МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Условные операторы и циклы в языке Python»

Отчет по лабораторной работе № 2.2

по дисциплине «Основы кроссплатформенного программирования»

Выполнил студент группы ИВТ-б-о-21-1

Сумин Никита Сергеевич.

«10 » мая 2022г.

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

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.

Порядок выполнения работы:

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

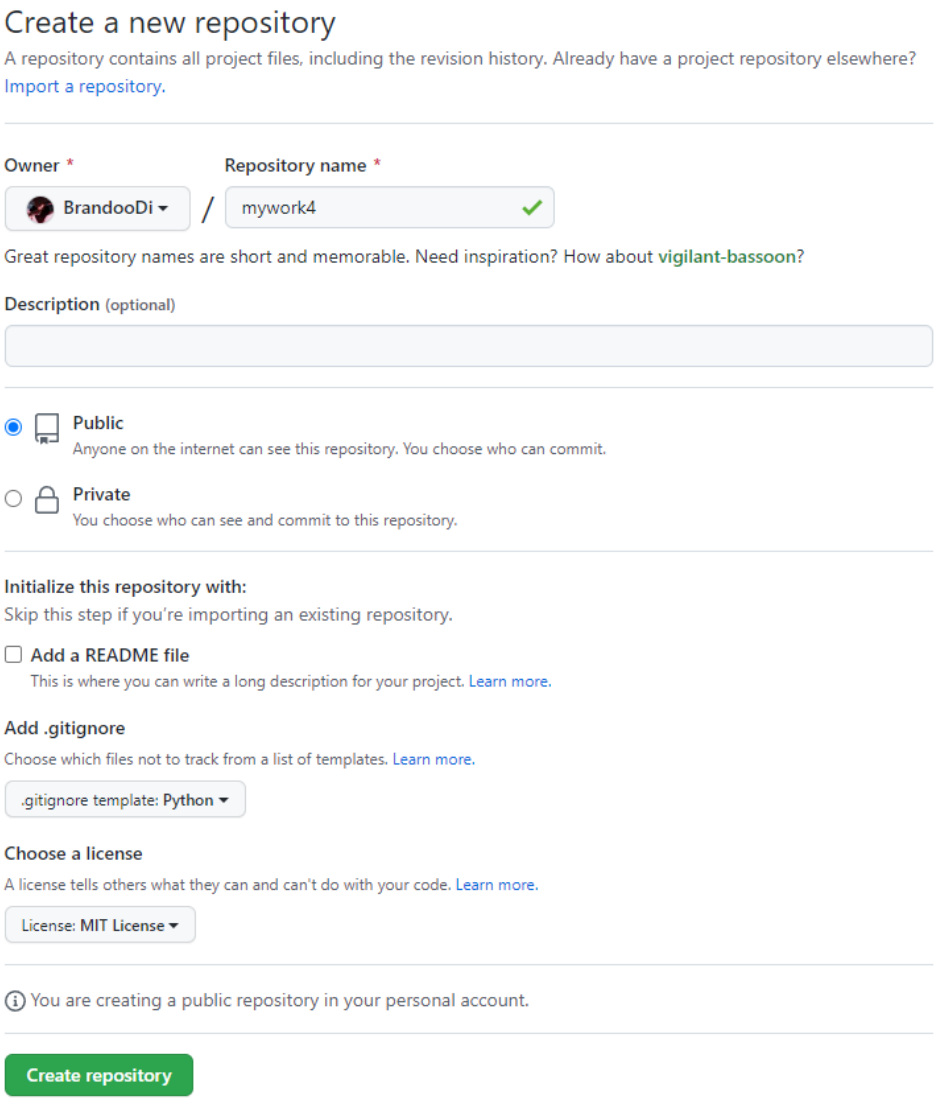


Рисунок 1 - Созданный репозиторий

2. Выполнил клонирование созданного репозитория.

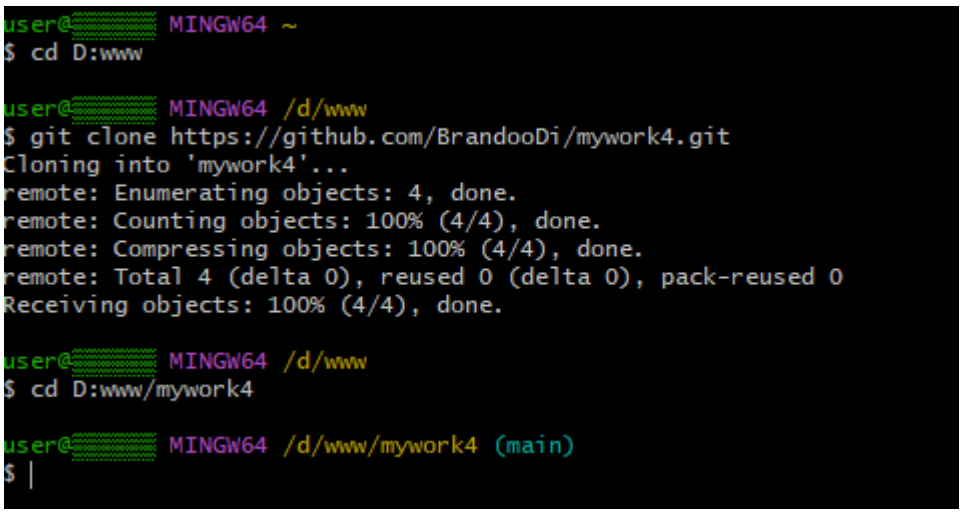
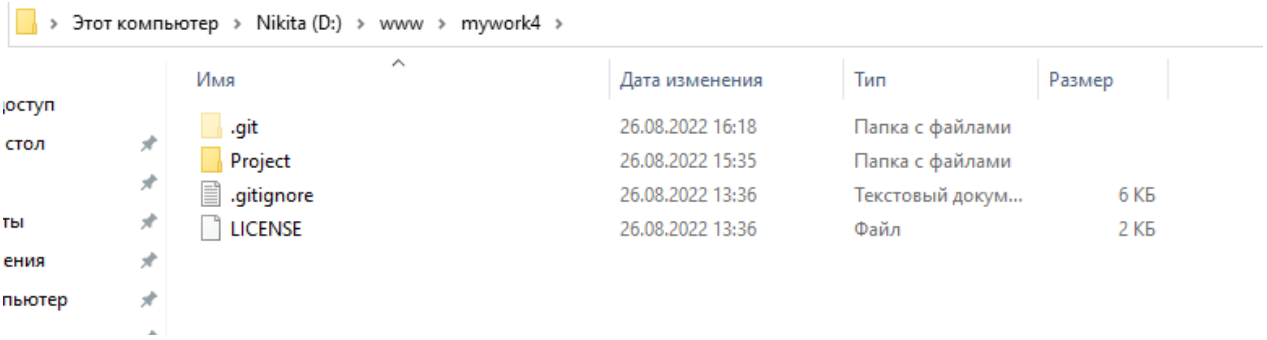


Рисунок 2 - Клонирование репозитория

3. Дополнил файл .gitignore необходимыми правилами для работы с IDE PyCharm.

Рисунок 3 - Файл .gitignore

4. Создал проект PyCharm в папке репозитория.



5. Проработал примеры лабораторной работы.

Пример 1. Составить программу с использованием конструкции ветвления и вычислить значение функции:

$$y = \begin{cases} 2x^2 + \cos x, & x \le 3.5, \\ x + 1, & 0 < x < 5, \\ \sin 2x - x^2, & x \ge 5. \end{cases}$$

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/pr1.py
Value of x? 45
y = -2024.149096475466

Process finished with exit code 0
```

Рисунок 4 - Результат выполнения программы

Пример 2. Составить программу для решения задачи: с клавиатуры вводится номер месяца от 1 до 12, необходимо для этого номера месяца вывести наименование времени года.

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/pr2.py
Введите номер месяца: 4
Весна

Process finished with exit code 0
```

Рисунок 5 - Результат выполнения программы

Пример 3. Написать программу, позволяющую вычислить конечную сумму:

$$S = \sum_{k=1}^{n} \frac{\ln kx}{k^2},$$

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/pr3.py
Value of n? 5
Value of x? 4
S = 2.4753715714873286

Process finished with exit code 0
```

Рисунок 6 - Результат выполнения программы

Пример 4. Найти значение квадратного корня из положительного числа вводимого с клавиатуры, с некоторой заданной точностью с помощью рекуррентного соотношения:

$$x_{n+1} = \frac{1}{2} \cdot \left(x_n + \frac{a}{x_n} \right).$$

В качестве начального значения примем $x_0 = 1$. Цикл должен выполняться до тех пор, пока не будет выполнено условие $|x_{n+1} - x_n| \leq \varepsilon$. Сравните со значением квадратного корня, полученным с использованием функций стандартной библиотеки. Значение $\varepsilon = 10^{-10}$.

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/pr4.py
Value of a? 5
x = 2.23606797749979
X = 2.23606797749979

Process finished with exit code 0
```

Рисунок 7 - Результат выполнения программы

Пример 5. Вычислить значение специальной (интегральной показательной) функции:

$$\mathrm{Ei}(x) = \int_{-\infty}^{x} \frac{\exp t}{t}\, dt = \gamma + \ln x + \sum_{k=1}^{\infty} \frac{x^k}{k \cdot k!},$$

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/pr5.py
Value of x? 5
Ei(5.0) = 40.18527535579794

Process finished with exit code 0
```

Рисунок 8 - Результат выполнения программы

6. Выполнил индивидуальные задания:

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

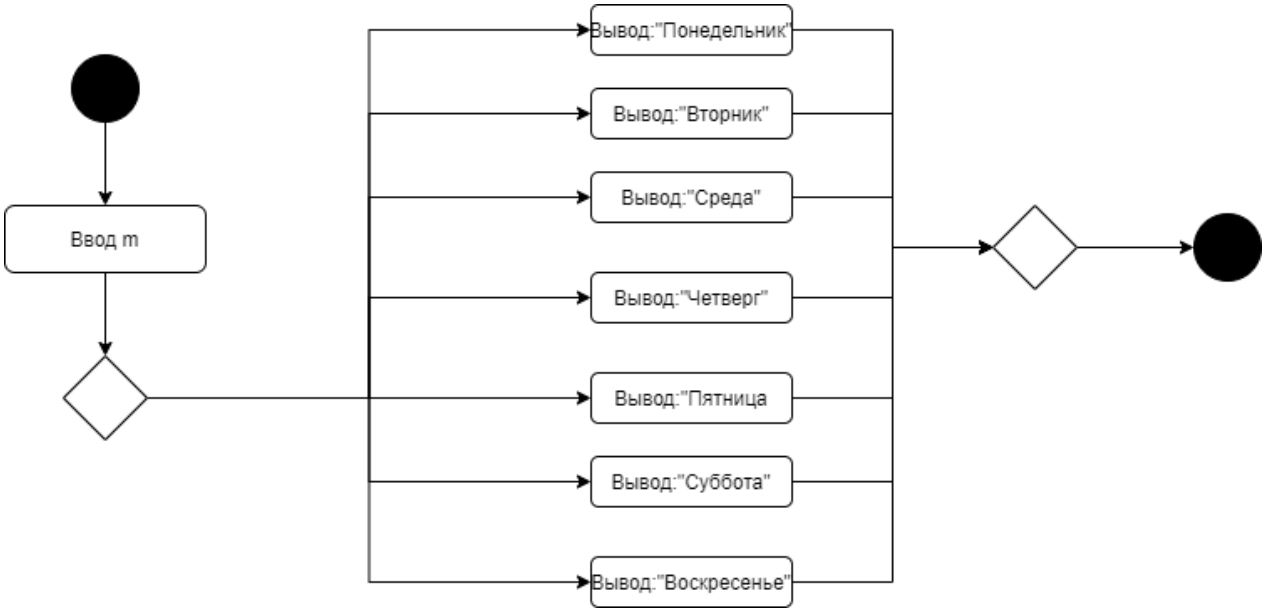


Рисунок 9 – UML-диаграмма деятельности

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/ind1.py
Введите номер дня в недели: 3
Среда

Process finished with exit code 0
```

Рисунок 10 - Результат выполнения программы Задание 1

Задание 2:Напечатать три данных действительных числа, и сначала в порядке их возрастания, затем - в порядке убывания.

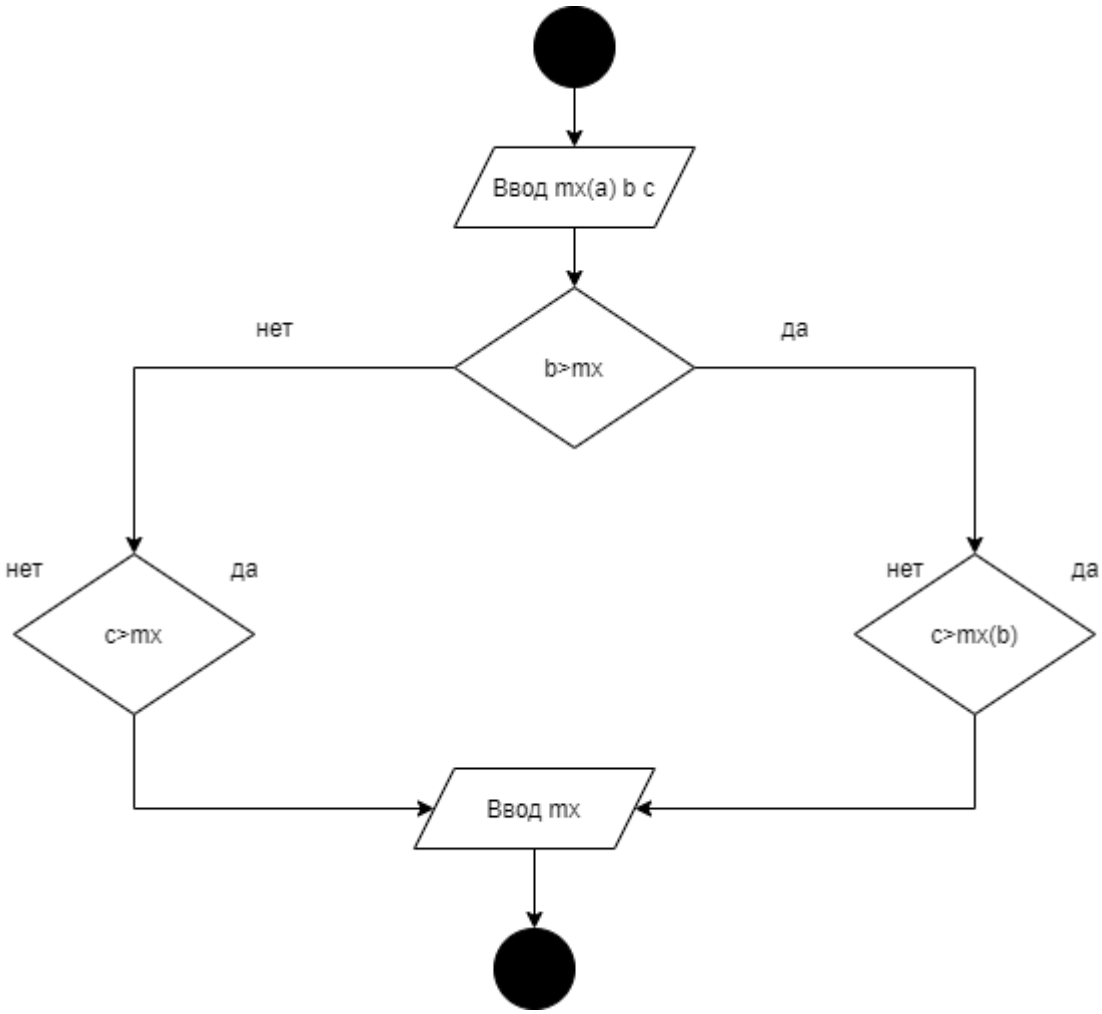


Рисунок 11 – UML-диаграмма деятельности

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/ind2.py
7
8
5
8

Process finished with exit code 0
```

Рисунок 12 - Результат выполнения программы 2

Задание 3 : Из трех действительных чисел , и выбрать те, модули которых не меньше 4.

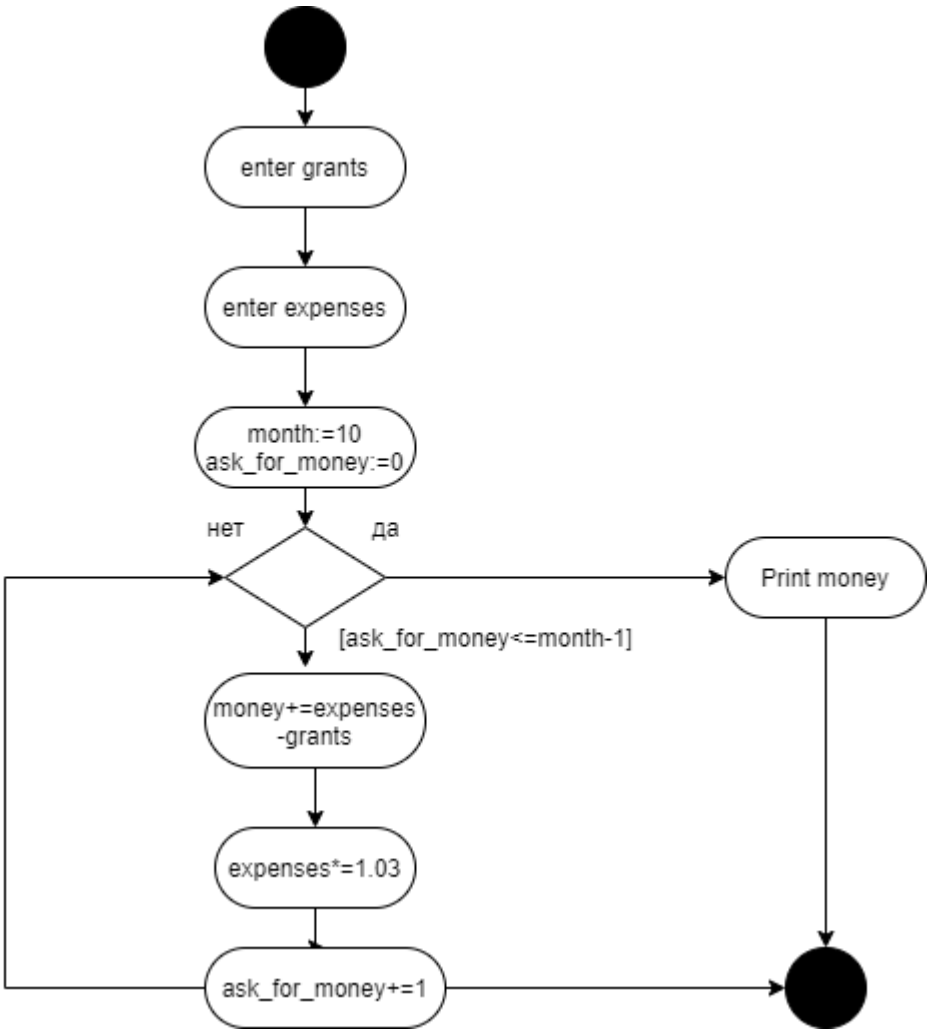


Рисунок 13 – UML-диаграмма деятельности

```
D:\www\mywork4\Project\venv\Scripts\python.exe D:/www/mywork4/Project/ind3.py
ежемесячная стипендия 2000
ежемесячные расходы 2100
4074.15

Process finished with exit code 0
```

Рисунок 14 - Результат выполнения программы 3

Задание повышенной сложности : Составить программу и произвести вычисления вычисление значения специальной функции по ее разложению в ряд с точностью $\varepsilon = 10^{-10}$, аргумент функции вводится с клавиатуры.

Дилогарифм:

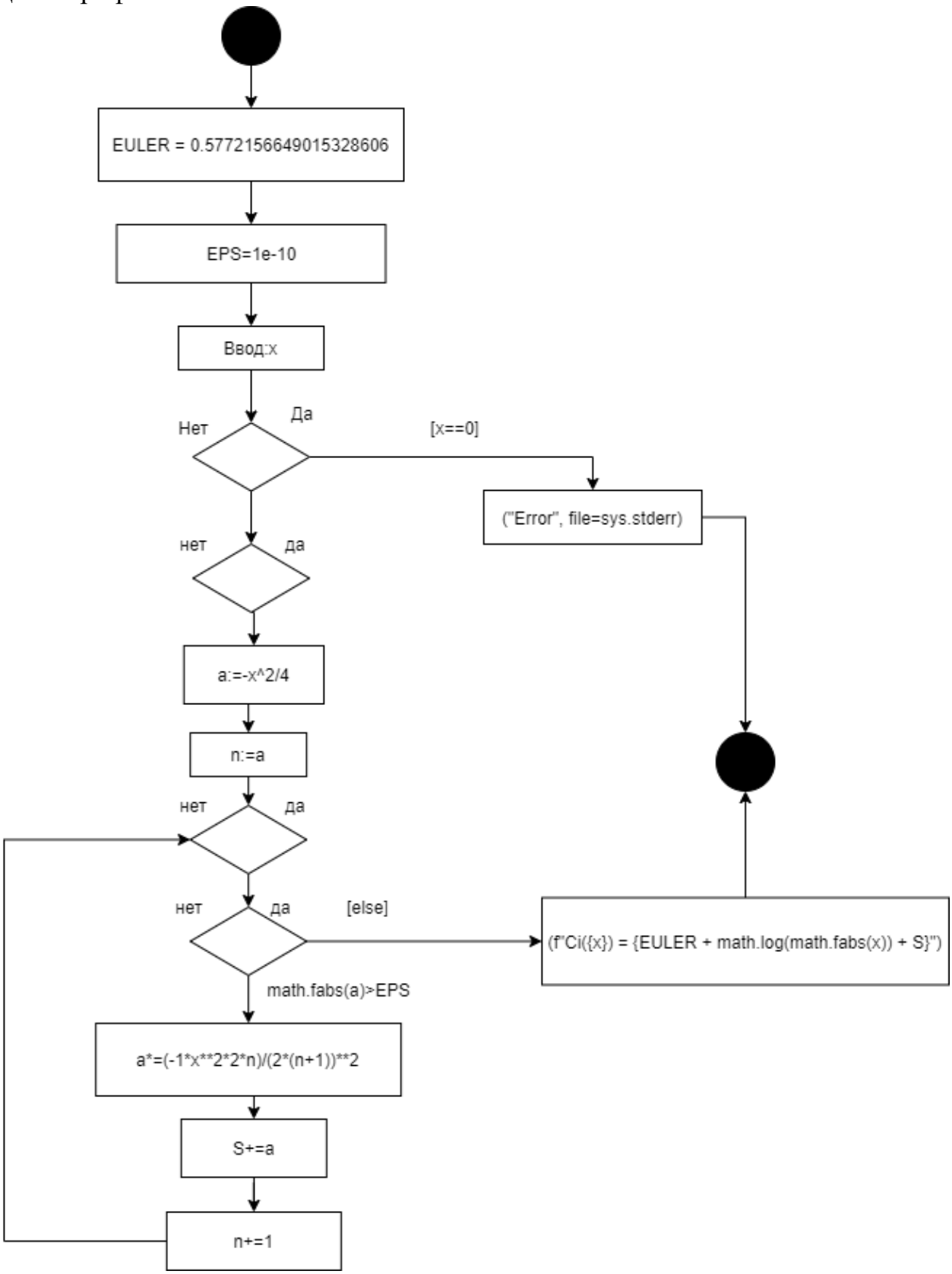


Рисунок 15 – UML-диаграмма деятельности

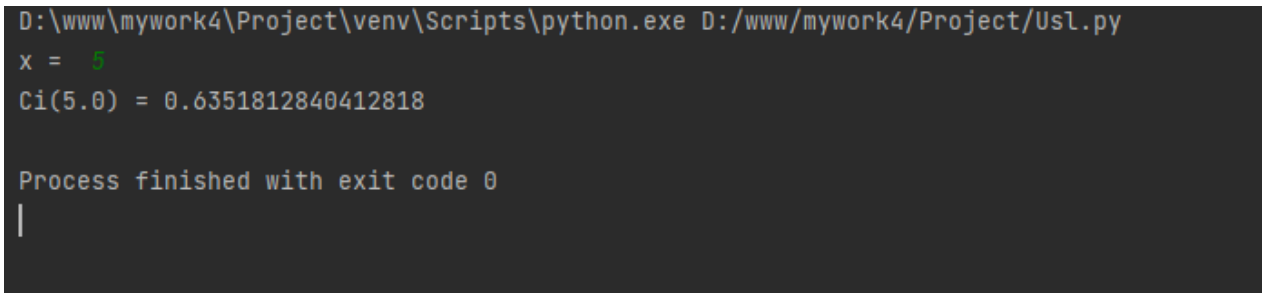


Рисунок 16 - Результат выполнения программы усложненной

Контрольные вопросы:

1. Для чего нужны диаграммы деятельности UML?

Диаграммы деятельности - это один из пяти видов диаграмм, применяемых в UML для моделирования динамических аспектов поведения системы. Диаграмма деятельности - это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, однако, по сравнению с последней, у ней есть явные преимущества: поддержка многопоточности и объектно-ориентированного проектирования. Диаграмма деятельности (Activity diagram) показывает поток переходов от одной деятельности к другой.

2. Что такое состояние действия и состояние деятельности?

В потоке управления, моделируемом диаграммой деятельности, происходят различные события. Вы можете вычислить выражение, в результате чего изменяется значение некоторого атрибута или возвращается некоторое значение. Также, например, можно выполнить операцию над объектом, послать ему сигнал или даже создать его или уничтожить. Все эти выполняемые атомарные вычисления называются состояниями действия, поскольку каждое из них есть состояние системы, представляющее собой выполнение некоторого действия.

Можно считать, что состояние действия - это частный вид состояния деятельности, а конкретнее – такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции. А состояние деятельности можно

представлять себе 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?

В UML переход представляется простой линией со стрелкой. Поток управления должен где-то начинаться и заканчиваться (разумеется, если это не бесконечный поток, у которого есть начало, но нет конца). Как показано на рисунке, вы можете задать как начальное состояние (закрашенный кружок), так и конечное (закрашенный кружок внутри окружности).

Точка ветвления представляется ромбом. В точку ветвления может входить ровно один переход, а выходить – два или более. Для каждого исходящего перехода задается булевское выражение, которое вычисляется только один раз при входе в точку ветвления. Ни для каких двух исходящих переходов эти сторожевые условия не должны одновременно принимать значение «истина», иначе поток управления окажется неоднозначным. Но эти условия должны покрывать все возможные варианты, иначе поток остановится.

Для удобства разрешается использовать ключевое слово else для пометки того из исходящих переходов, который должен быть выбран в случае, если условия, заданные для всех остальных переходов, не выполнены.

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. Программа разветвляющейся структуры реализует такой алгоритм. В программе разветвляющейся структуры имеется один или несколько условных операторов. Для программной реализации условия используется логическое

выражение. В сложных структурах с большим числом ветвей применяют оператор выбора

5. Чем отличается разветвляющийся алгоритм от линейного?

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

Алгоритм циклической структуры - это алгоритм, в котором происходит многократное повторение одного и того же участка программы.

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

Оператор ветвления if позволяет выполнить определенный набор инструкций в зависимости отнекоторого условия.

После оператора if записывается выражение. Если это выражение истинно, то выполняются инструкции, определяемые данным оператором. Выражение является истинным, если его результатом является число не равное нулю, непустой объект, либо логическое True. После выражения нужно поставить двоеточие “:”.

Бывают случаи, когда необходимо предусмотреть альтернативный вариант выполнения программы. Т. е. при истинном условии нужно выполнить один набор инструкций, при ложном – другой. Для этого используется конструкция if – else.

Для реализации выбора из нескольких альтернатив можно использовать конструкцию if – elif – else.

7. Какие операторы сравнения используются в Python?

В языках программирования используются специальные знаки, подобные тем, которые используются в математике: > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), == (равно), != (не равно).

8. Что называется простым условием? Приведите примеры.

Логические выражения типа kByte >= 1023 являются простыми, так как в них выполняется только одна логическая операция.

9. Что такое составное условие? Приведите примеры.

Может понадобиться получить ответа "Да" или "Нет" в зависимости от результата выполнения двух простых выражений. Например, "на улице идет снег или дождь", "переменная news больше 12 и меньше 20".

10. Какие логические операторы допускаются при составлении сложных условий?

В таких случаях используются специальные операторы, объединяющие два и более простых логических выражения. Широко используются два оператора – так называемые логические И (and) и ИЛИ (or).

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

Может.

12. Какой алгоритм является алгоритмом циклической структуры?

Алгоритм циклической структуры - это алгоритм, в котором происходит многократное повторение одного и того же участка программы.

Такие повторяемые участки вычислительного процесса называются циклами.

13. Типы циклов в языке Python.

Различают детерминированные циклы с заранее известным числом повторений и итерационные циклы, в которых число повторений заранее неизвестно.

14. Назовите назначение и способы применения функции range .

Функция range возвращает неизменяемую последовательность чисел в виде объекта range.

Функция range хранит только информацию о значениях start, stop и step и вычисляет значения по мере необходимости. Это значит, что независимо от размера диапазона, который описывает функция range, она всегда будет занимать фиксированный объем памяти.

15. Как с помощью функции range организовать перебор значений от

15 до 0 с шагом 2?

list(range(15, 0, -2))

16. Могут ли быть циклы вложенными?

Могут.

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

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

Несмотря на то, что в бесконечном цикле никогда не выполняется условие выхода из него, можно произвести его остановку, например, при помощи оператора break. Используя данный оператор, можно вызвать немедленное прерывание цикла, даже если условие цикла ещё не было выполнено.

18. Для чего нужен оператор break ?

Оператор break предназначен для досрочного прерывания работы цикла while.

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

Оператор continue запускает цикл заново, при этом код, расположенный после данного оператора, не выполняется.

20. Для чего нужны стандартные потоки stdout и stderr?

В операционной системе по умолчанию присутствуют стандартных потока вывода на консоль: буферизованный поток stdout для вывода данных и информационных сообщений, а также небуферизованный поток stderr для

вывода сообщений об ошибках. По умолчанию функция print использует поток stdout. Хорошим стилем программирования является наличие вывода ошибок в стандартный поток stderr поскольку вывод в потоки stdout и stderr может обрабатываться как операционной системой, так и сценариями пользователя по разному.

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

Для того, чтобы использовать поток stderr необходимо передать его в параметре file функции print. Само же определение потоков stdout и stderr находится в стандартном пакете Python sys.

22. Каково назначение функции exit?

Помимо вывода сообщения об ошибке необходимо как-то информировать операционную систему о некорректном завершении программы. Сделать это можно, передав операционной системе код возврата. Если программа завершается успешно, то она передает код возврата равный 0 (любая программа на Python делает это по умолчанию). Если в процессе выполнения программы произошли ошибки, программа должна передать операционной системе код возврата отличный от нуля. В Python завершить программу и передать операционной системе заданный код возврата можно посредством функции exit.

Вывод: приобрел навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоил операторы языка Python версии 3.x if , while , for , break и continue , позволяющие реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.