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

Кафедра инфокоммуникаций

Лабораторная работа 4

Условные
операторы и циклы в языке Python3

Выполнил студент группы ИВТ-б-о-20-1
Симанский М.Ю « »_____20__г.
Подпись студента_____
Работа защищена « »_____20__г.
Проверил Воронкин Р.А. _____
(подпись)

Цель работы: приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3 if , while , for , break и continue , позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Ссылка на репозиторий: https://github.com/MaksimSimanskiy/lab4.git

Задание 1

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

Код

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
m = int(input('Введите число от 1 до 7 '))
if m == 7:
    print('Воскресение')
elif m == 6:
    print('Суббота')
elif m == 5:
    print('Пятница')
elif m == 4:
    print('Четверг')
elif m == 3:
    print('Среда')
elif m == 2:
    print('Вторник')
elif m == 1:
    print('Понедельник')
else:
    print('Number not include in sequence 1-7')
```

UML-диаграмма

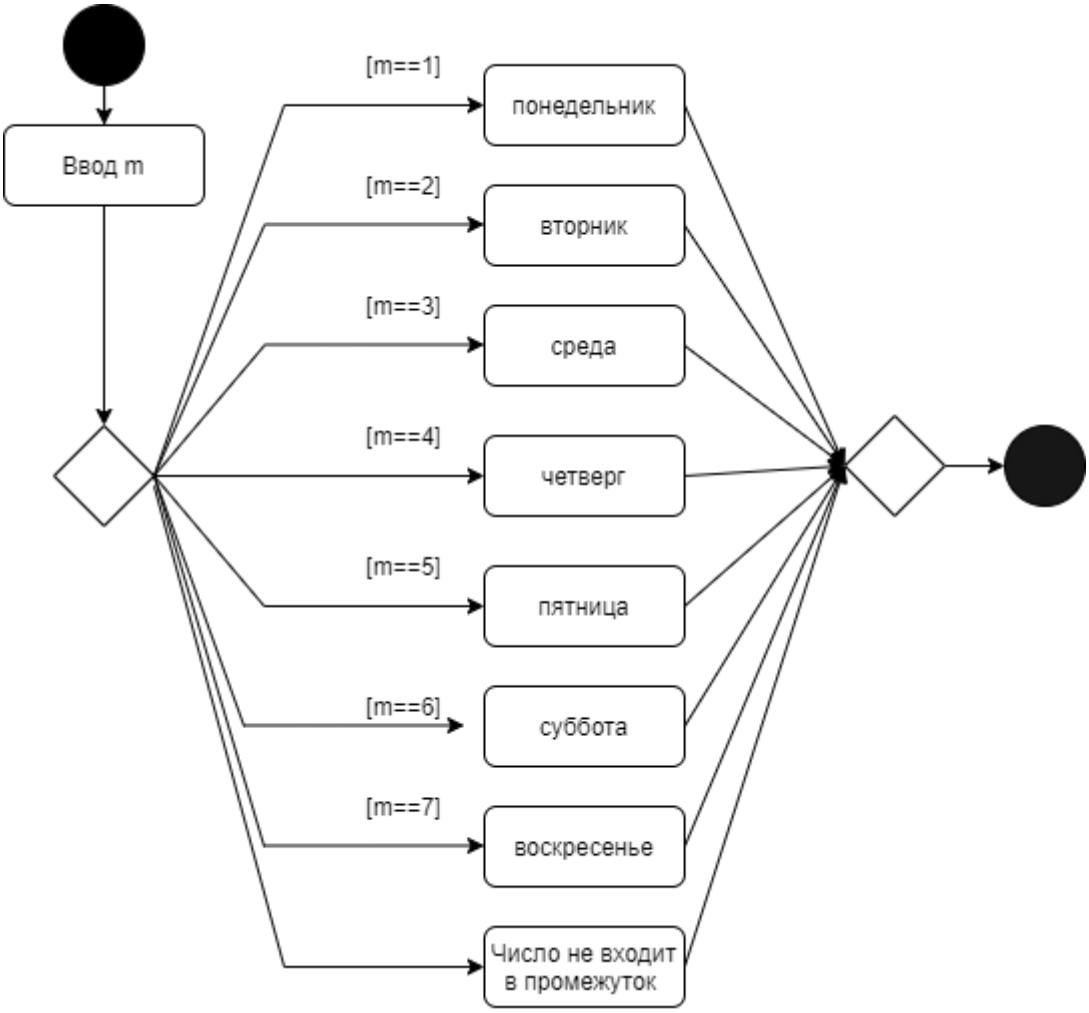

Рисунок 1 – UML-диаграмма

Результат

```
Введите число от 1 до 7 4
Четверг
```

Рисунок 2 – Результат работы

Задание 2

15. Составить программу решения квадратного уравнения. Выводить также комплексные решения.

Код

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
print('Решаем уравнение a•x²+b•x+c=0')
a = float(input('Введите значение a: '))
b = float(input('Введите значение b: '))
c = float(input('Введите значение c: '))

d = b ** 2 - 4 * a * c
print('Дискриминант = ' + str(d))
if d == 0:
    x = -b / (2 * a)
    print('x = ' + str(x))
else:
    x1 = (-b + d ** 0.5) / (2 * a)
    x2 = (-b - d ** 0.5) / (2 * a)
    print('x₁ = ' + str(x1))
    print('x₂ = ' + str(x2))
```

UML-диаграмма

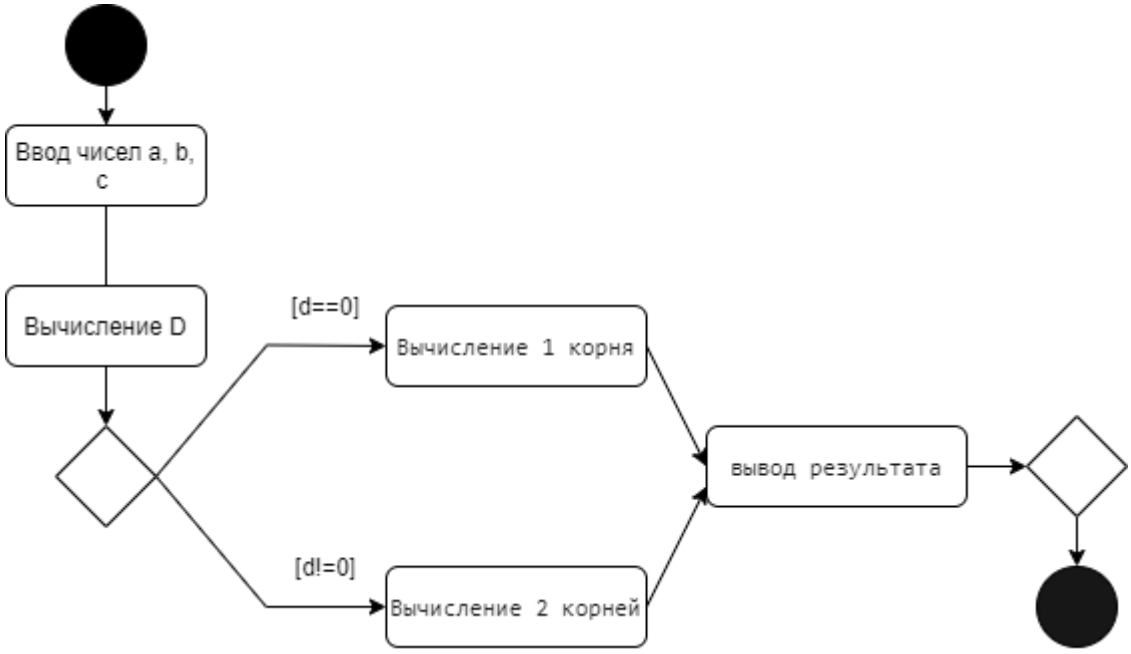

Рисунок 3 – UML-диаграмма

Результат

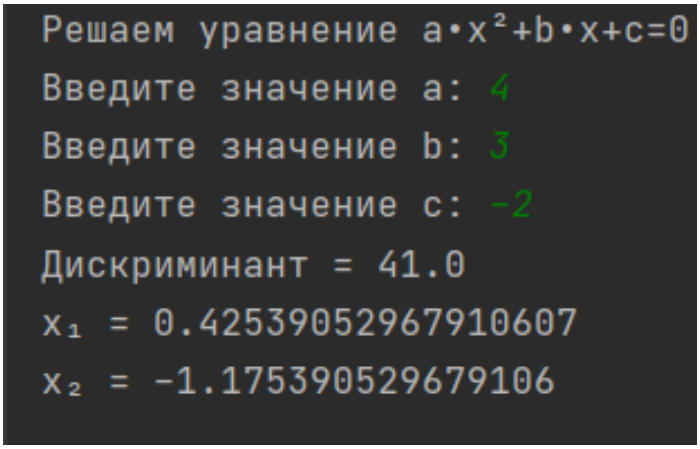

Рисунок 4 – Результат работы

Задание 3

14. Вычислить сумму всех n-значных чисел (1<=n<=4).

Код

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import math
n = int(input('Введите значение n от 1 до 4: '))
if(n < 1 or n > 4):
    print("Число вне интервала")
    exit(1)
a = math.pow(10, n - 1)
b = math.pow(10, n) - 1
s = (b * (b + 1) - a * (a - 1)) / 2
print("Результат = ",s)
```

UML-диаграмма

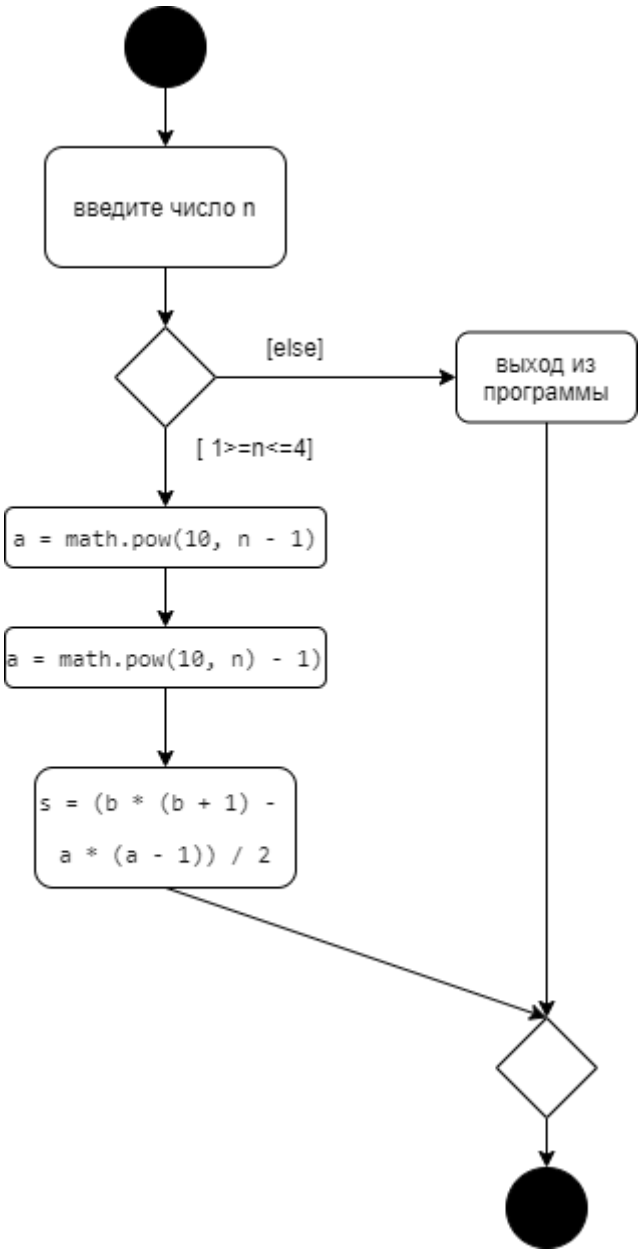

Рисунок 5 – UML- диаграмма

Результат

```
Введите значение n от 1 до 4: 1
Результат =  45.0
```

Рисунок 6 – Результат работы

Задание повышенной сложности. Вариант 4

4. Интегральный гиперболический косинус:

$$\operatorname{Chi}(x) = \gamma + \ln x + \int_0^x \frac{\operatorname{ch} t - 1}{t}\, dt = \gamma + \ln x + \sum_{n=1}^{\infty} \frac{x^{2n}}{(2n)(2n)!}.$$

Текущий член ряда задается выражением:

$$a_n = \frac{x^{2n}}{(2n)(2n)!}$$

Следующий член ряда с использованием свойств факториала:

$$a_{n+1} = \frac{a^{2n+2}}{(2n+1)(2n+1)!} = \frac{x^{2n+1}}{(2n+1)^2 * 2n!}$$

Найдем отношение следующего и текущего членов ряда:

$$\frac{a_{n+1}}{a_n} = \frac{x^{2n+1}}{(2n+1)^2 * 2n!} : \frac{x^{2n}}{(2n)(2n)!}$$

Результат:

$$\frac{2nx}{(2n+1)^2}$$

Код

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import math
import sys
# Постоянная Эйлера.
EULER = 0.5772156649015328606
# Точность вычислений.
EPS = 1e-10

if __name__ == '__main__':
    x = float(input("Value of x? "))
if x == 0:
    print("Illegal value of x", file=sys.stderr)
    exit(1)
a = x
S, n = a, 1
# Найти сумму членов ряда.
while math.fabs(a) > EPS:
    a *= 2 * n * x / pow((2 * n + 1), 2)
    S += a
    n += 1
    # Вывести значение функции.
    print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")
```

UML-диаграмма

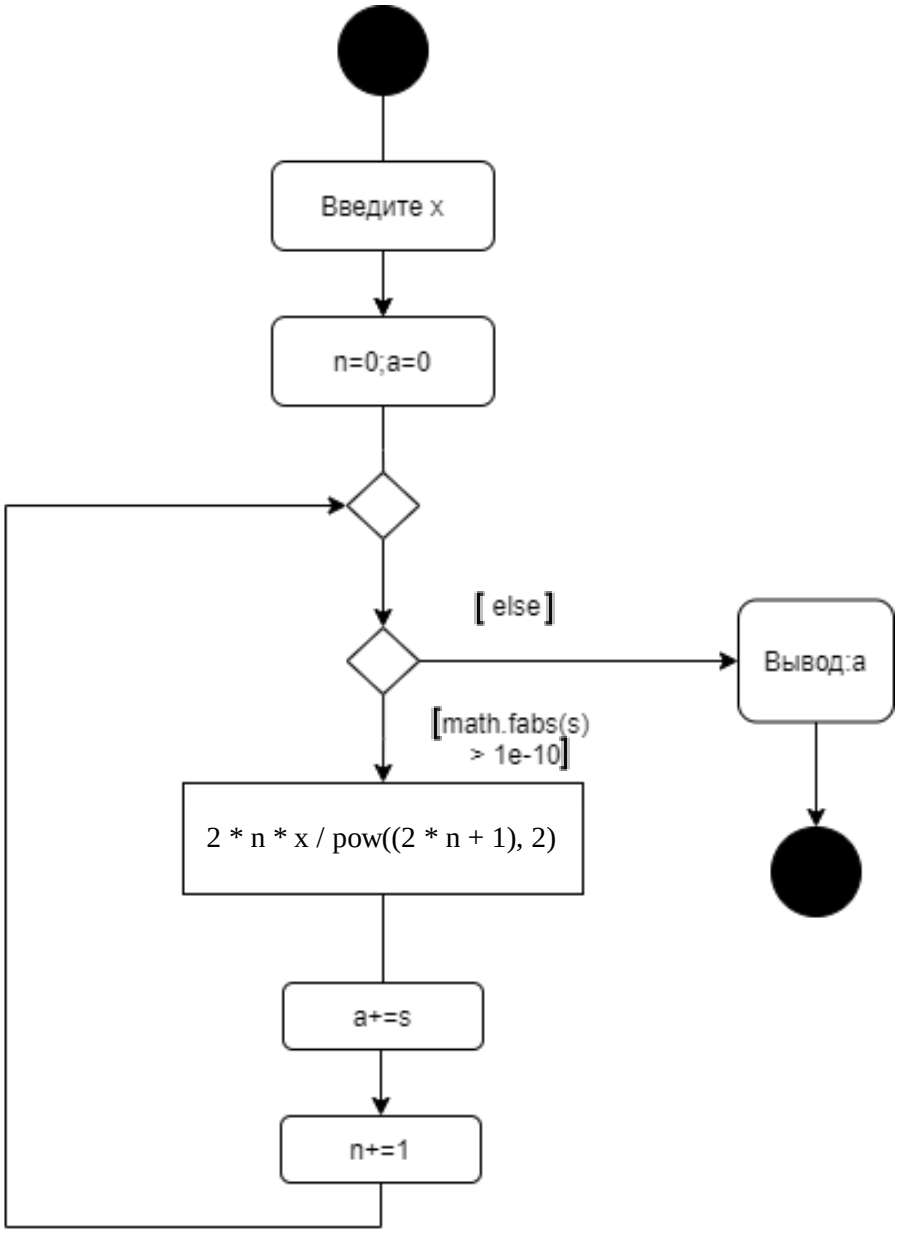

Рисунок 7 – UML-диаграмма

Результат

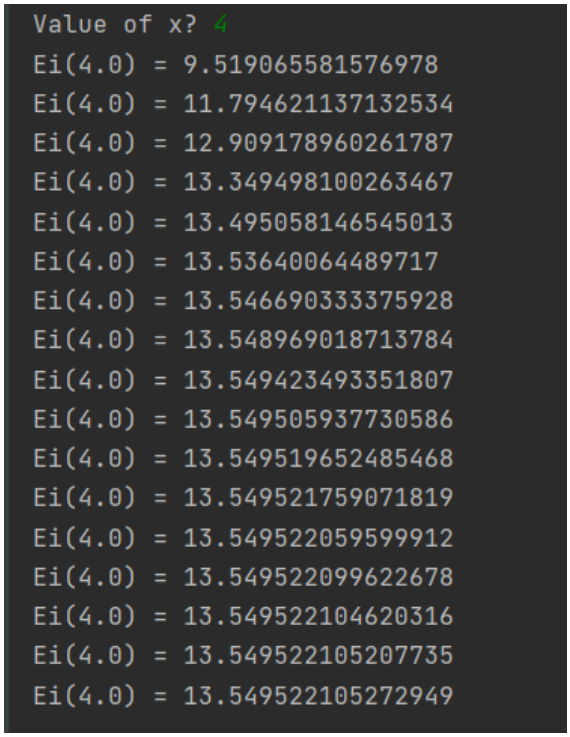

Рисунок 8 – Результат работы

Ответы на контрольные вопросы

1. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем. UML пригоден для моделирования любых систем: от информационных систем масштаба предприятия до распределенных Web-приложений и даже встроенных систем реального времени
2. 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.
3. Переходы. Когда действие или деятельность в некотором состоянии завершается, поток управления сразу переходит в следующее состояние действия или деятельности. Для описания этого потока используются

переходы, показывающие путь из одного состояния действия или деятельности в другое. В UML переход представляется простой линией со стрелкой

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

5. Линейный идет лишь в одном направлении , а в разветвляющемся возможны разные исходы.

6. Условные операторы:if и while.Условный оператор позволяет выполнять действия в зависимости от булева значения условия.

7. Операторы сравнения: ==,!=, <>,>, <=

8. If a>4: print('Hello')

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

10. В сложных условия можно использовать операторы: &&, ||

11. Да операторы ветвления могу иметь внутри другие ветвления в виде множественного ветвления

12. Оператор for выполняет указанный набор инструкций заданное количество раз, которое определяется количеством элементов в наборе.

13. Существует 2 вида циклов: while и for

14. Функция range возвращает неизменяемую последовательность чисел в виде объекта range.

```
>>> range(5)
range(0, 5)
>>> list(range(5))
[0, 1, 2, 3, 4]
```

15. >>> list(range(15, 0, 2))

16. Python позволяет вкладывать циклы друг друга. Вложенный цикл – это цикл, который встречается внутри другого цикла.

17. Бесконечный цикл while — это цикл, в котором условие никогда не становится ложным. Это значит, что тело исполняется снова и снова, а цикл никогда не заканчивается. Прервать его можно с помощью break и continue

18. Оператор break предназначен для досрочного прерывания работы цикла while.

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

20. 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.

21. Для того, чтобы использовать поток stderr необходимо передать его в параметре file функции print.

22. В Python завершить программу и передать операционной системе заданный код возврата можно посредством функции exit.