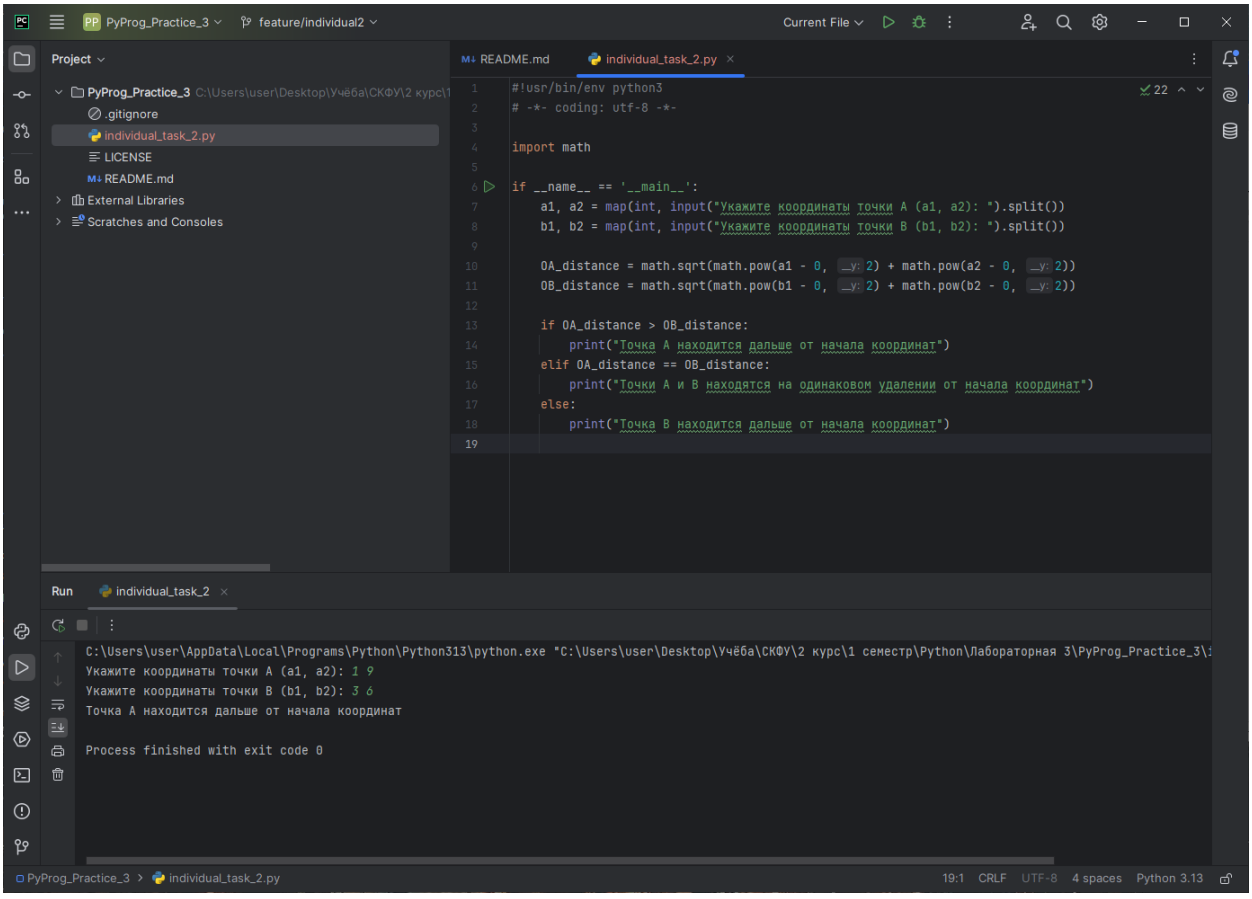

Рисунок 10. Демонстрация работы программы индивидуального задания 2

UML-диаграмма:

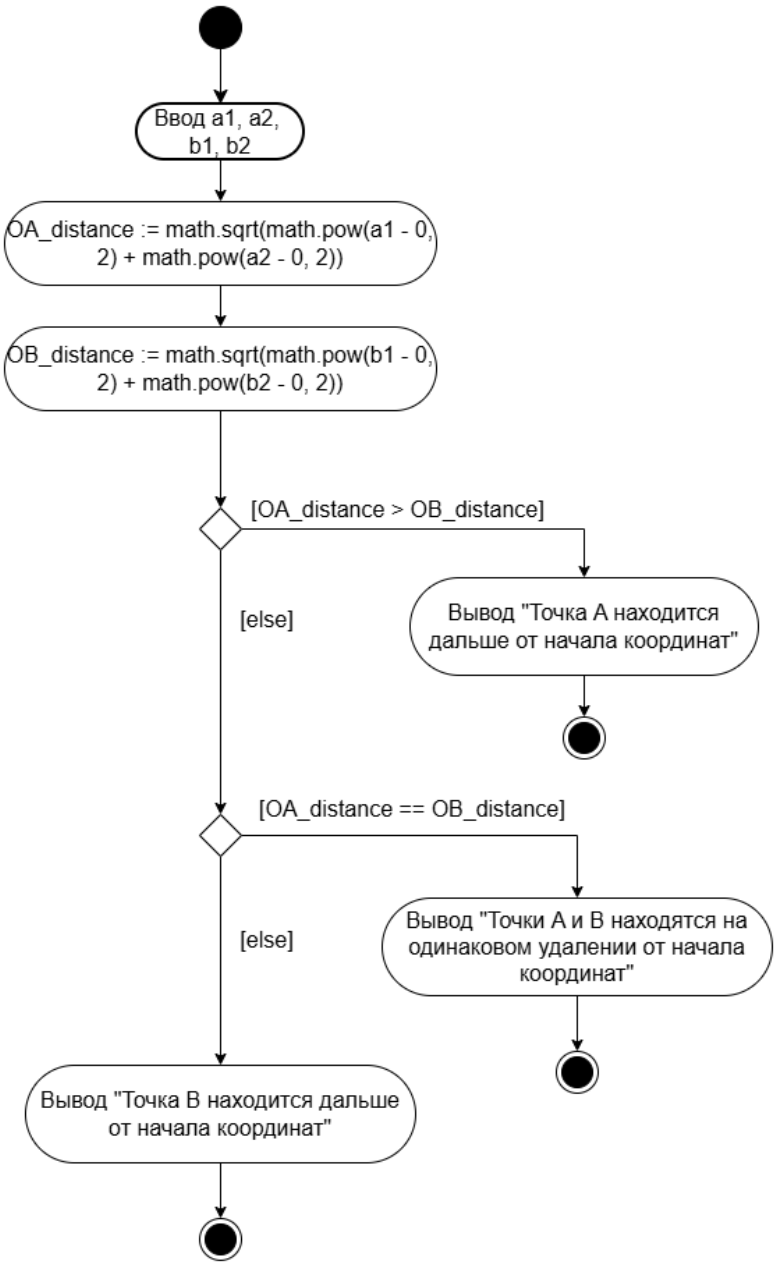

Рисунок 11. UML-диаграмма для индивидуального задания 2

Задание 3. У гусей и кроликов вместе 64 лапы. Сколько могло быть кроликов и гусей (указать все сочетания, которые возможны)

Код программы:

```
#!usr/bin/env python3
# -*- coding: utf-8 -*-

TOTAL_PAWS = 64

if __name__ == '__main__':
    for c, i in enumerate(range(0, TOTAL_PAWS + 1, 4), 1):
        goose_count = (TOTAL_PAWS - i) // 2
        rabbit_count = (TOTAL_PAWS - goose_count * 2) // 4
```

```
        print(f"[#{c}] Гусей - {goose_count}, Кроликов -
{rabbit_count}. "
              f"Лап: {goose_count * 2} + {rabbit_count * 4} =
{goose_count * 2 + rabbit_count * 4}")
```

Демонстрация работы программы:

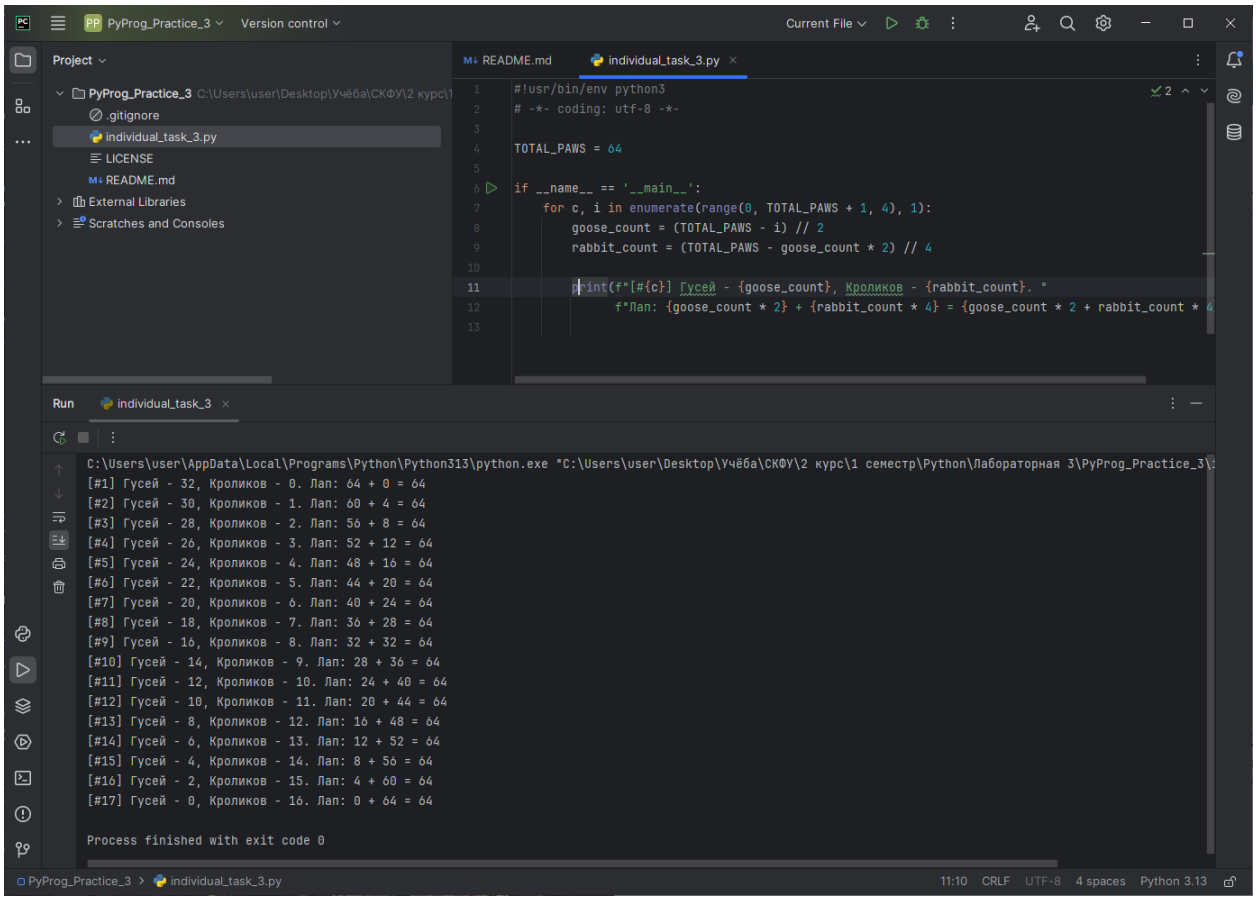

Рисунок 12. Демонстрация работы программы индивидуального задания 3

UML-диаграмма:

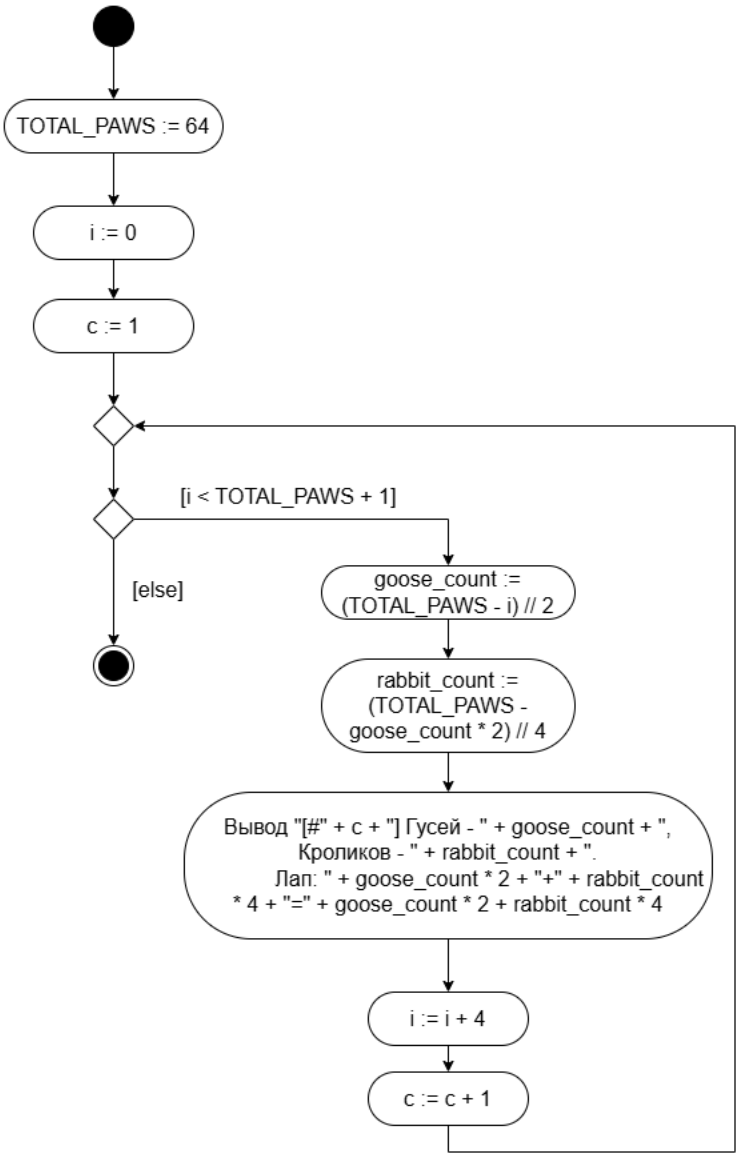

Рисунок 13. UML-диаграмма для индивидуального задания 3

Далее, после выполнения всех примеров и индивидуальных заданий, необходимо внести все изменения в основную ветку согласно модели ветвления git-flow.

Для этого сначала все feature-ветки были слиты в develop:

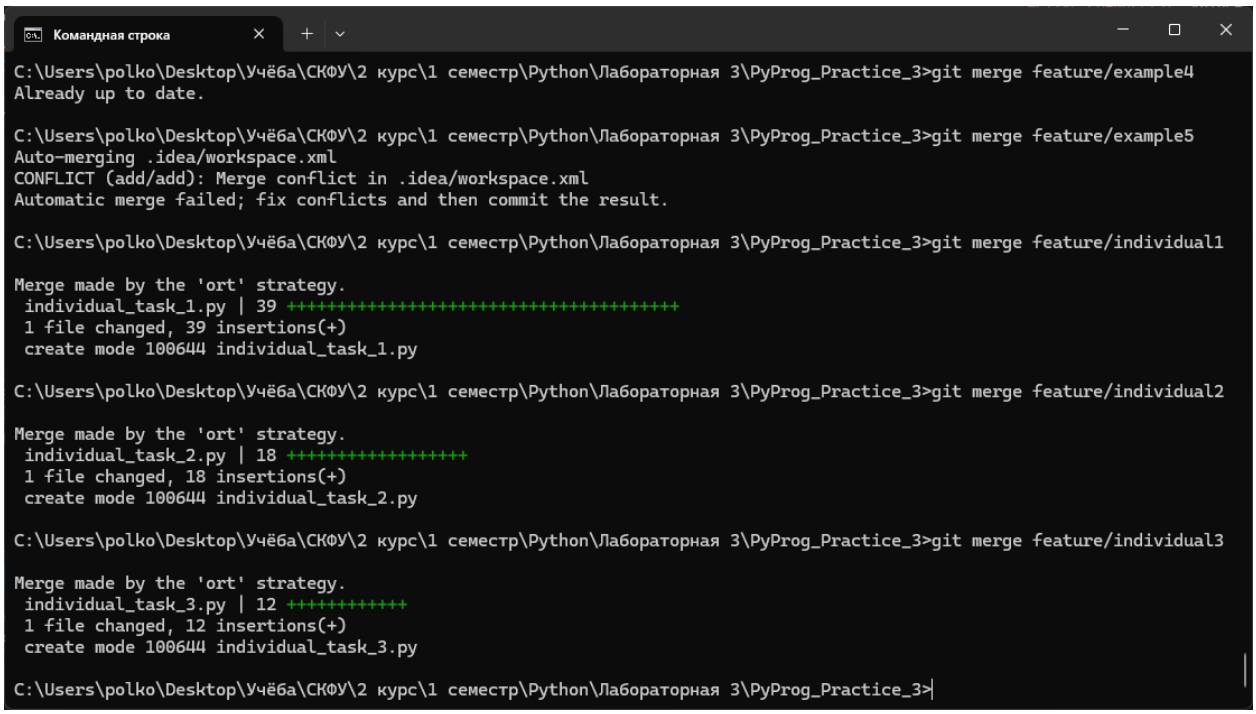

Рисунок 14. Процесс сливания feature-веток в develop

Далее была подготовлена release-ветка: