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 ЛАБОРАТОРНОЙ РАБОТЕ №5**

**дисциплины «****Программирование на python»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Кожуховский Виктор Андреевич  2 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем  », очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Роман Александрович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

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

**Цель работы:** 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.

# Методика и порядок выполнения работы

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.

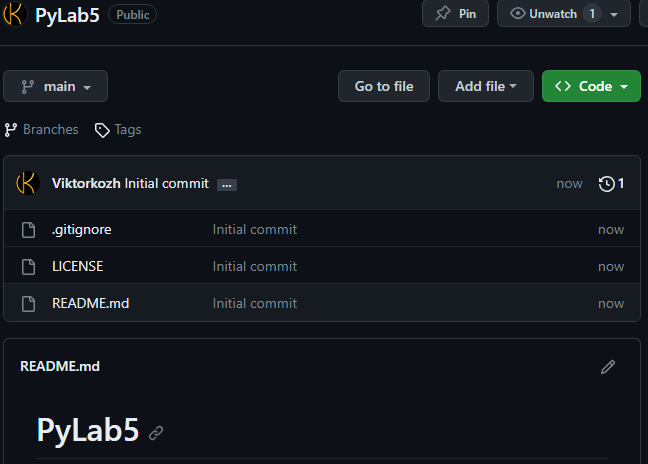


Рисунок 1. Созданный репозиторий

1. Выполнил клонирование созданного репозитория.

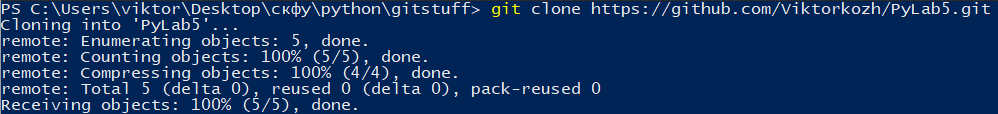


Рисунок 2. Клонированный репозиторий

1. Организовал свой репозиторий в соответствие с моделью ветвления git-flow.



Рисунок 3. Добавленная ветка developer

1. Проработал примеры лабораторной работы. Создал для каждого примера отдельный модуль языка Python. Зафиксировал изменения в репозитории.



Рисунок 4. Код и выполнение примера 1

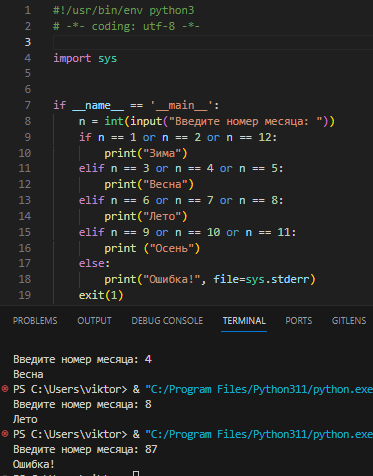


Рисунок 5. Код и выполнение примера 2

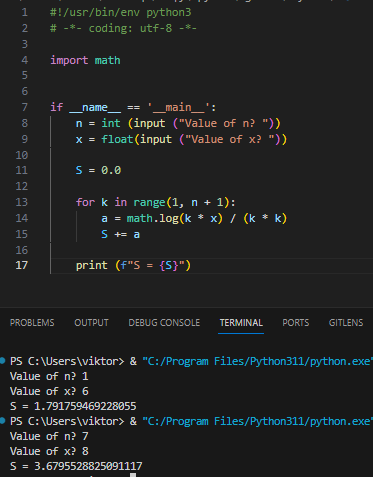


Рисунок 6. Код и выполнение примера 3

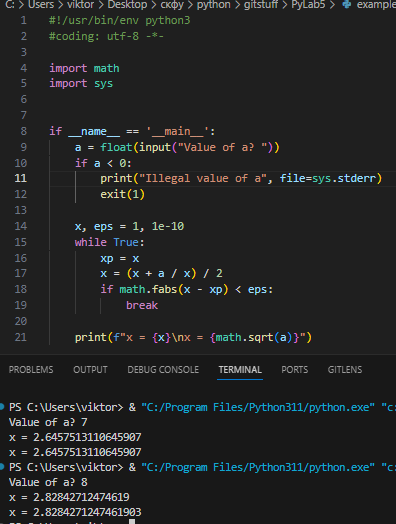


Рисунок 7. Код и выполнение примера 4

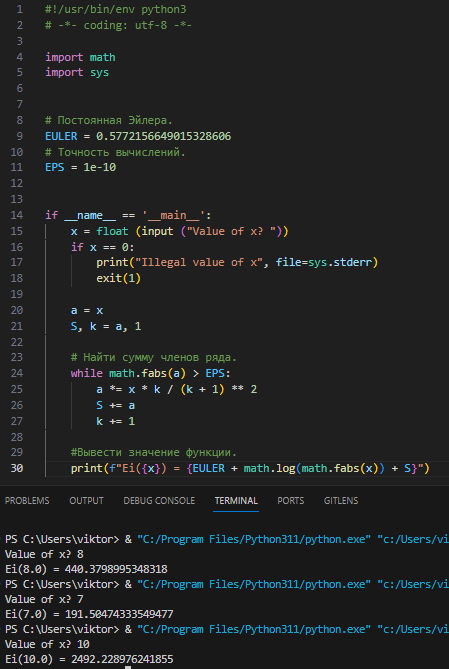


Рисунок 8. Код и выполнение примера 5

1. Для примеров 4 и 5 построил UML-диаграмму деятельности. Для построения диаграмм деятельности использовал веб-сервис Google https://www.diagrams.net/.

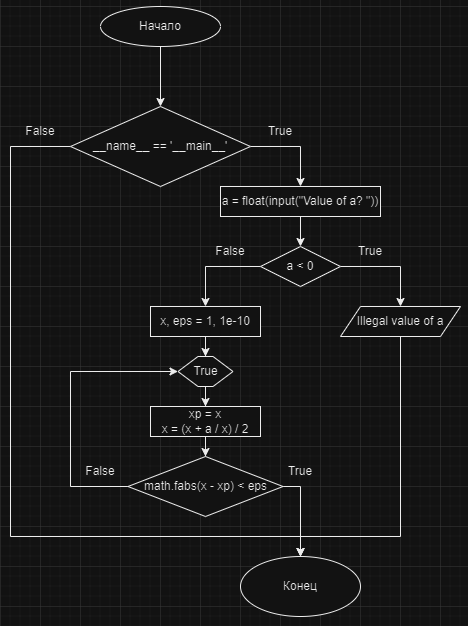


Рисунок 9. Схема примера 4

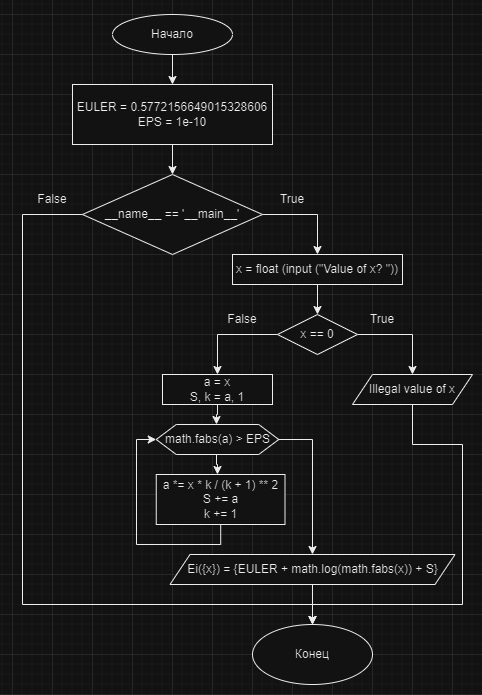


Рисунок 10. Схема примера 5

1. Выполнил индивидуальные задания, согласно своему варианту.

Дано число m (1<=m<=7). Вывести на экран название дня недели, который соответствует этому номеру.

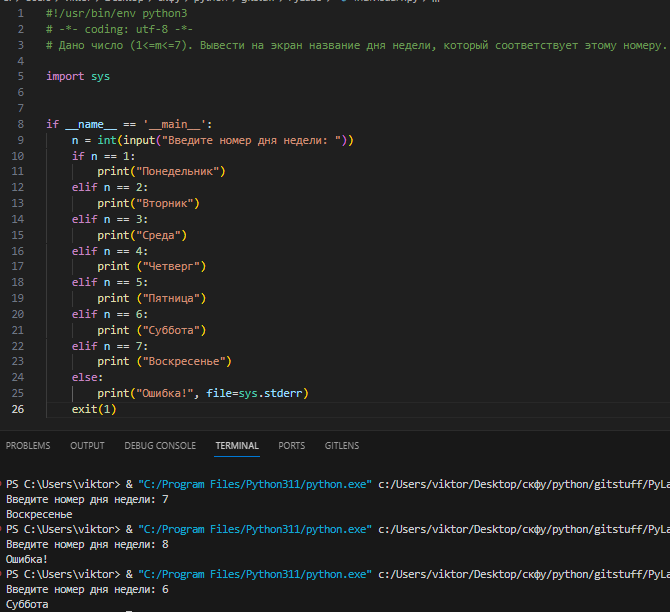


Рисунок 11. Код и выполнение индивидуального задания 1

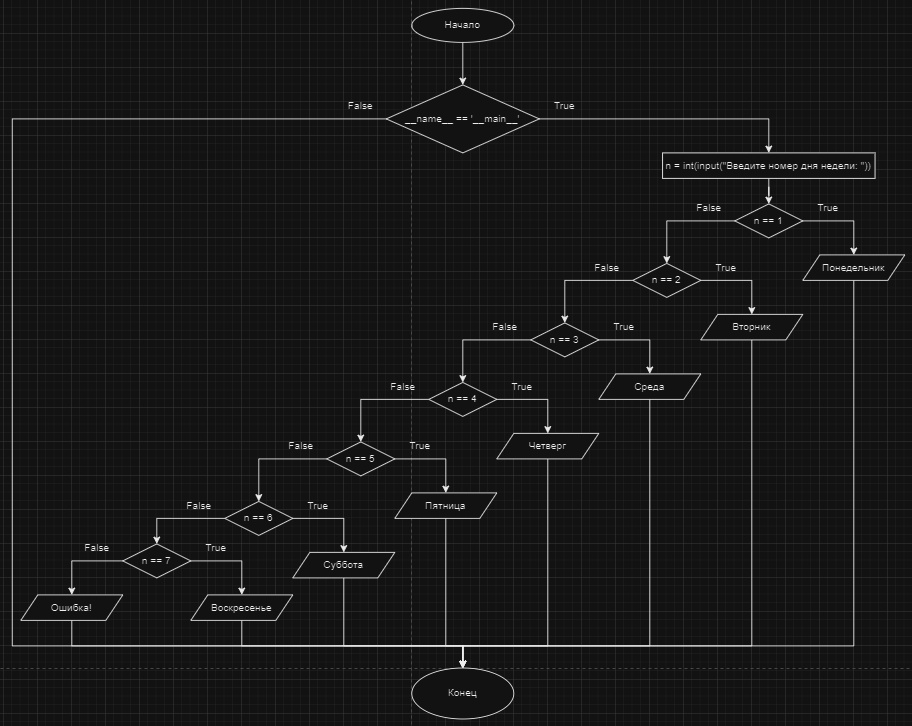


Рисунок 12. UML схема индивидуального задания 1

Даны три действительных числа. Составить программу с использованием конструкций ветвления, выбирающую из них те, которые принадлежат интервалу (0, 1).

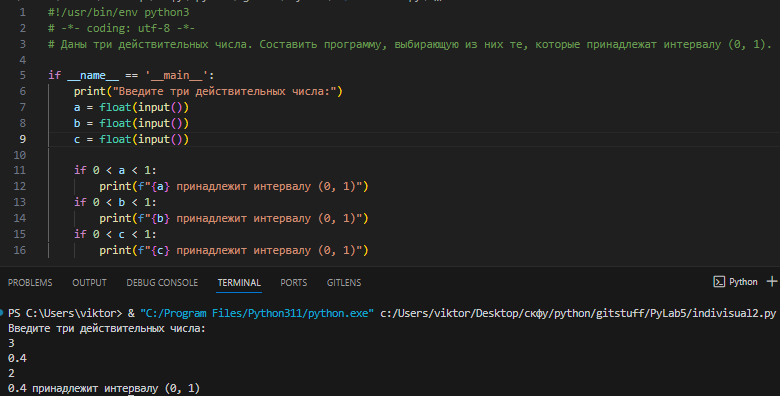


Рисунок 13. Код и выполнение индивидуального задания 2

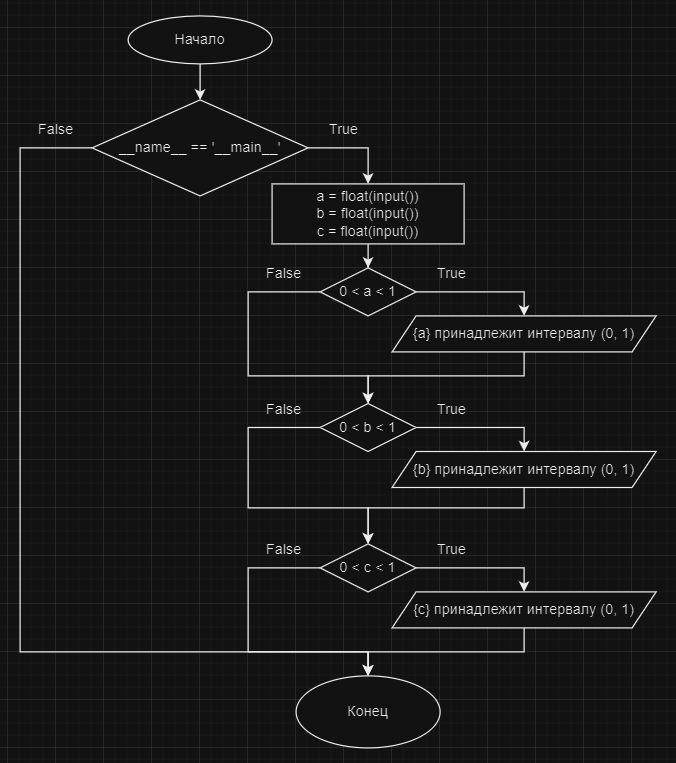


Рисунок 14. UML схема индивидуального задания 2

Ученик выучил в первый день 5 английских слов. В каждый следующий день он выучивал на 2 слова больше, чем в предыдущий. Сколько английских слов выучит ученик в 10-ый день занятий. Программа с использованием конструкций цикла.

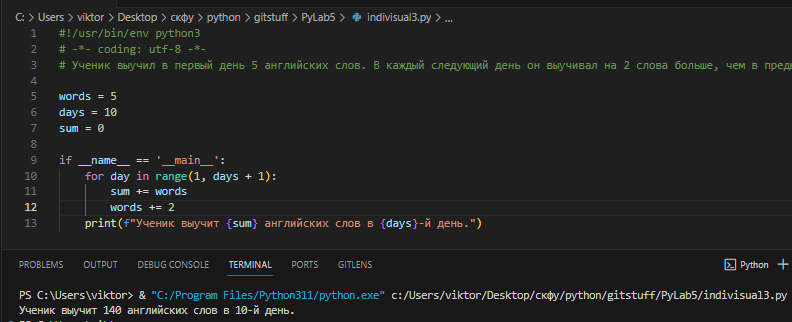


Рисунок 15. Код и выполнение индивидуального задания 3

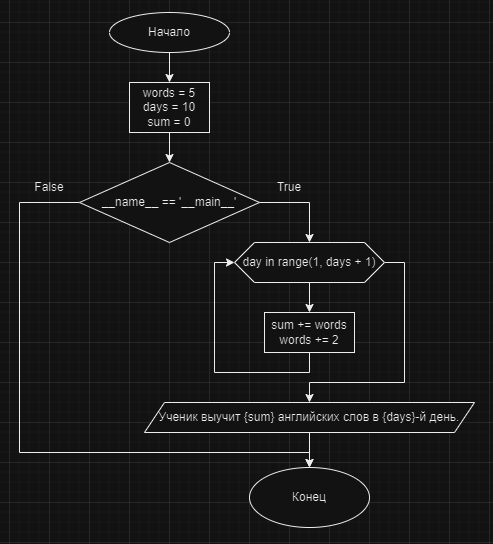


Рисунок 16. UML схема индивидуального задания 3

Функция Бесселя первого рода In(x), значение n = 0, 1, 2, ... - также должно вводиться с клавиатуры

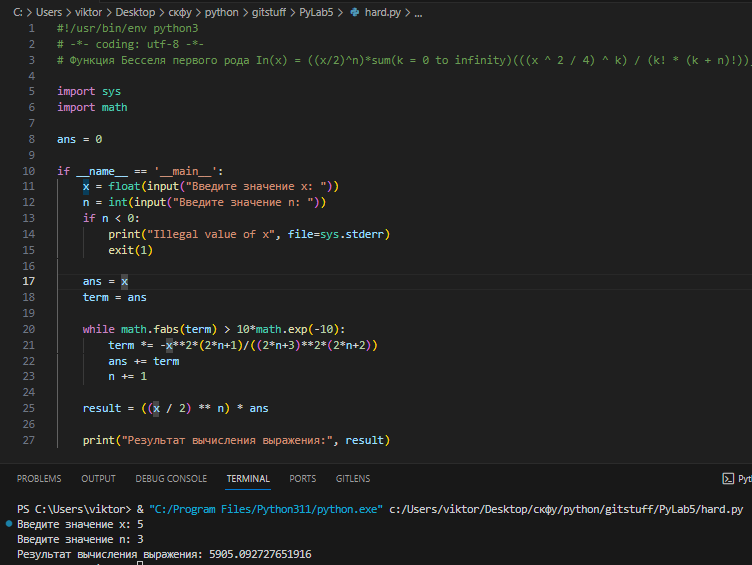


Рисунок 17. Код и выполнение усложненного задания

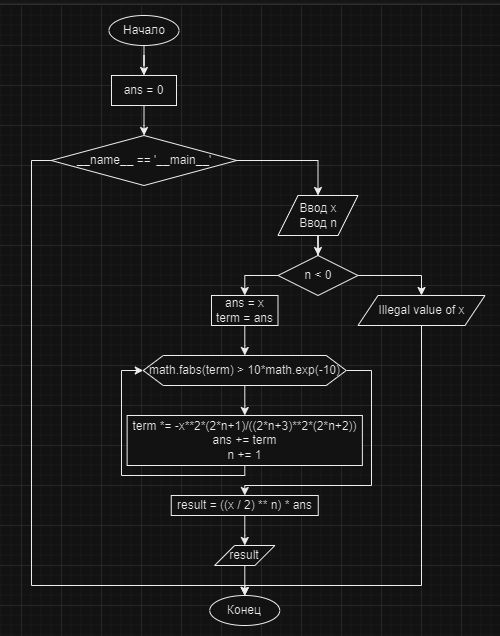


Рисунок 18. UML схема усложненного задания

1. Зафиксируйте сделанные изменения в репозитории.
2. Выполните слияние ветки для разработки с веткой main / master.
3. Отправьте сделанные изменения на сервер GitHub.

Вопросы для защиты работы

1. Для чего нужны диаграммы деятельности UML?

2. Что такое состояние действия и состояние деятельности?

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

4. Какой алгоритм является алгоритмом разветвляющейся структуры?

5. Чем отличается разветвляющийся алгоритм от линейного?

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

7. Какие операторы сравнения используются в Python?

8. Что называется простым условием? Приведите примеры.

9. Что такое составное условие? Приведите примеры.

10. Какие логические операторы допускаются при составлении сложных условий?

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

12. Какой алгоритм является алгоритмом циклической структуры?

13. Типы циклов в языке Python.

14. Назовите назначение и способы применения функции range.

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

16. Могул ли быть циклы вложенными?

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

18. Для чего нужен оператор break?

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

20. Для чего нужны стандартные потоки stdout и stderr?

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

22. Каково назначение функции exit?