

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:

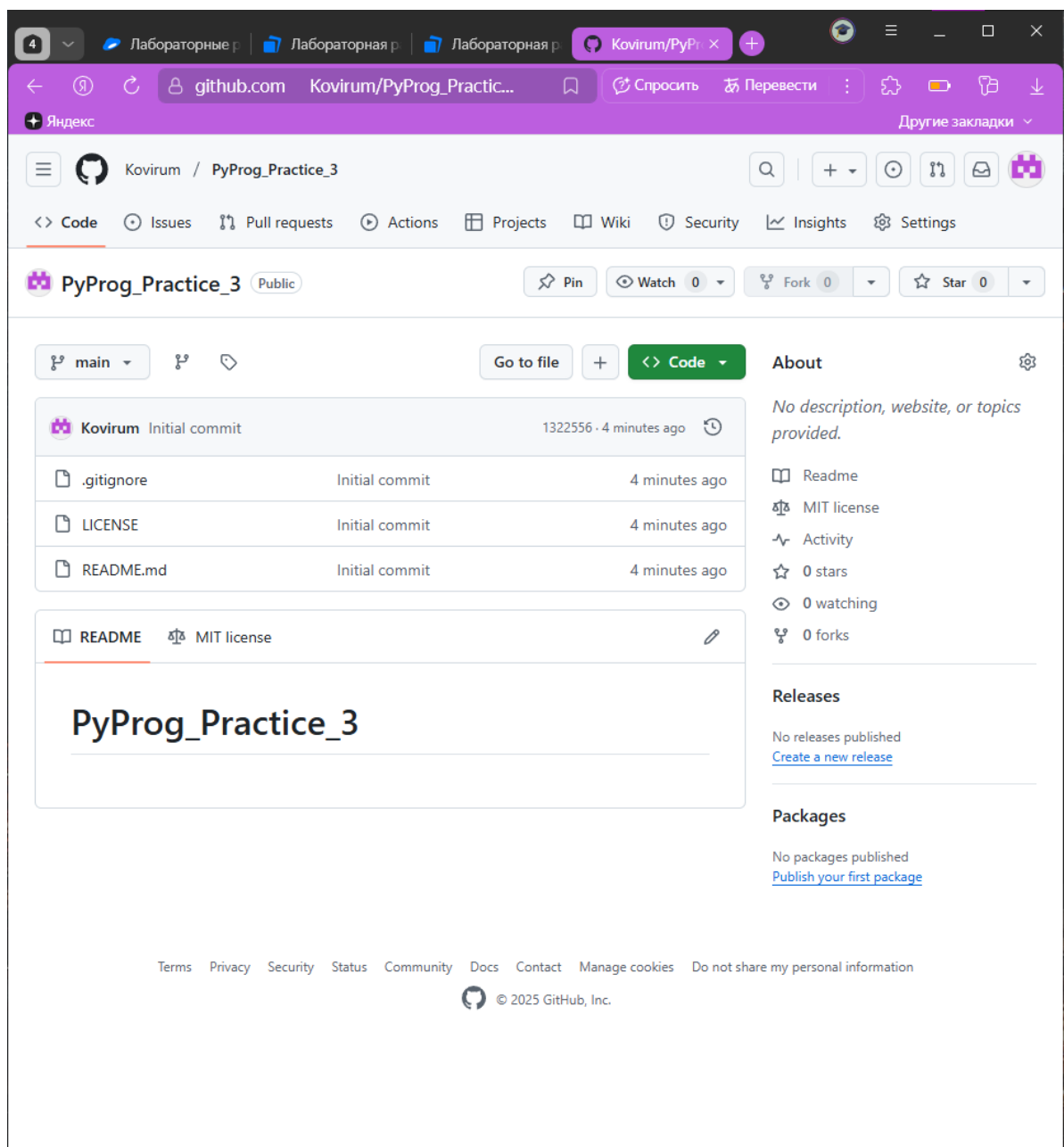


Рисунок 1. Страница созданного репозитория

Ссылка на репозиторий: [https://github.com/Kovirum/PyProg\\_Practice\\_3](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:

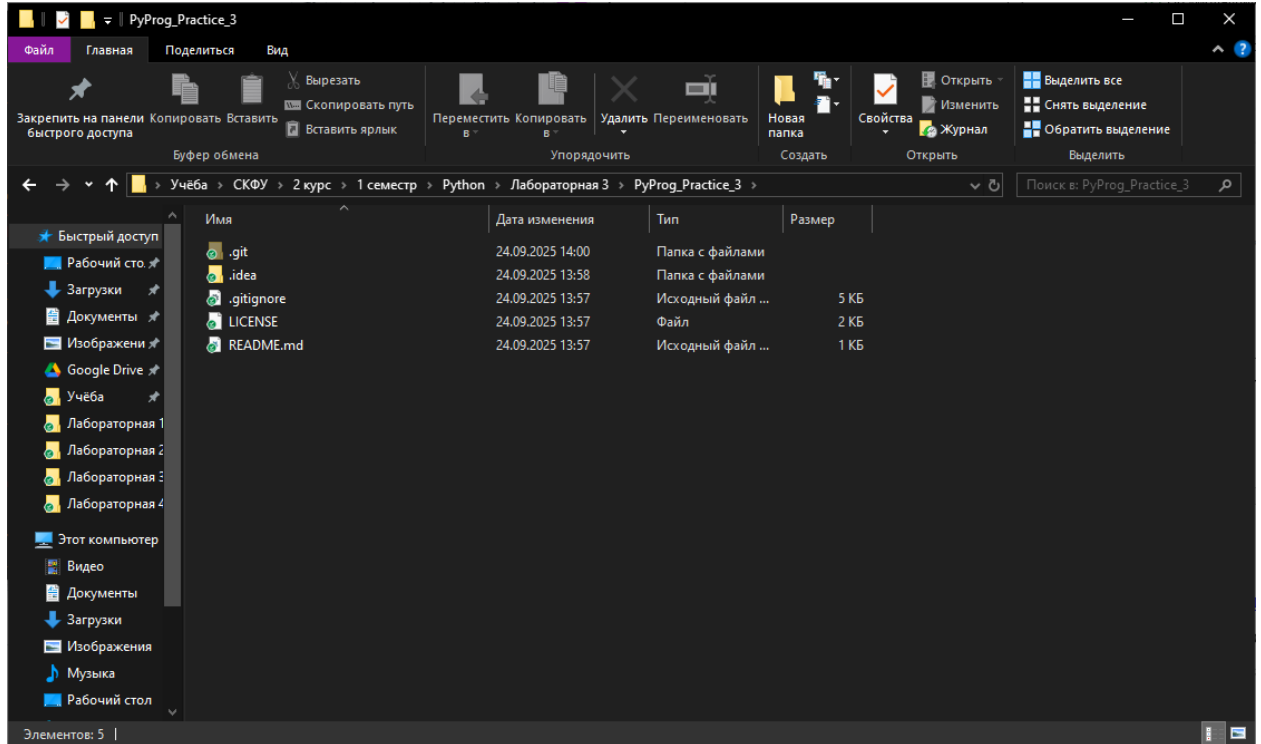


Рисунок 2. Директория локального репозитория

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:

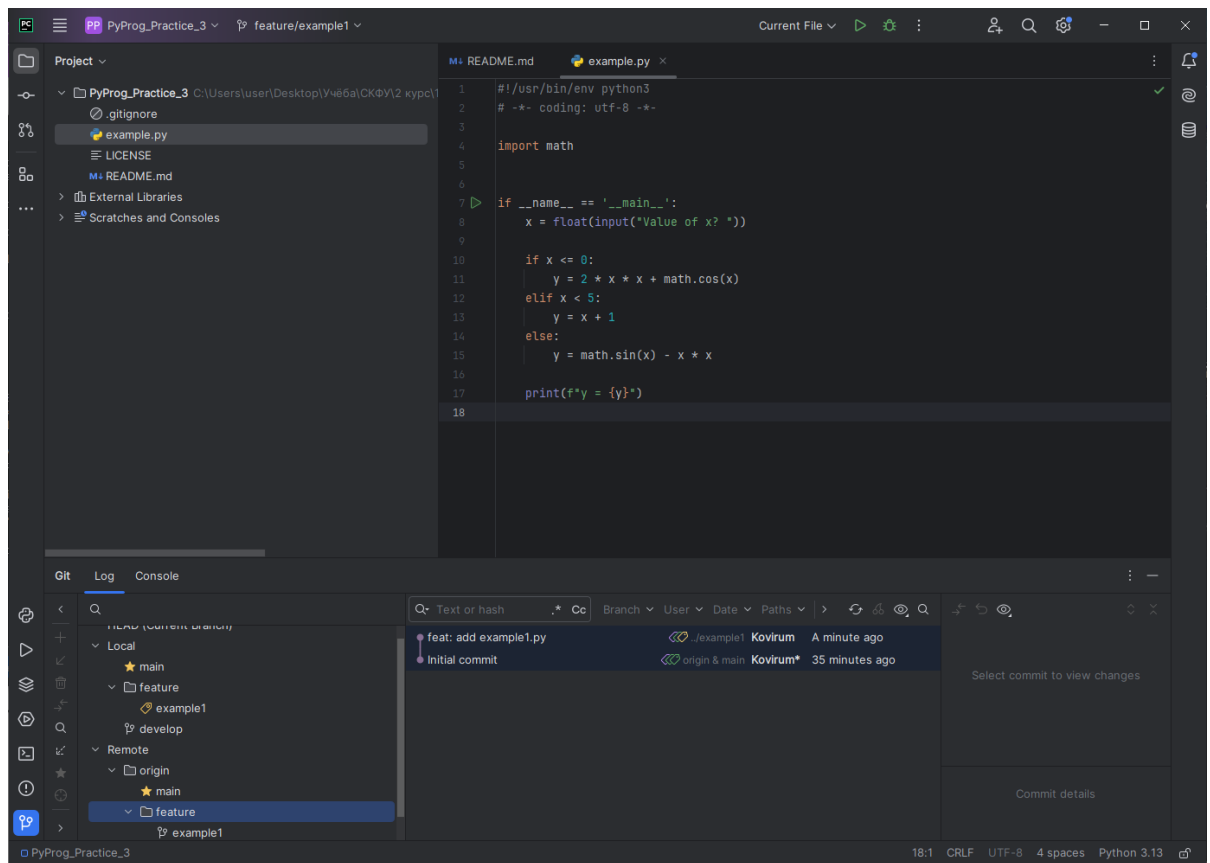


Рисунок 3. Код и git-информация примера 1

Пример 2:

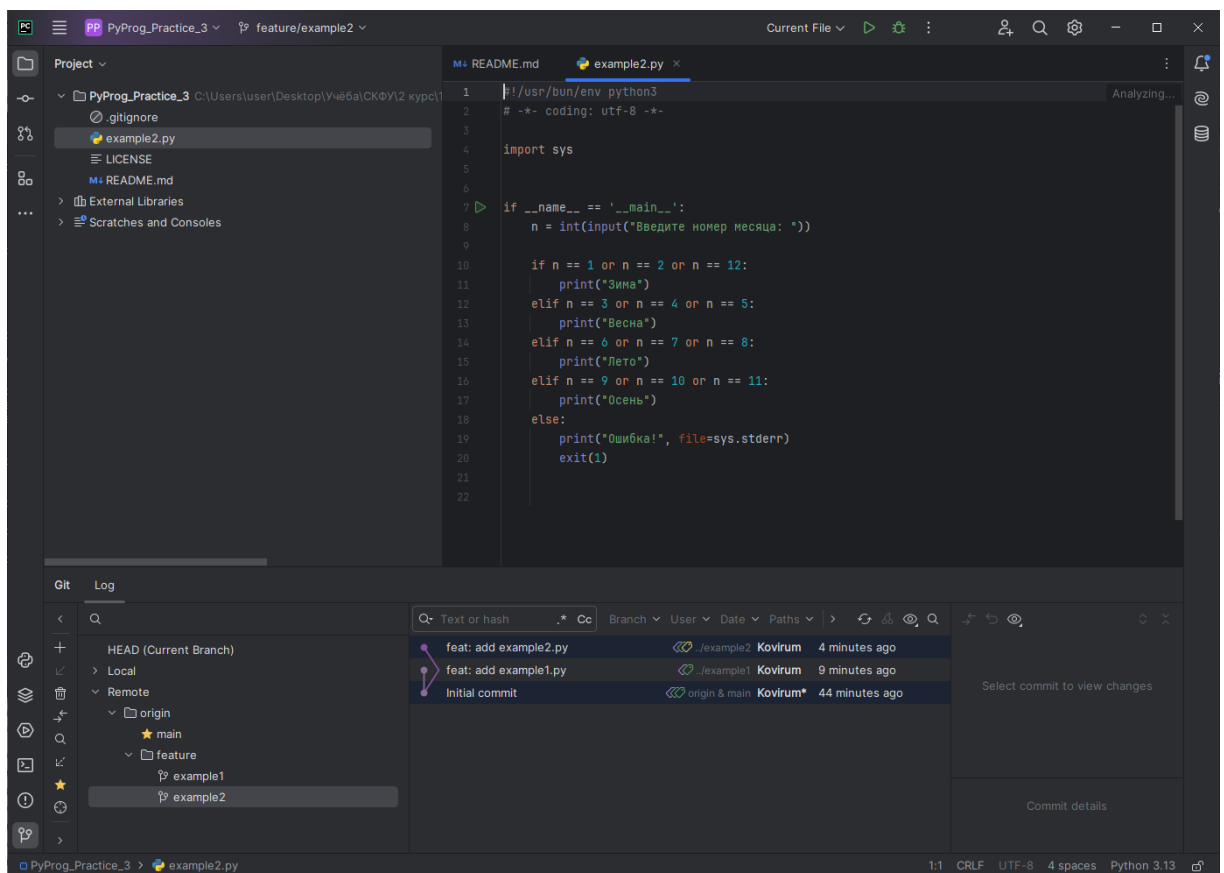


Рисунок 4. Код и git-информация примера 2

### Пример 3:

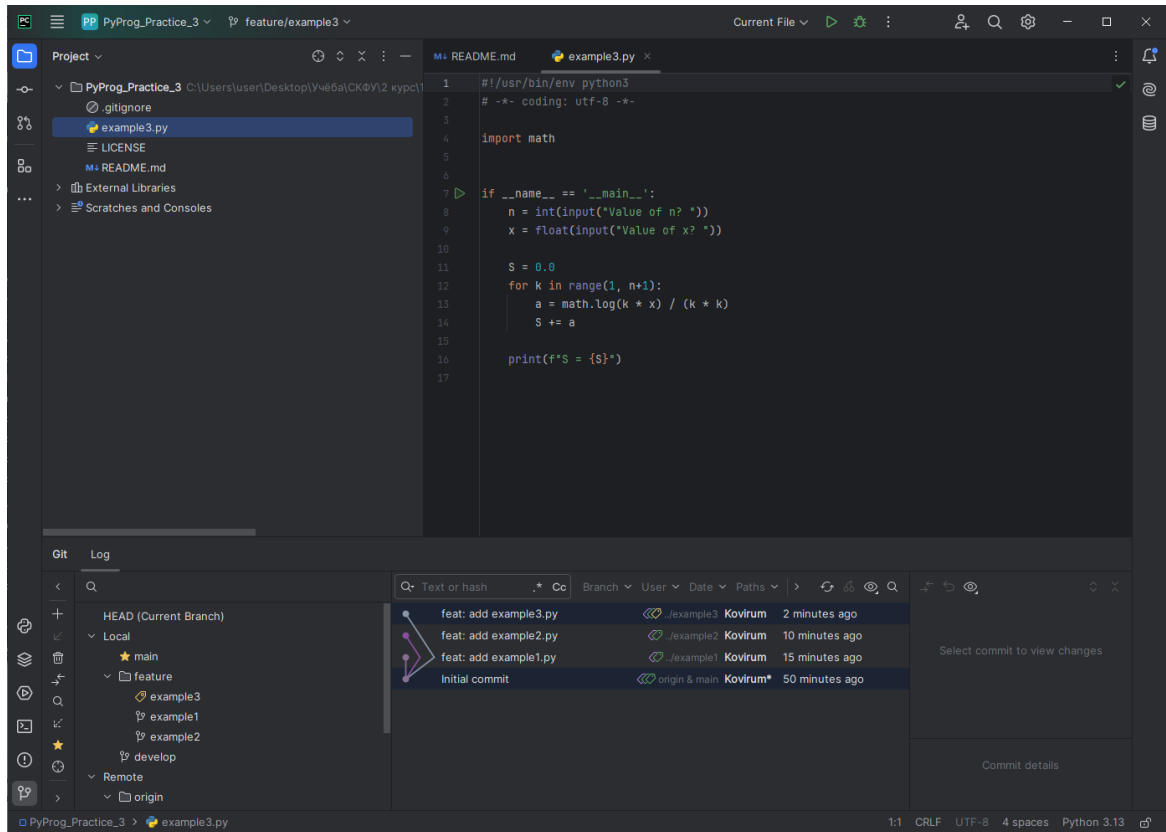


Рисунок 5. Код и git-информация примера 3

### Пример 4:

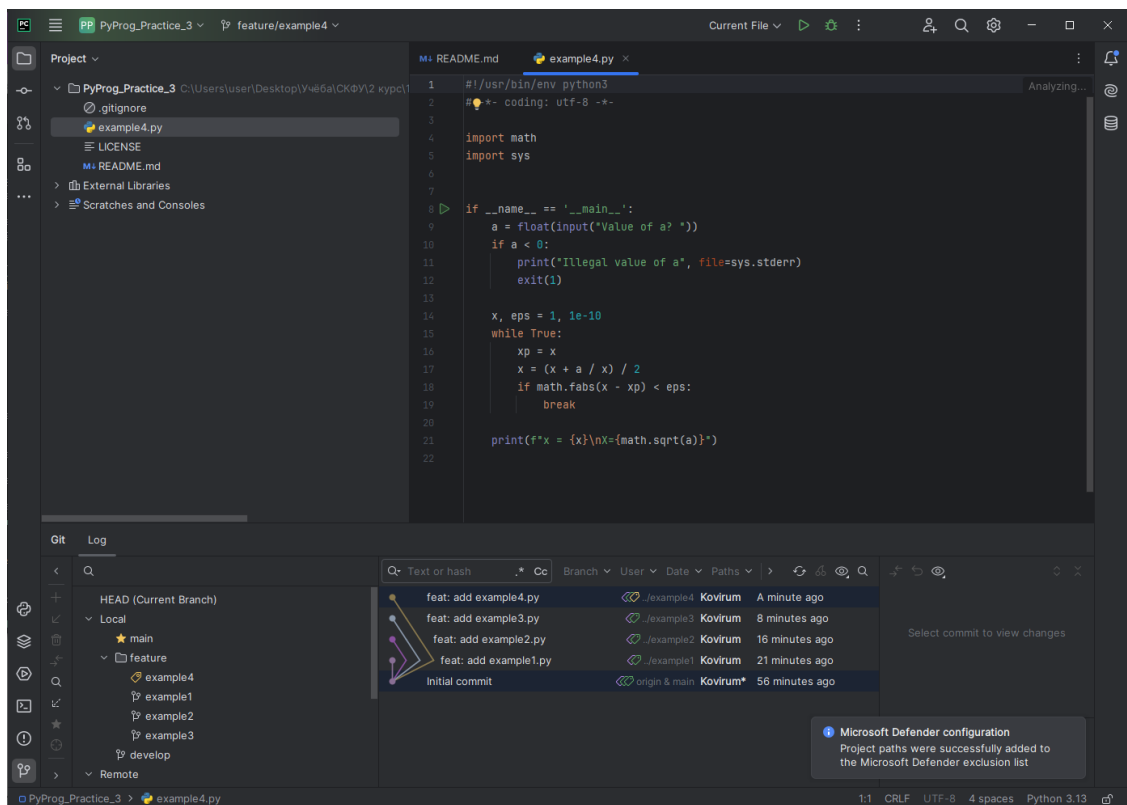


Рисунок 6. Код и git-информация примера 4

## Пример 5:

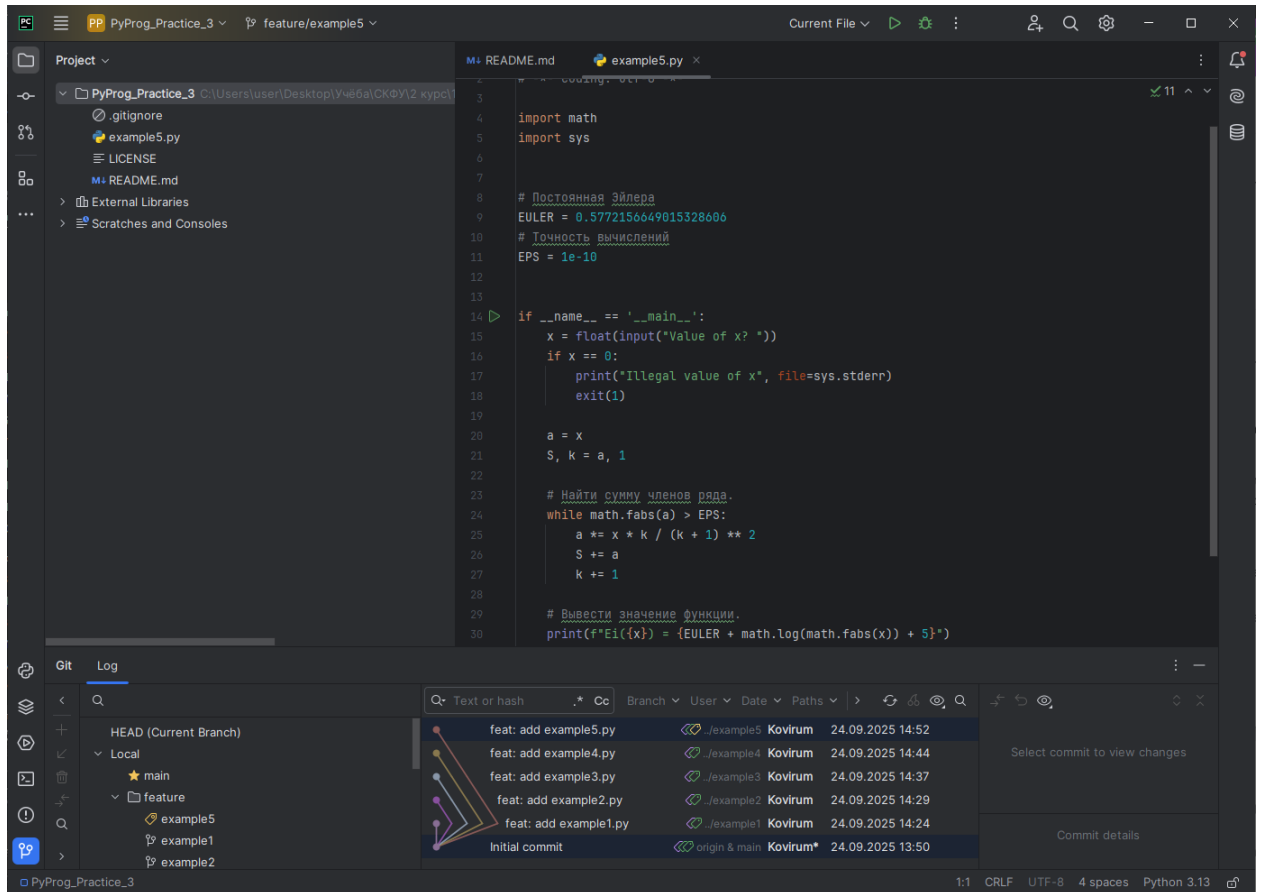


Рисунок 7. Код и git-информация примера 5

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:

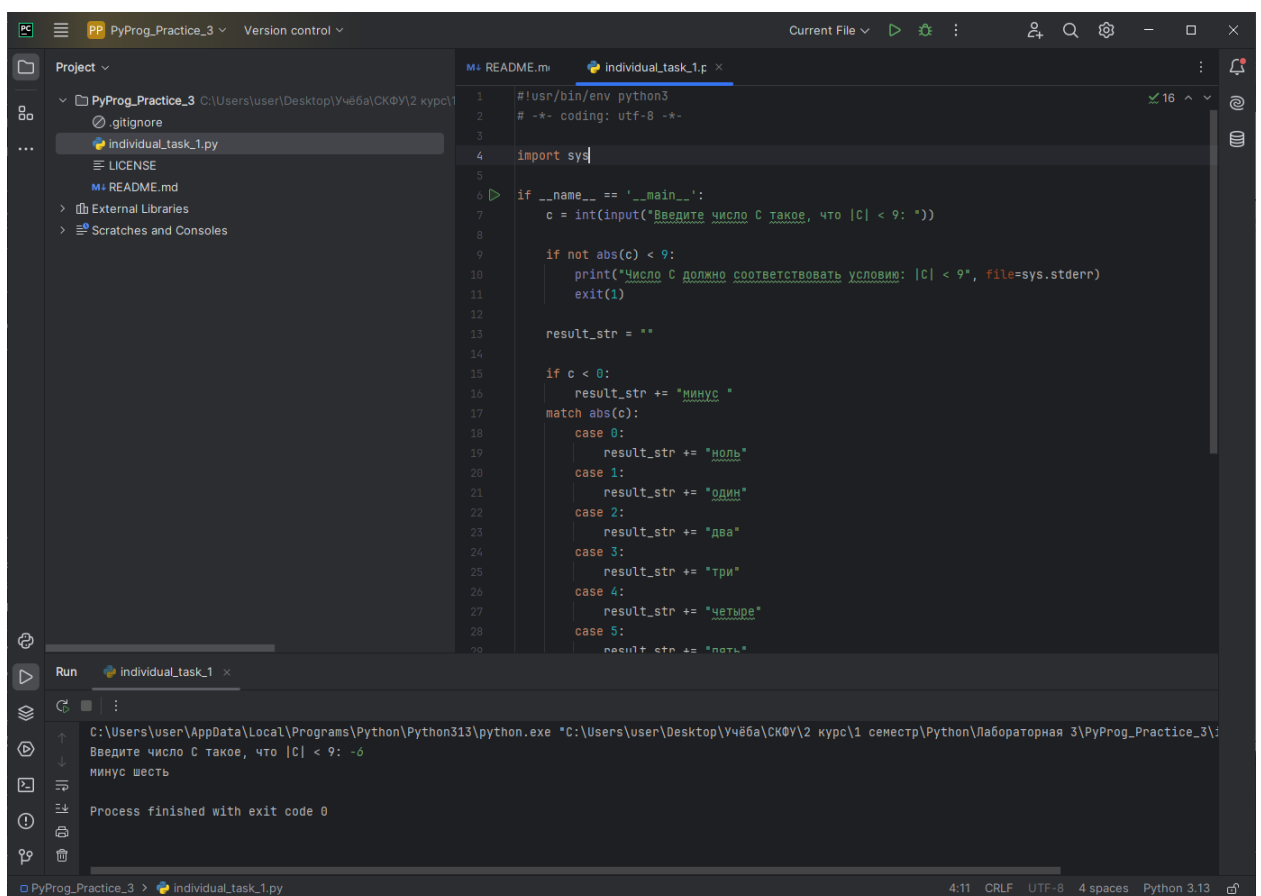


Рисунок 8. Демонстрация работы программы индивидуального задания 1

UML-диаграмма деятельности:

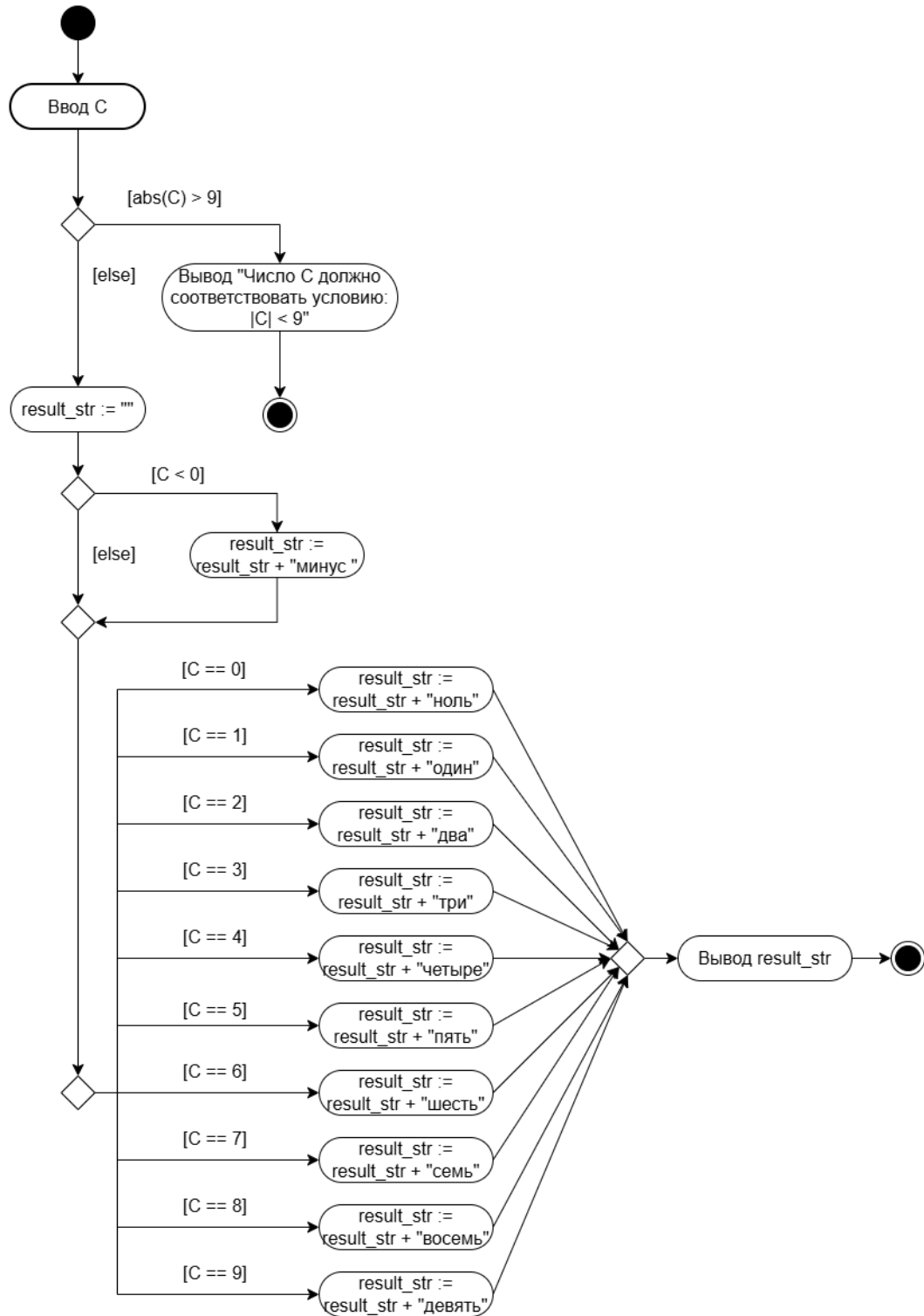


Рисунок 9. UML-диаграмма для индивидуального задания 1

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:

The screenshot displays the PyCharm IDE interface. The main editor window shows the Python script from the previous block. The 'Run' console at the bottom shows the execution process:

```

C:\Users\user\AppData\Local\Programs\Python\Python313\python.exe "C:\Users\user\Desktop\Учеба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3\
Укажите координаты точки A (a1, a2): 1 9
Укажите координаты точки B (b1, b2): 3 6
Точка A находится дальше от начала координат
Process finished with exit code 0

```

Рисунок 10. Демонстрация работы программы индивидуального задания 2

UML-диаграмма:

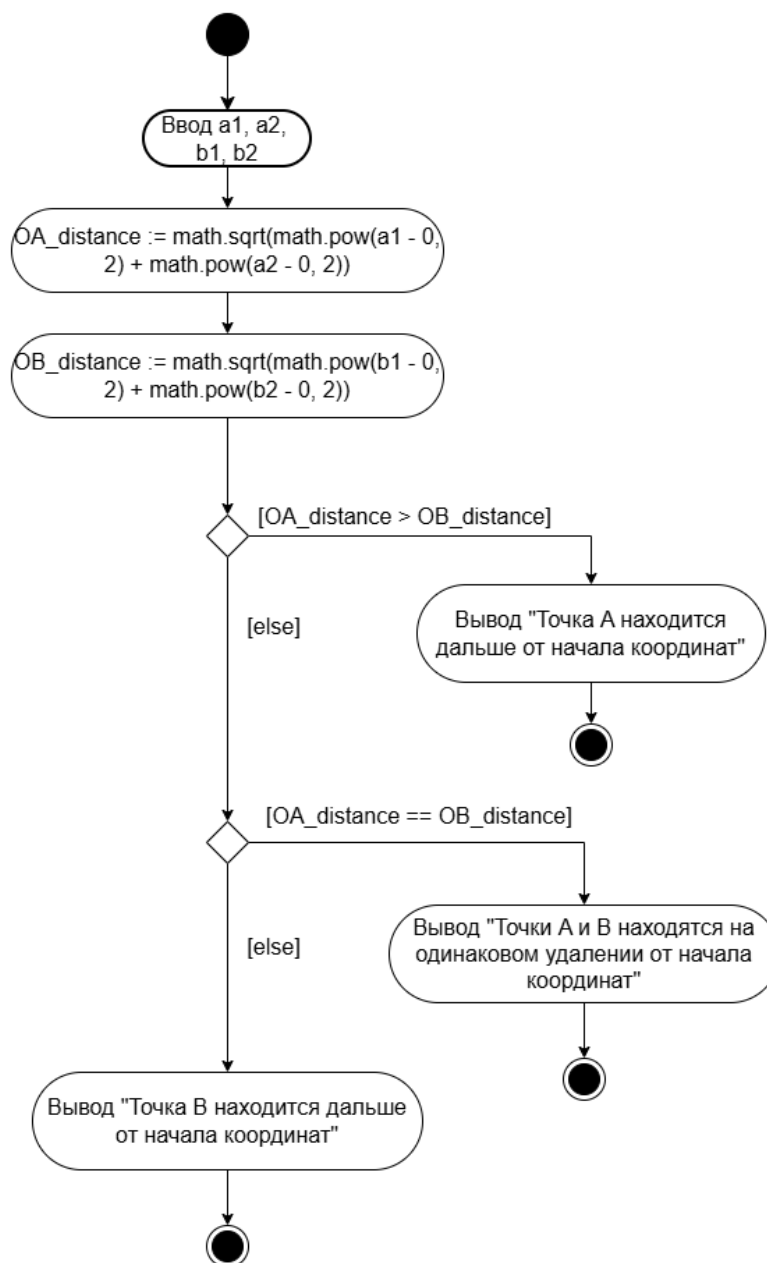


Рисунок 11. UML-диаграмма для индивидуального задания 2

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:

The screenshot shows a Python IDE with a project named 'PyProg\_Practice\_3'. The file 'individual\_task\_3.py' is open, showing the following code:

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 TOTAL_PAWS = 64
5
6 if __name__ == '__main__':
7     for c, i in enumerate(range(0, TOTAL_PAWS + 1, 4), 1):
8         goose_count = (TOTAL_PAWS - i) // 2
9         rabbit_count = (TOTAL_PAWS - goose_count * 2) // 4
10
11         print(f"[#{c}] Гусей - {goose_count}, Кроликов - {rabbit_count}. "
12               f"Лап: {goose_count * 2} + {rabbit_count * 4} = {goose_count * 2 + rabbit_count * 4}")
13

```

The 'Run' window shows the output of the program, which lists 17 iterations of the loop, showing the number of geese, rabbits, and the total number of legs for each iteration. The output is as follows:

