**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное**

**образовательное учреждение высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №2.2**

Условные операторы и циклы в языке Python

|  |
| --- |
| Выполнил студент группы  ИВТ-б-о-21-1 |
| Богадуров В.И. « »\_\_\_\_\_\_\_\_\_20\_\_г. |
| Подпись студента\_\_\_\_\_\_\_\_\_\_\_ |
| Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Проверил доцент  Кафедры инфокоммуникаций, старший преподаватель  Воронкин Р.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Ставрополь 2022

**Тема:** Условные операторы и циклы в языке Python

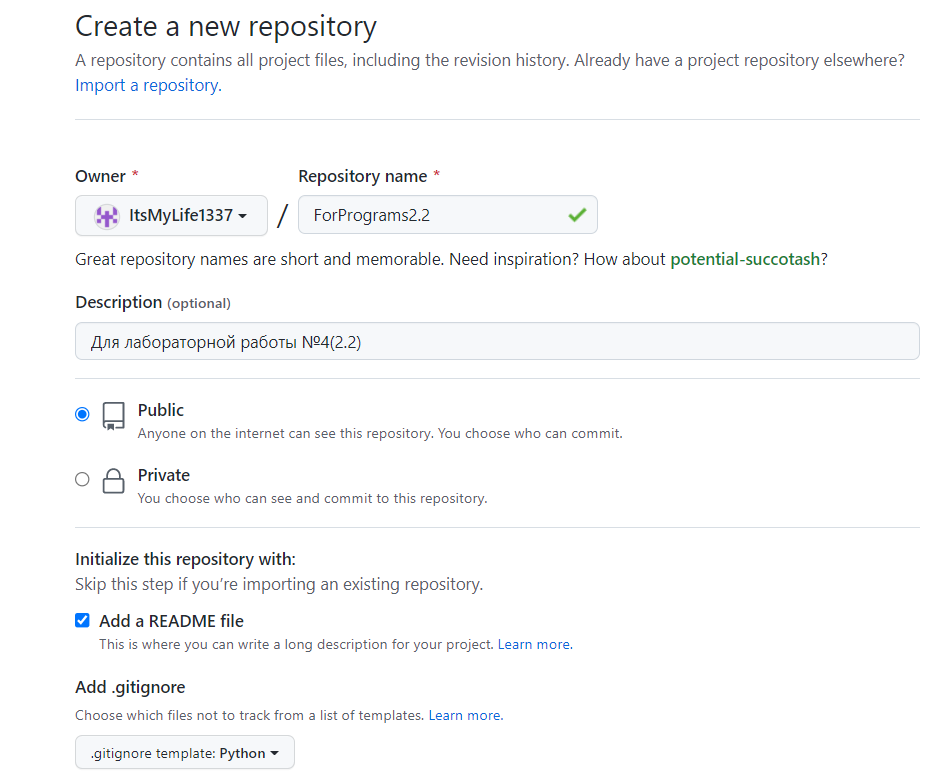
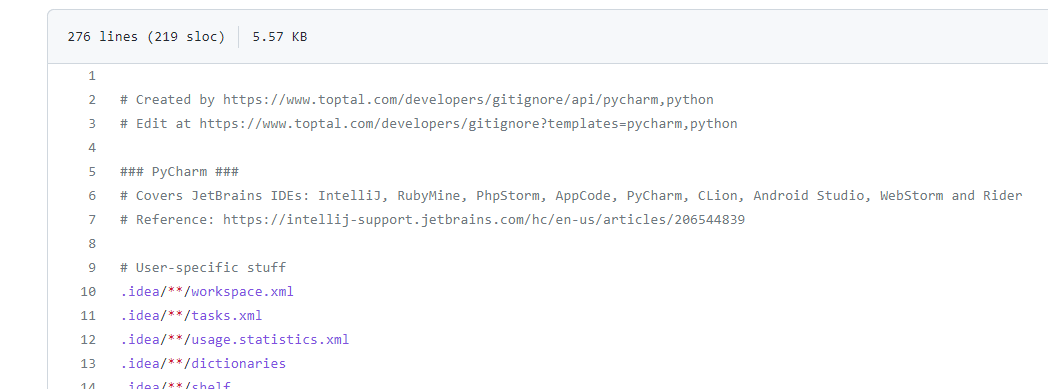
**№1.** Создать новый репозиторий и проработать примеры из лабораторной работы.

Рисунок 1 – Создание нового репозитория

Рисунок 2 – Изменённый файл .gitignore

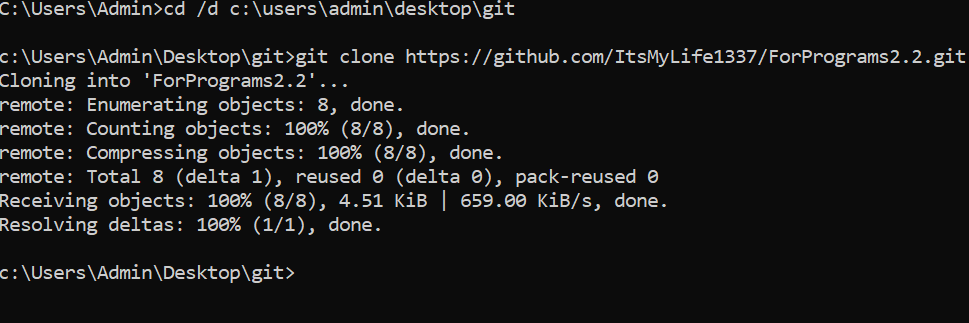
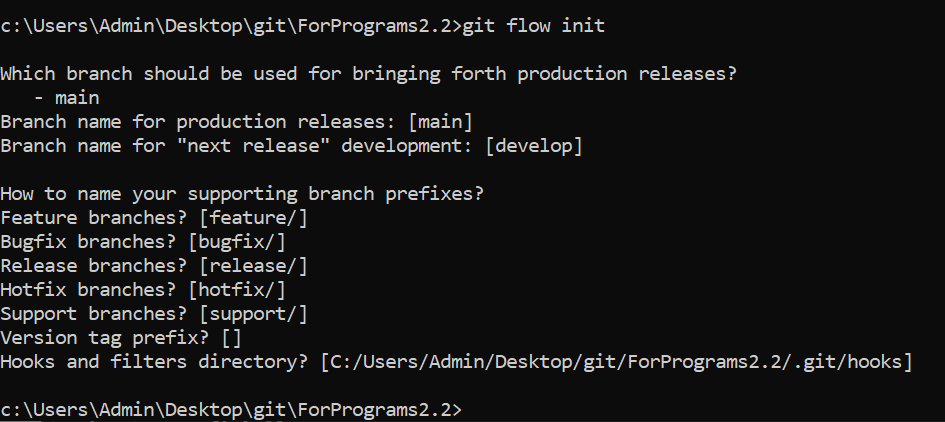
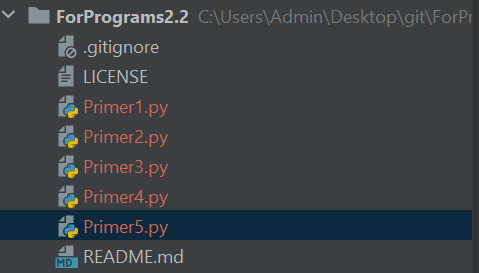
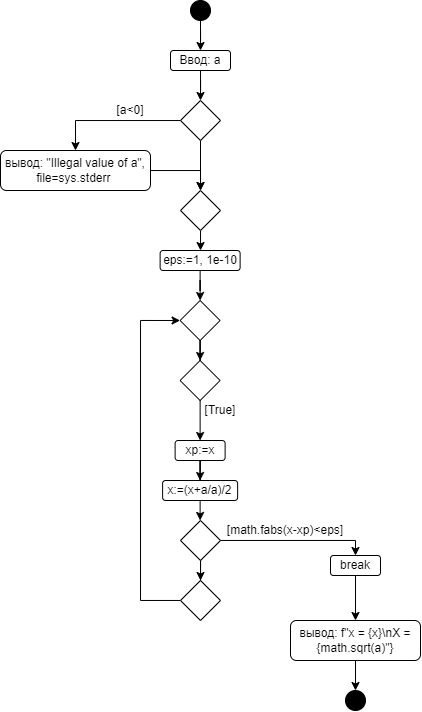
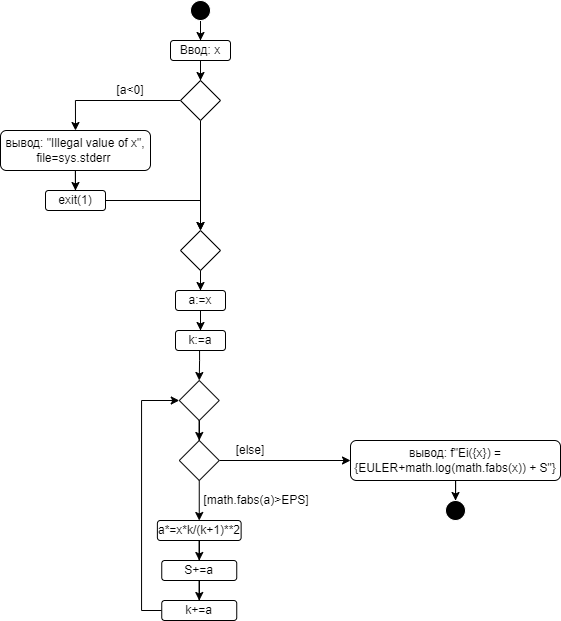
Рисунок 3 – Клонировал репозиторий себе на компьютер

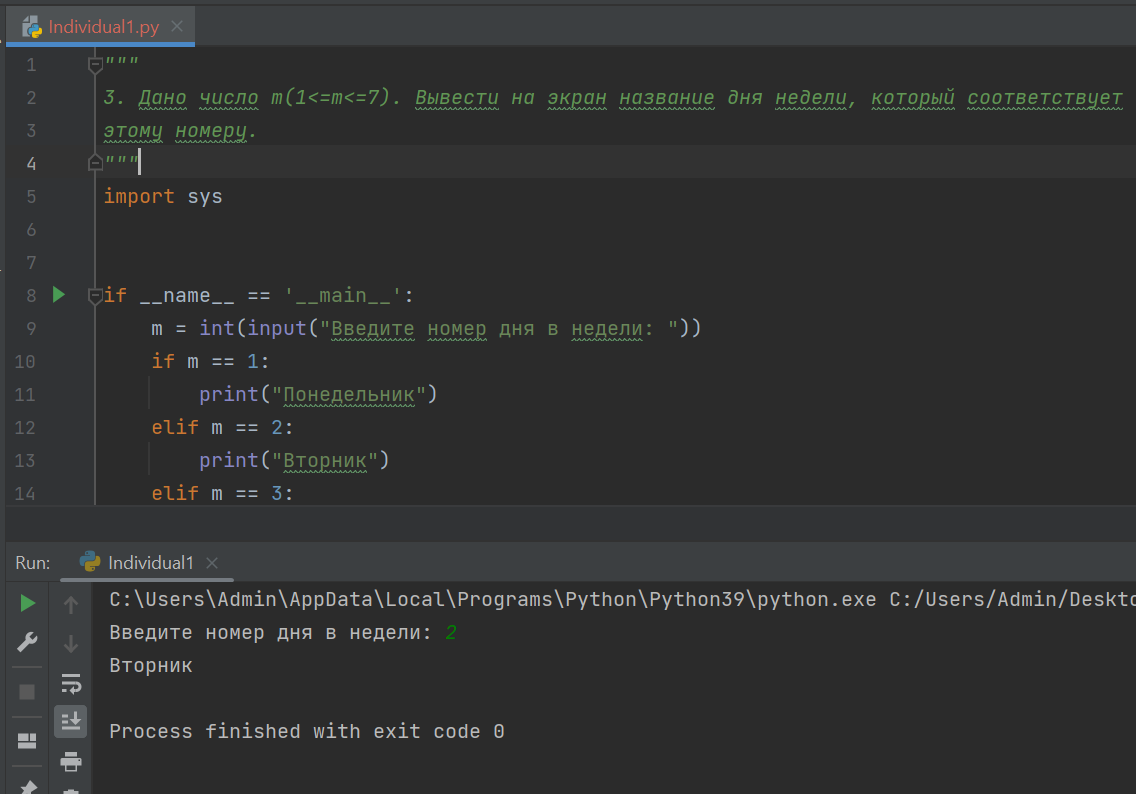
Рисунок 4 – Организация репозитория согласно модели git-flow

Рисунок 5 – Все проработанные примеры в лабораторной работе

Рисунок 6 – UML-диаграмма для примера №4

Рисунок 7 – UML-диаграмма для примера №5

**№2.** Выполнить индивидуальные задания и задание повышенной сложности согласно своему варианту(В-3). Построить UML-диаграммы написанных программ.

Рисунок 2.1 – Индивидуальное задание №1

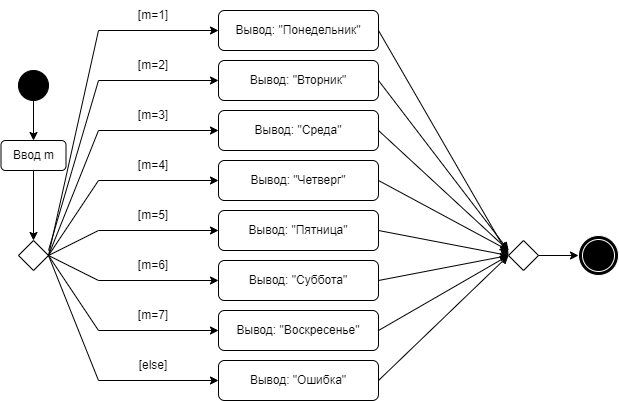


Рисунок 2.2 – UML-диаграмма для индивидуального задания №1

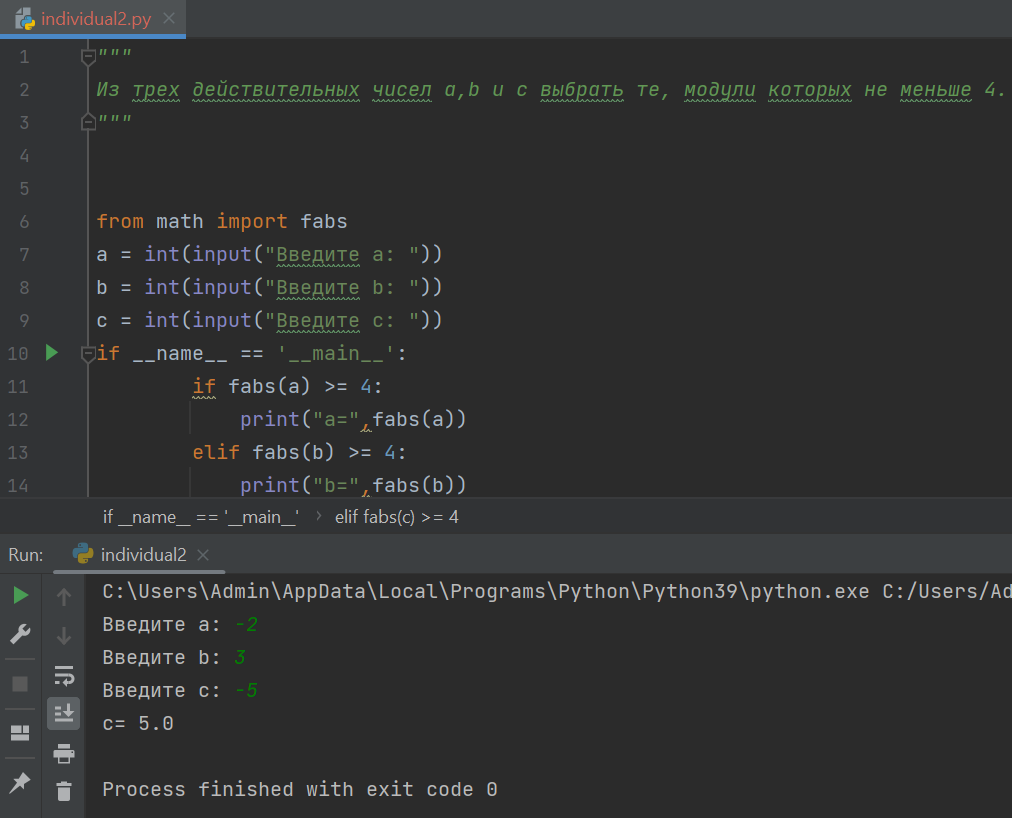


Рисунок 2.3 – Индивидуальное задание №2

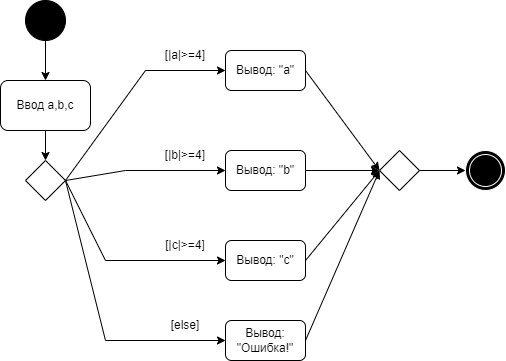


Рисунок 2.4 – UML-диаграмма для индивидуального задания №2

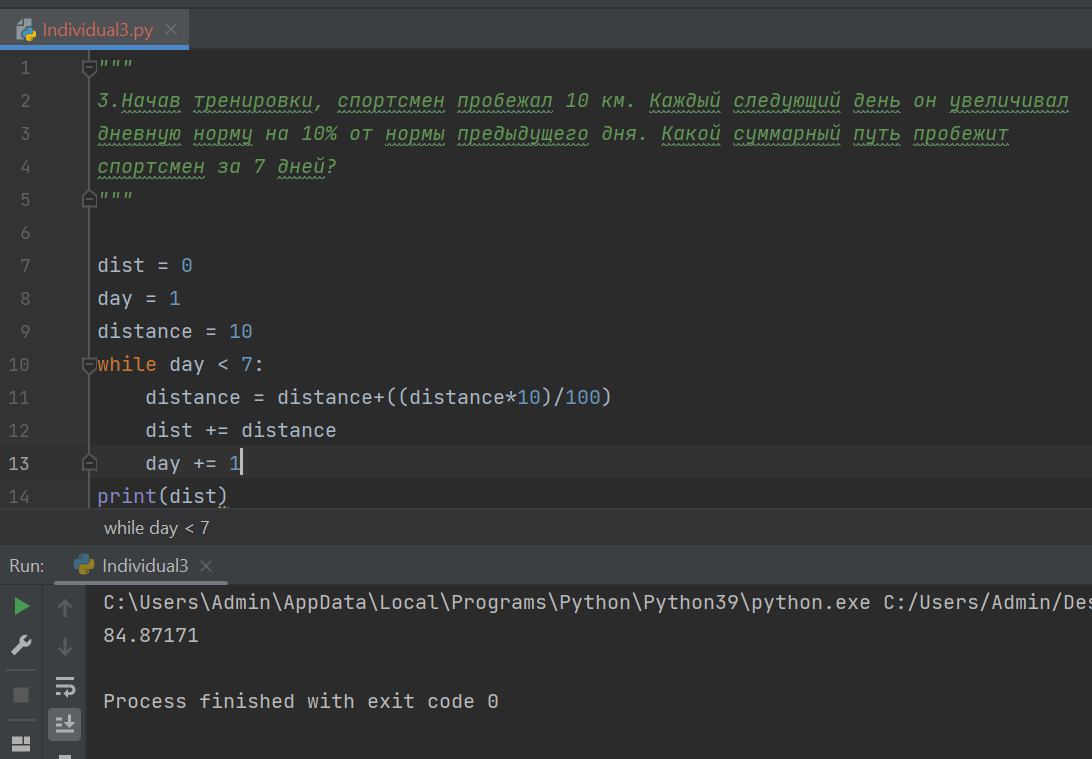
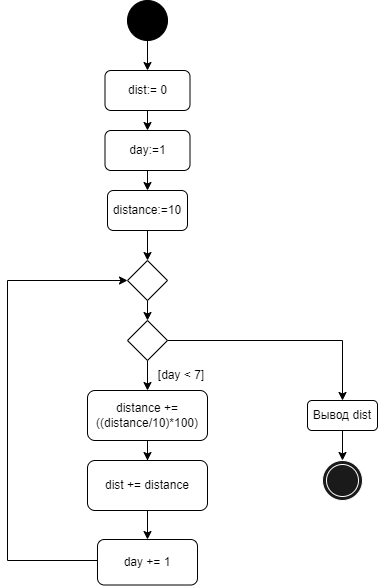
Рисунок 2.5 – Индивидуальное задание №3

Рисунок 2.6 – UML-диаграмма для индивидуального задания №3

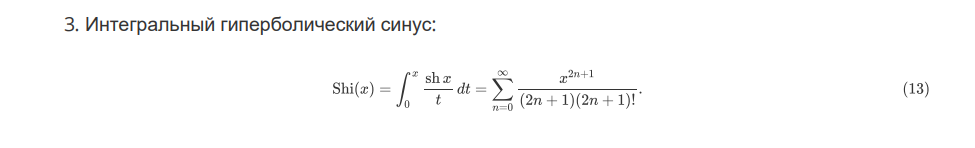
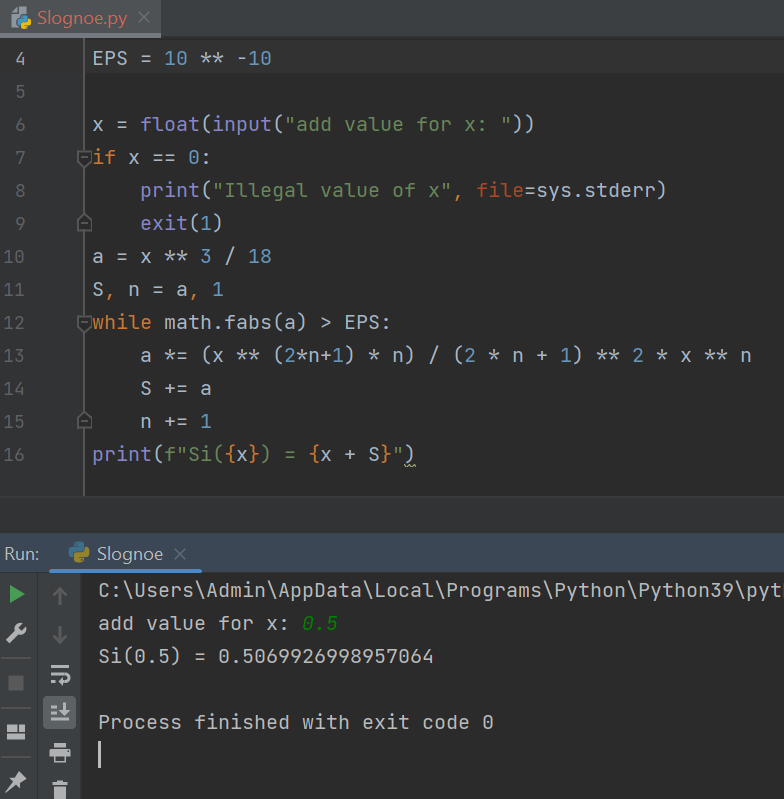
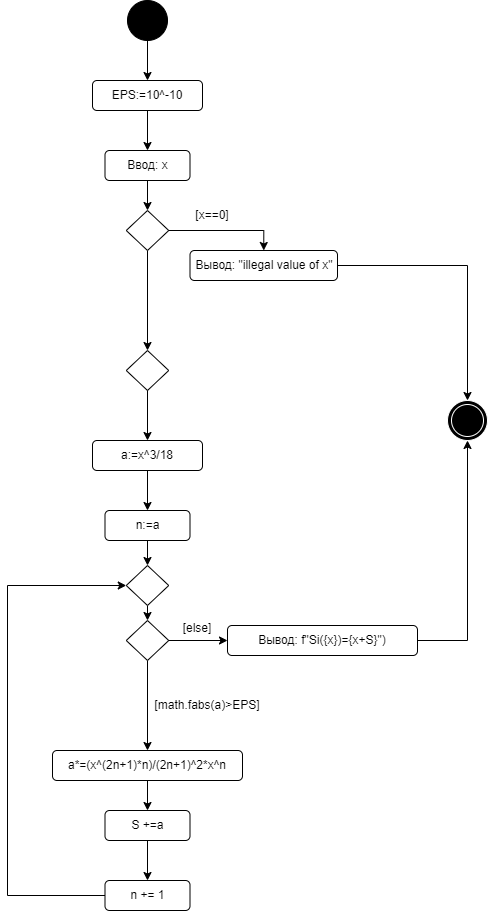
Рисунок 2.7 – Задание повышенной сложности В-3

Рисунок 2.8 – Задание повышенной сложности В-3

Рисунок 2.9 – UML-диаграмма для задачи повышенной сложности В-3



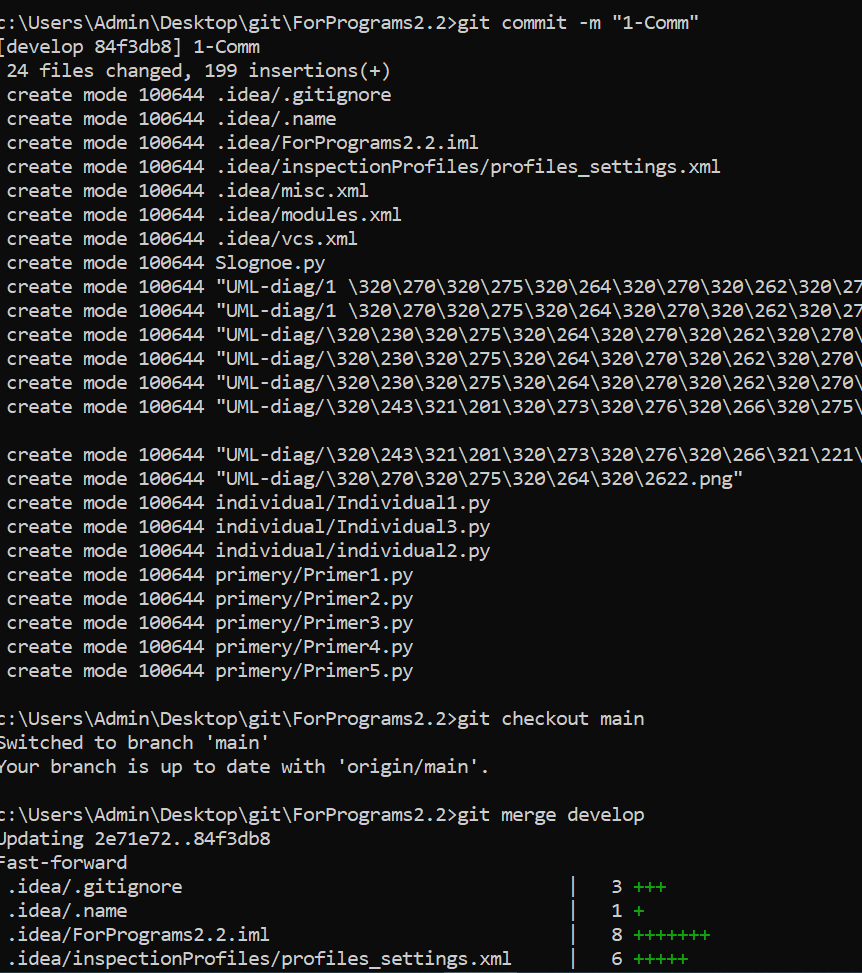
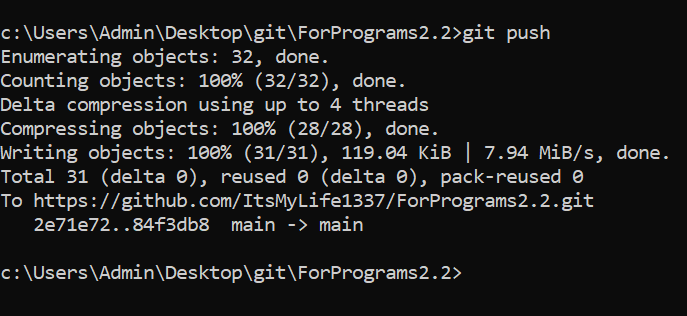
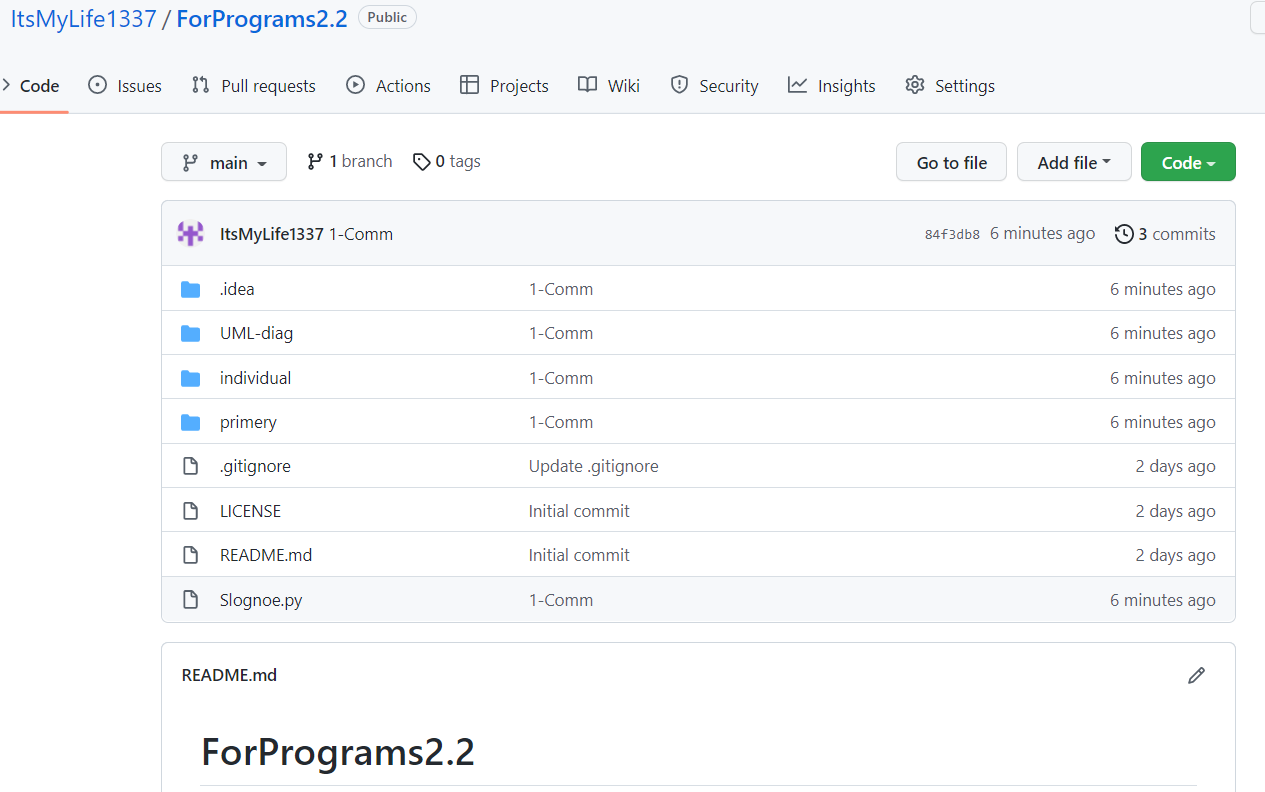
Рисунок 3.1 - Слияние ветки develop с веткой main

Рисунок 3.2 – Отправка изменений на удалённый репозиторий

Рисунок 3.3 – Зафиксировал изменения на удалённом репозитории

**Ответы на контрольные вопросы:**

**1. Для чего нужны диаграммы деятельности UML?**

Позволяет наглядно визуализировать алгоритм программы.

**2. Что такое состояние действия и состояние деятельности?**

Состояние действия - частный вид состояния деятельности, а конкретнее – такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции.

Состояние деятельности можно представить как составное состояние, поток управления которого включает только другие состояния деятельности и действий.

**3. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?**

Переходы, ветвление, алгоритм разветвляющейся структуры, алгоритм циклической структуры.

**4. Какой алгоритм является алгоритмом разветвляющейся структуры?**

Алгоритм разветвляющейся структуры - это алгоритм, в котором вычислительный процесс осуществляется либо по одной, либо по другой ветви, в зависимости от выполнения некоторого условия.

**5. Чем отличается разветвляющийся алгоритм от линейного?**

Линейный алгоритм - алгоритм, все этапы которого выполняются однократно и строго последовательно.

Разветвляющийся алгоритм - алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из нескольких возможных шагов.

**6. Что такое условный оператор? Какие существуют его формы?**

Оператор, конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд.

Условный оператор имеет полную и краткую формы.

**7. Какие операторы сравнения используются в Python?**

If, elif, else

**8. Что называется простым условием? Приведите примеры.**

Простым условием называется выражение, составленное из двух арифметических выражений или двух текстовых величин.

Пример: a == b

**9. Что такое составное условие? Приведите примеры.**

Составное условие – логическое выражение, содержащее несколько простых условий объединенных логическими операциями. Это операции not, and, or.

Пример: (a == b or a == c)

**10. Какие логические операторы допускаются при составлении сложных условий?**

not, and, or.

**11. Может ли оператор ветвления содержать внутри себя другие ветвления?**

Может.

**12. Какой алгоритм является алгоритмом циклической структуры?**

Циклический алгоритм — это вид алгоритма, в процессе выполнения которого одно или несколько действий нужно повторить.

**13. Типы циклов в языке Python.**

В Python есть 2 типа циклов: - цикл while, - цикл for.

**14. Назовите назначение и способы применения функции range.**

Функция range генерирует серию целых чисел, от значения start до stop,

указанного пользователем. Мы можем использовать его для цикла for и обходить весь диапазон как список.

**15. Как с помощью функции range организовать перебор значений от 15 до 0 с шагом 2?**

Range (15, 0, 2)

**16. Могул ли быть циклы вложенными?**

Могут.

**17. Как образуется бесконечный цикл и как выйти из него?**

Бесконечный цикл в программировании — цикл, написанный таким образом, что условие выхода из него никогда не выполняется.

**18. Для чего нужен оператор break?**

Используется для выхода из цикла.

**19. Где употребляется оператор continue и для чего он используется?**

Оператор continue используется только в циклах. В операторах for , while , do while , оператор continue выполняет пропуск оставшейся части кода тела цикла и переходит к следующей итерации цикла.

**20. Для чего нужны стандартные потоки stdout и stderr?**

Ввод и вывод распределяется между тремя стандартными потоками:

stdin — стандартный ввод (клавиатура), stdout — стандартный вывод (экран),

stderr — стандартная ошибка (вывод ошибок на экран)

**21. Как в Python организовать вывод в стандартный поток stderr?**

Указать в print (…, file=sys.stderr).

**22. Каково назначение функции exit?**

Функция exit() модуля sys - выход из Python.

**Вывод:** в результате выполнения работы были приобретены навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Также, освоены операторы языка Python версии 3.x if , while , for , break и continue , позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.