1инистерство 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 ПРАКТИЧЕСКОЙ РАБОТЕ №1**

**дисциплины «Программирование на Python»**

Вариант №13

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнила:  Ковжого Елизавета Андреевна  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 u continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Порядок выполнения работы:

Адес репозитория: https://github.com/LissKovzogo/Python\_LAB\_3.git

1. Создали, настроили и клонировали репозиторий Python\_LAB\_3.
2. Создали проект PyCharm в папке репозитория.
3. Проработали все примеры лабораторной работы и создали для каждого отдельный модуль Python.

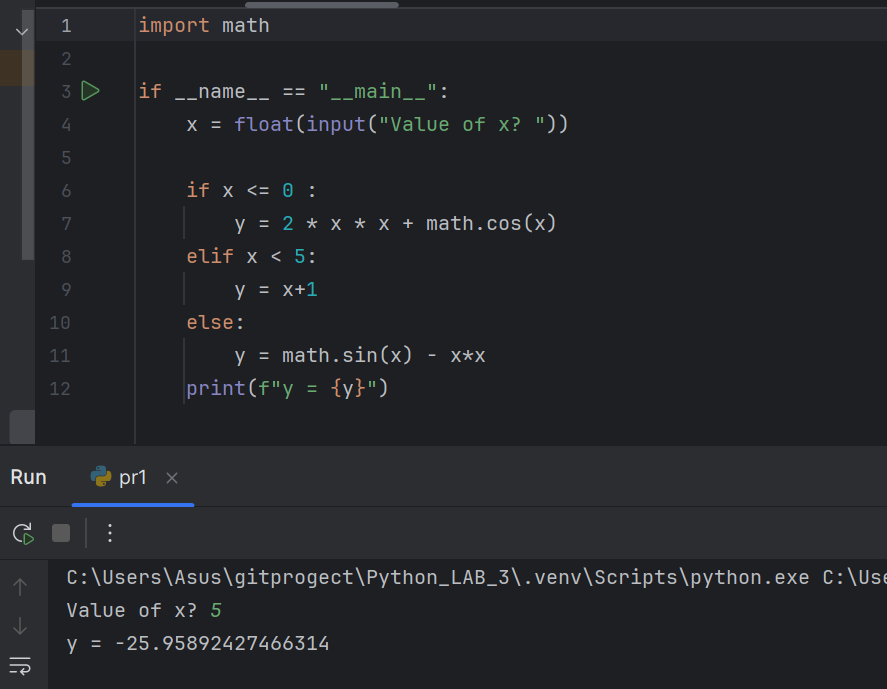


Рис. 1 — Работа с примером № 1

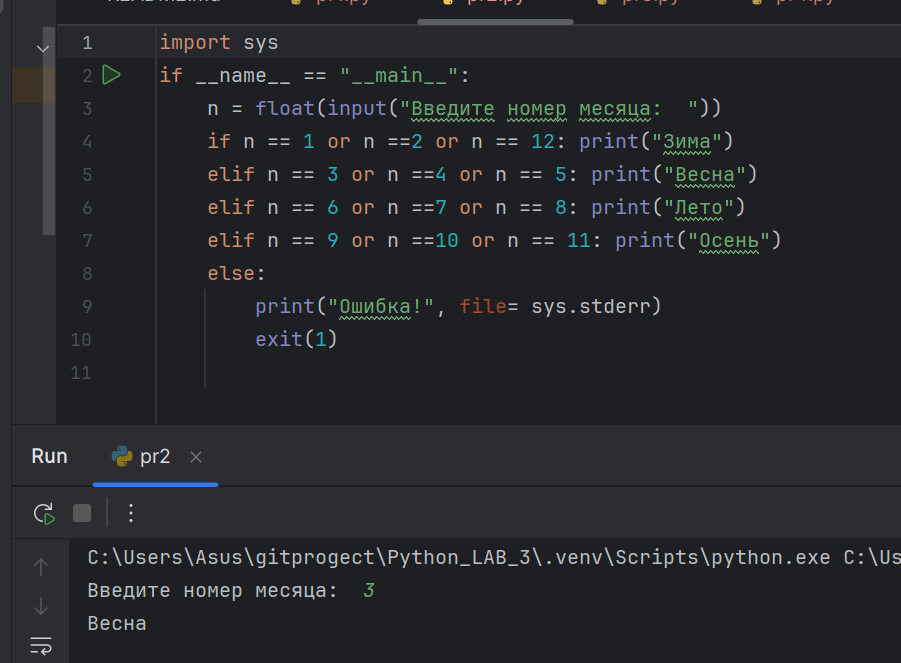


Рис. 2 — Работа с примером № 2

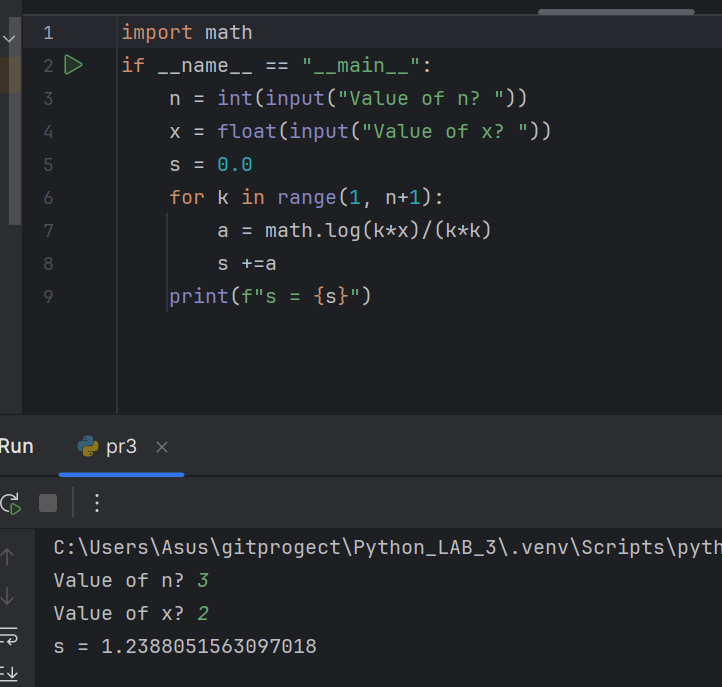


Рис. 3 — Работа с примером № 3

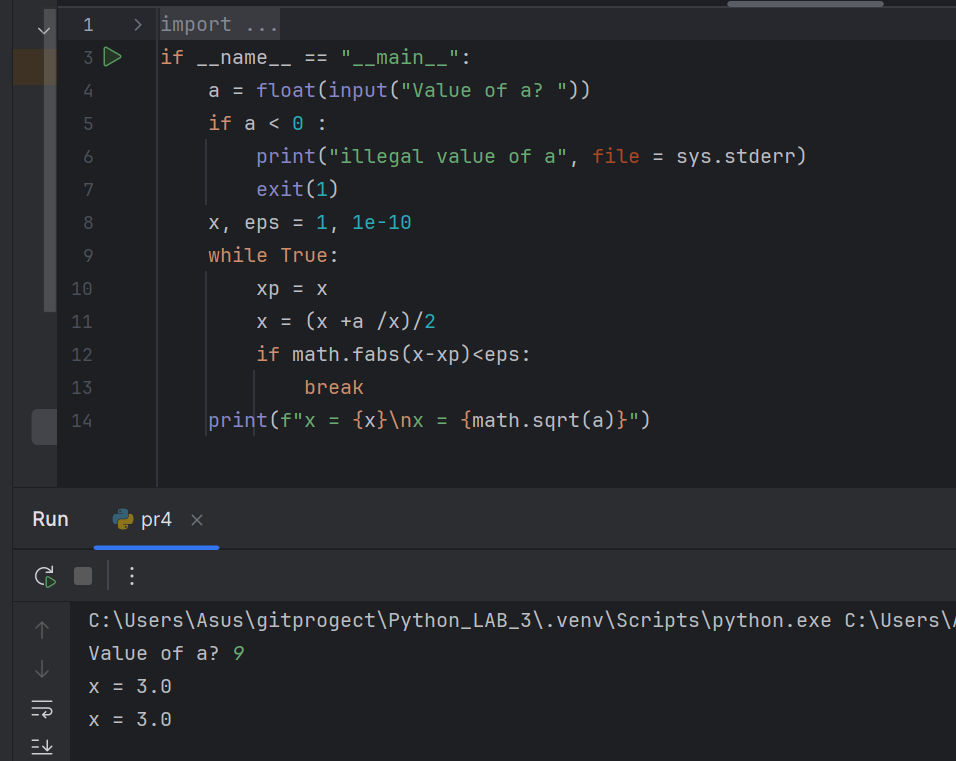


Рис. 4 — Работа с примером № 4

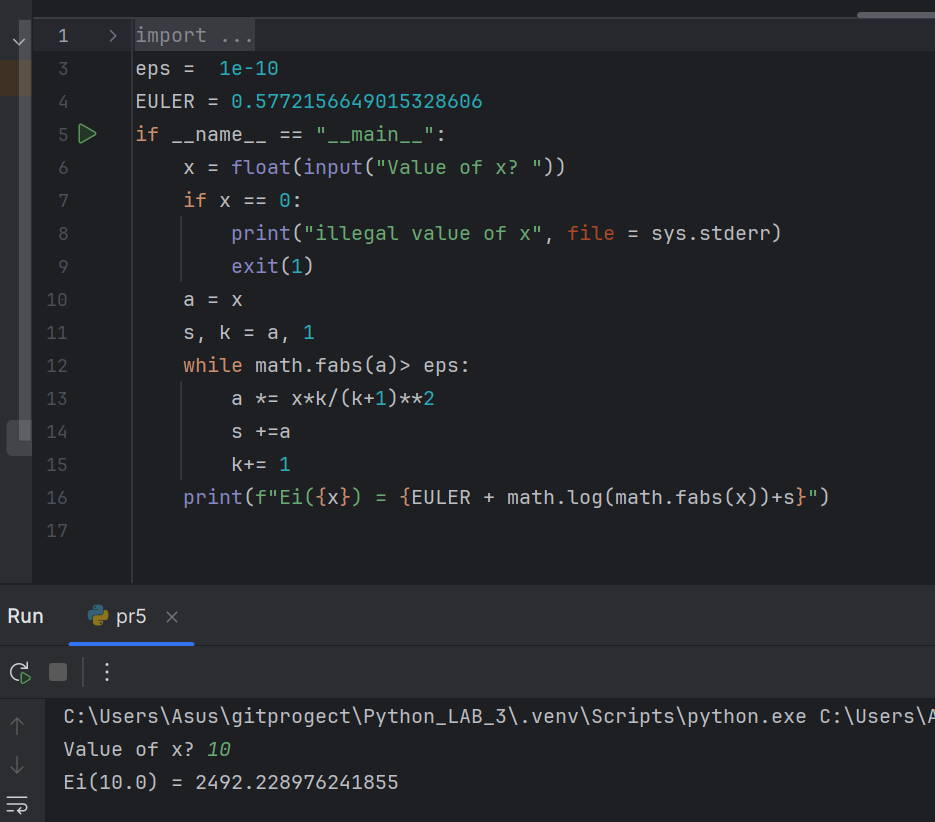


Рис. 5 — Работа с примером № 5

1. Построили UML-диаграммы деятельности для примеров №4 и №5.

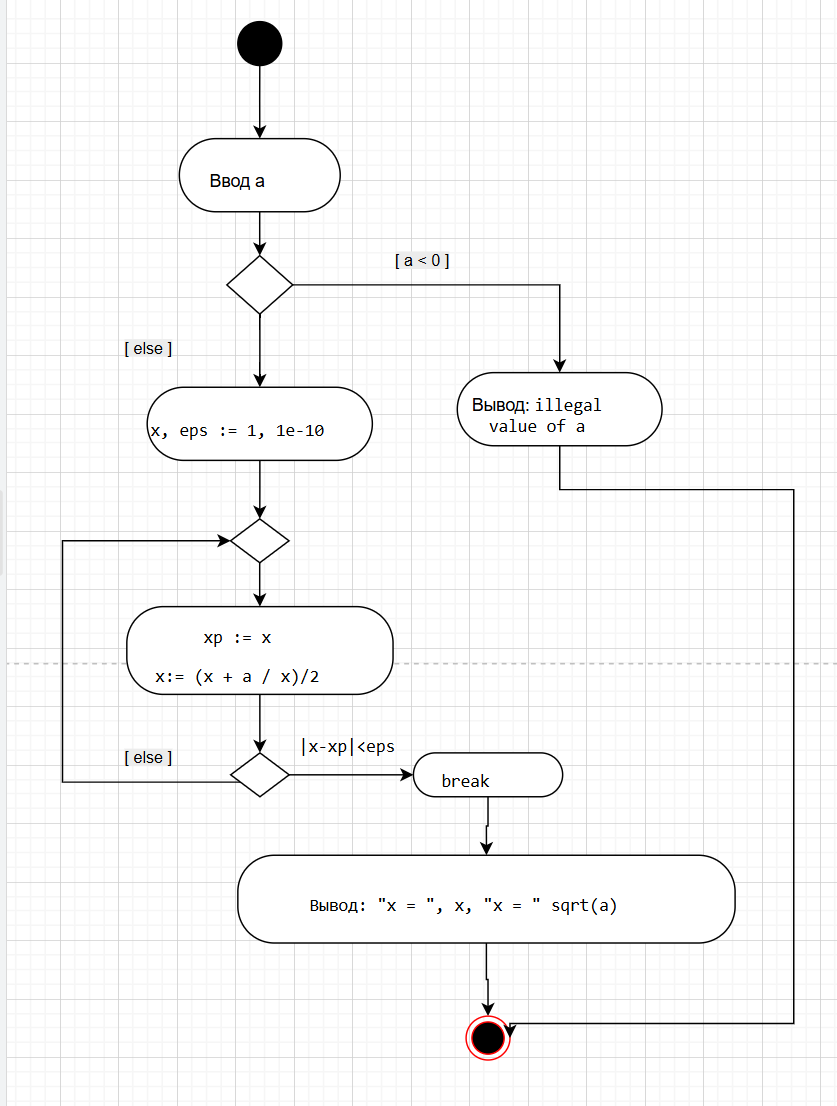


Рис. 6 — UML-диаграммы деятельности для примеров №4

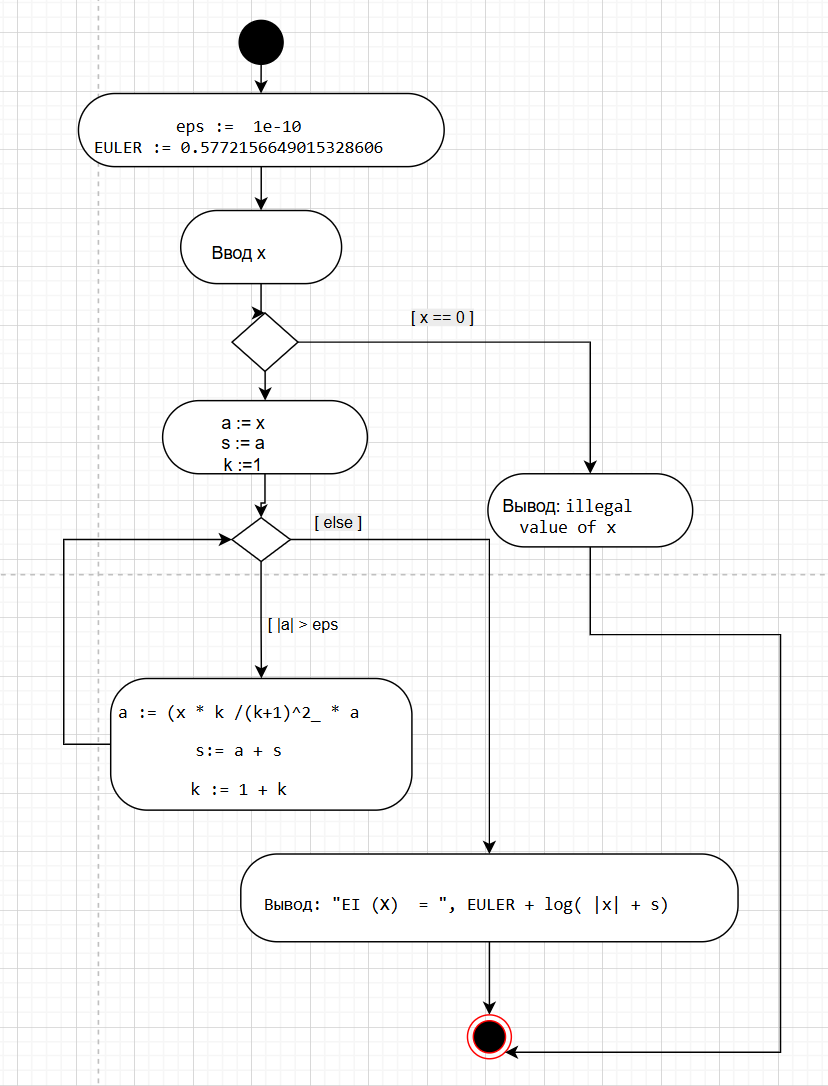


Рис. 7 — UML-диаграммы деятельности для примеров №5

1. Выполнили индивидуальные задания согласно варианту и построили UML-диаграммы деятельности.

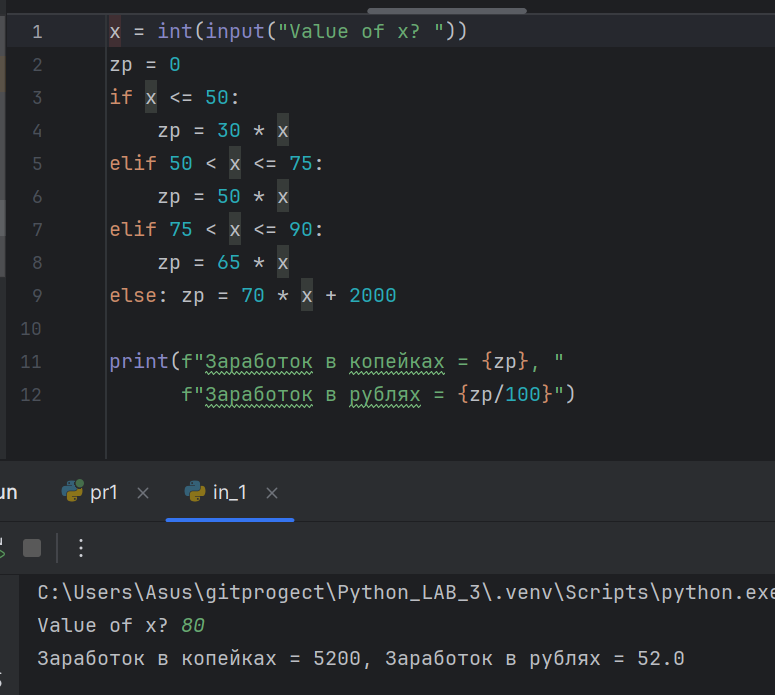


Рис. 8 — Индивидуальное задание №1

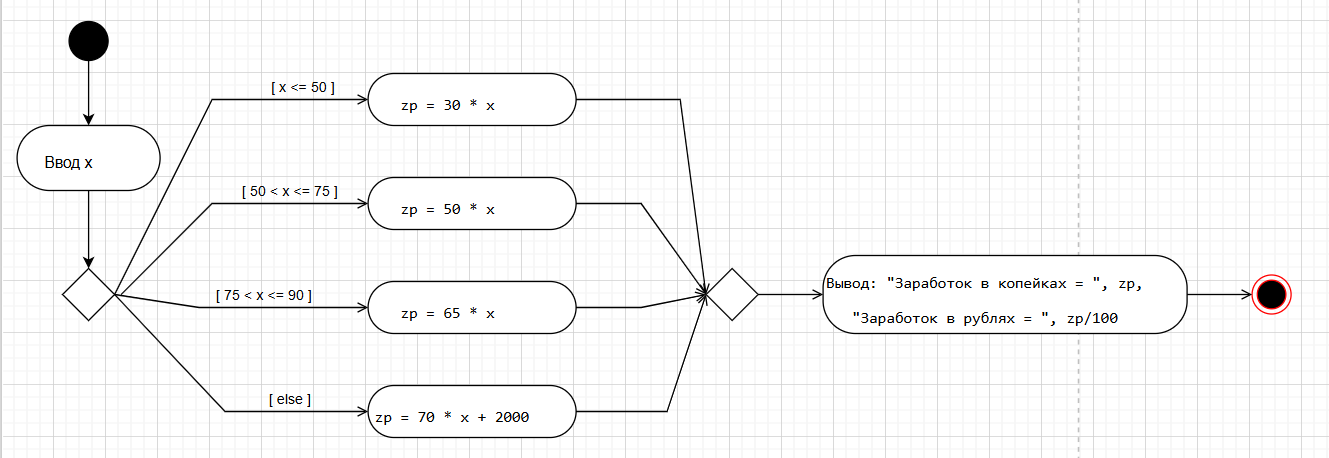


Рис. 9 — UML-диаграммы деятельности для индивидуальное задания №1

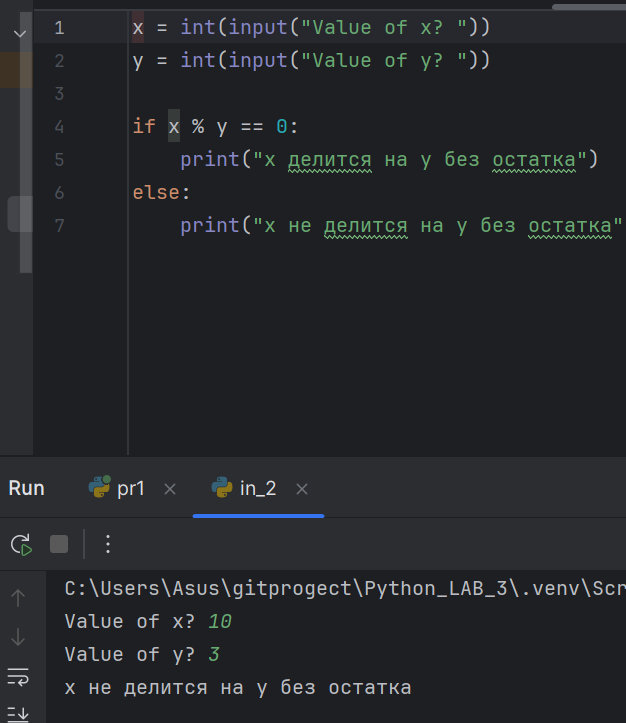


Рис. 10 — Индивидуальное задание №2

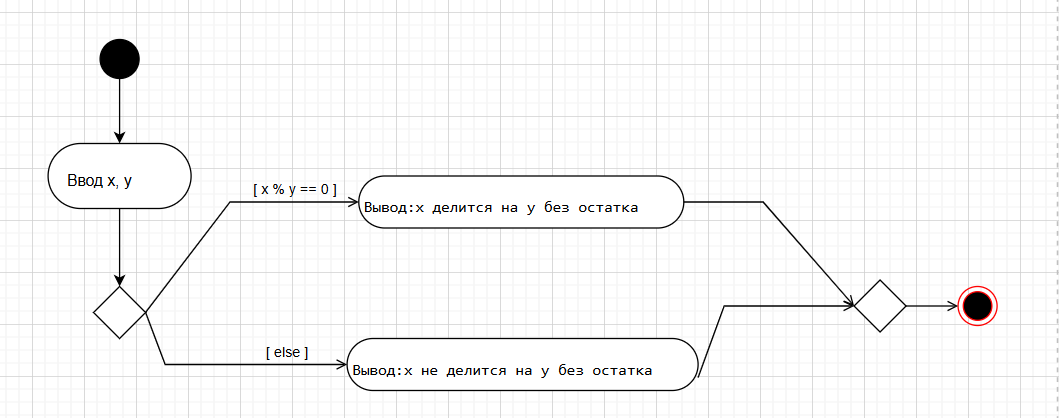


Рис. 11 — UML-диаграммы деятельности для индивидуальное задания №2

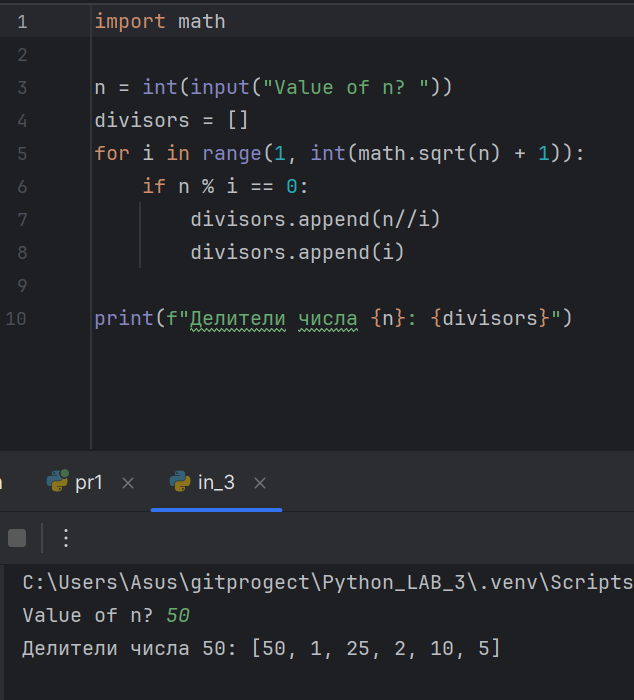


Рис. 12 — Индивидуальное задание №3

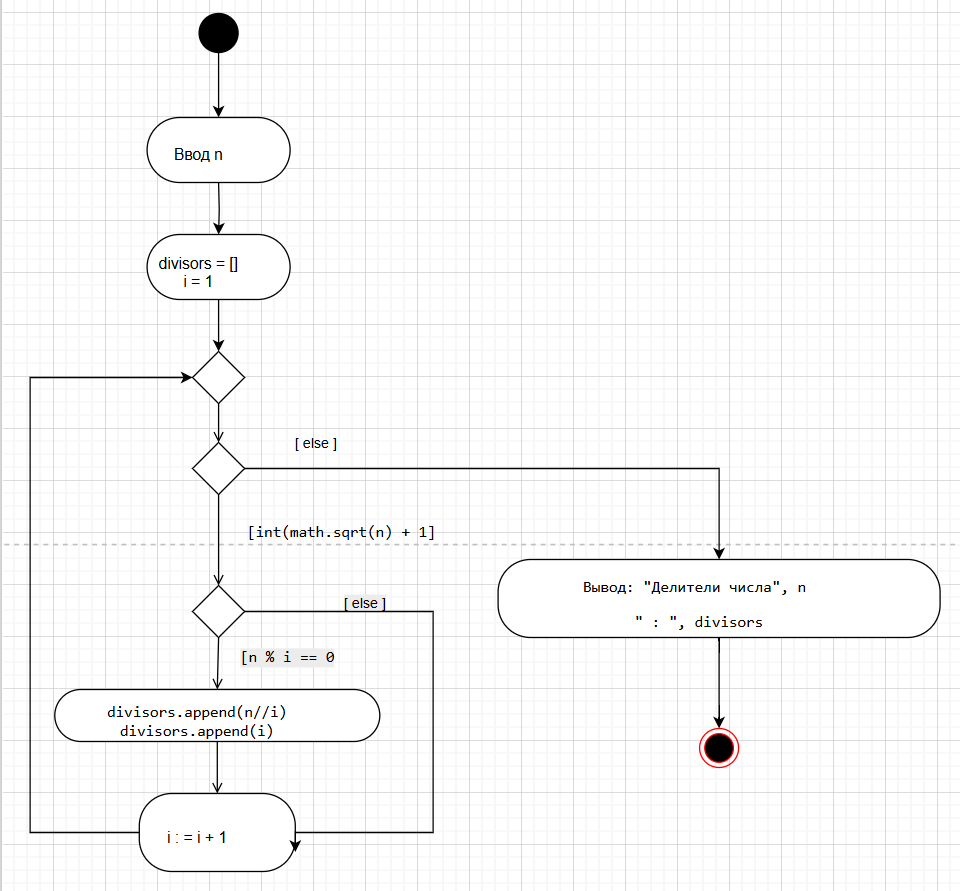


Рис. 13 — UML-диаграммы деятельности для индивидуальное задания №3

1. Зафиксировали все изменения в репозиторий.

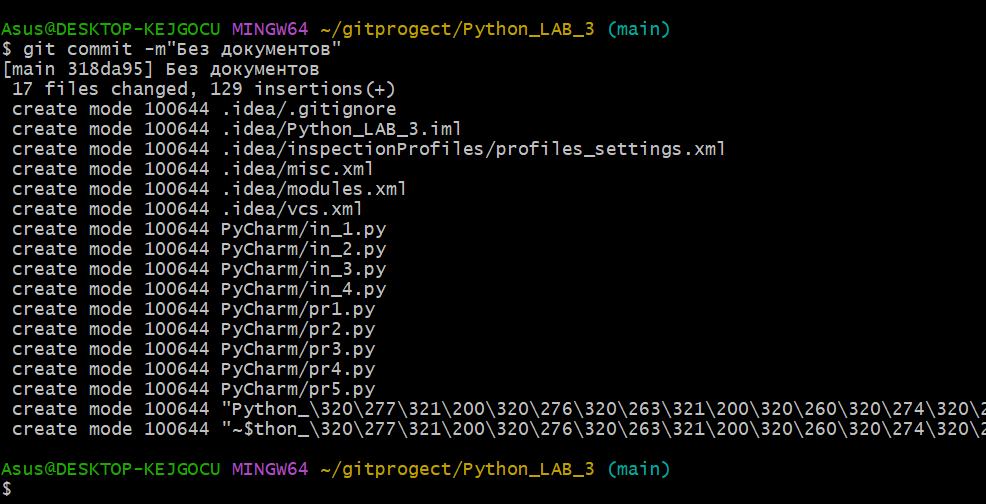


Рис. 14 — Фиксирование изменений

1. Выполнили слияние веток.



Рис. 15 — Слияние веток

Контрольные вопросы:

1. Для чего нужны диаграммы деятельности UML?

Для моделирования бизнес-процессов, рабочих потоков и алгоритмов программ.

2. Что такое состояние действия и состояние деятельности?

Состояние действия - атомарная операция. Состояние деятельности - составная операция, содержащая поддеятельности.

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

Ромб для ветвлений, стрелки для переходов.

4. Какой алгоритм является алгоритмом разветвляющейся структуры?

Алгоритм с условиями и выбором разных путей выполнения.

5. Чем отличается разветвляющийся алгоритм от линейного?  
Наличием условий и альтернативных ветвей выполнения.

6. Что такое условный оператор? Какие существуют его формы?  
Оператор ветвления. Формы: if, if-else, if-elif-else.

7. Какие операторы сравнения используются в Python?  
==, !=, <, >, <=, >=.

8. Что называется простым условием? Приведите примеры.  
Условие с одним сравнением: x > 5, name == "John".

9. Что такое составное условие? Приведите примеры.  
Условие с логическими операторами: x > 5 and y < 10, age >= 18 or parent\_consent.

10. Какие логические операторы допускаются при составлении сложных условий?

Логические операторы: and, or, not.

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

Да, это называется вложенные условия.

12. Какой алгоритм является алгоритмом циклической структуры?

Алгоритм с повторяющимися операциями.

13. Типы циклов в языке Python.

for, while, do while.

14. Назовите назначение и способы применения функции range.

Генерация последовательностей чисел: range(start, stop, step).

15. Как с помощью функции range организовать перебор значений от 15 до 0 с шагом 2? for i in range(15, -1, -2):

16. Могут ли быть циклы вложенными?

Да, цикл внутри другого цикла.

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

while True:….

Выйти через break.

18. Для чего нужен оператор break?

Для досрочного выхода из цикла.

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

В циклах, для перехода к следующей итерации.

20. Для чего нужны стандартные потоки stdout и stderr?

stdout - стандартный вывод, stderr - вывод ошибок.

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

import sys; sys.stderr.write("Ошибка")

22. Каково назначение функции exit?

Завершение программы с кодом возврата.

Вывод: приобрели навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоили операторы языка Python версии 3.x if, while, for, break u continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.