```

C:\Users\user\AppData\Local\Programs\Python\Python313\python.exe "C:\Users\user\Desktop\Учѐба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3\
[#1] Гусей - 32, Кроликов - 0. Лап: 64 + 0 = 64
[#2] Гусей - 30, Кроликов - 1. Лап: 60 + 4 = 64
[#3] Гусей - 28, Кроликов - 2. Лап: 56 + 8 = 64
[#4] Гусей - 26, Кроликов - 3. Лап: 52 + 12 = 64
[#5] Гусей - 24, Кроликов - 4. Лап: 48 + 16 = 64
[#6] Гусей - 22, Кроликов - 5. Лап: 44 + 20 = 64
[#7] Гусей - 20, Кроликов - 6. Лап: 40 + 24 = 64
[#8] Гусей - 18, Кроликов - 7. Лап: 36 + 28 = 64
[#9] Гусей - 16, Кроликов - 8. Лап: 32 + 32 = 64
[#10] Гусей - 14, Кроликов - 9. Лап: 28 + 36 = 64
[#11] Гусей - 12, Кроликов - 10. Лап: 24 + 40 = 64
[#12] Гусей - 10, Кроликов - 11. Лап: 20 + 44 = 64
[#13] Гусей - 8, Кроликов - 12. Лап: 16 + 48 = 64
[#14] Гусей - 6, Кроликов - 13. Лап: 12 + 52 = 64
[#15] Гусей - 4, Кроликов - 14. Лап: 8 + 56 = 64
[#16] Гусей - 2, Кроликов - 15. Лап: 4 + 60 = 64
[#17] Гусей - 0, Кроликов - 16. Лап: 0 + 64 = 64
Process finished with exit code 0

```

Рисунок 12. Демонстрация работы программы индивидуального задания 3

UML-диаграмма:

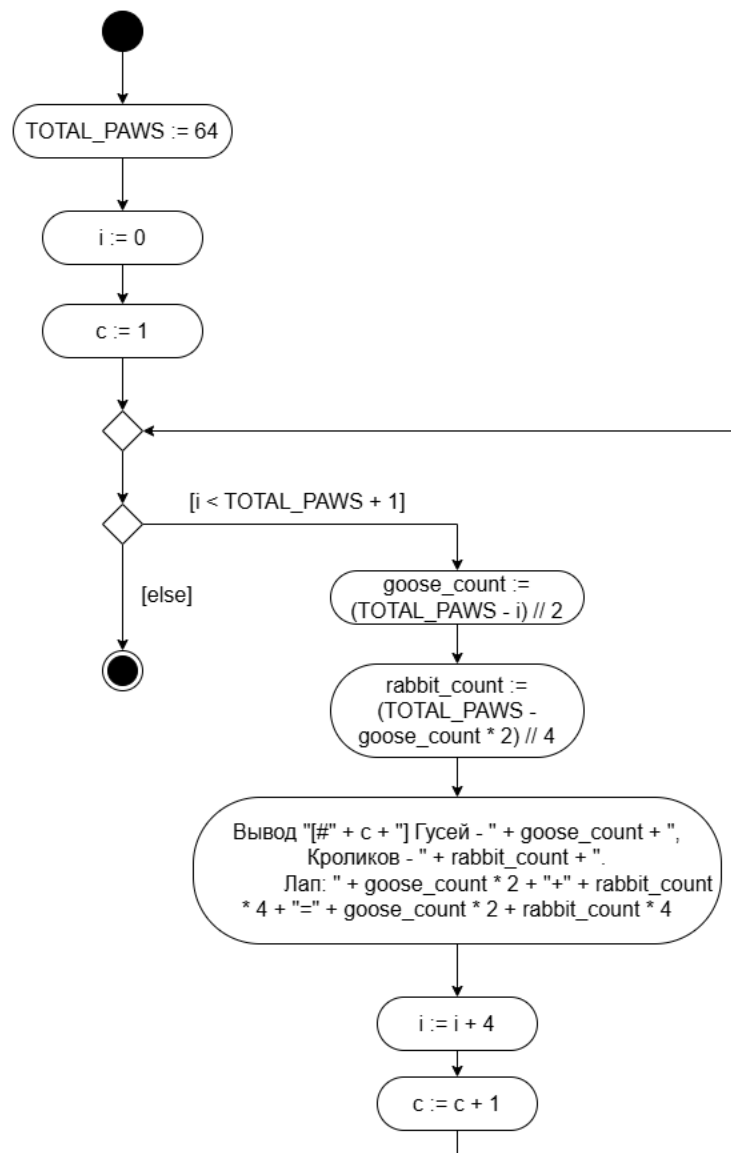


Рисунок 13. UML-диаграмма для индивидуального задания 3

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:

```
Командная строка
C:\Users\polko\Desktop\Учёба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3>git merge feature/example4
Already up to date.

C:\Users\polko\Desktop\Учёба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3>git merge feature/example5
Auto-merging .idea/workspace.xml
CONFLICT (add/add): Merge conflict in .idea/workspace.xml
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\polko\Desktop\Учёба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3>git merge feature/individual1
Merge made by the 'ort' strategy.
 individual_task_1.py | 39 +++++
 1 file changed, 39 insertions(+)
 create mode 100644 individual_task_1.py

C:\Users\polko\Desktop\Учёба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3>git merge feature/individual2
Merge made by the 'ort' strategy.
 individual_task_2.py | 18 +++++
 1 file changed, 18 insertions(+)
 create mode 100644 individual_task_2.py

C:\Users\polko\Desktop\Учёба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3>git merge feature/individual3
Merge made by the 'ort' strategy.
 individual_task_3.py | 12 +++++
 1 file changed, 12 insertions(+)
 create mode 100644 individual_task_3.py

C:\Users\polko\Desktop\Учёба\СКОУ\2 курс\1 семестр\Python\Лабораторная 3\PyProg_Practice_3>
```

Рисунок 14. Процесс сливания feature-веток в develop

Далее была подготовлена release-ветка: