

**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 программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1 Кучеренко С. Ю. « » 2022г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 2022г. 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.

## Выполнение работы:

### Учебные задачи:

**Пример 1.** Составить UML-диаграмму деятельности и программу с использованием конструкции ветвления и вычислить значение функции

$$y = \begin{cases} 2x^2 + \cos x, & x \leq 3.5, \\ x + 1, & 0 < x < 5, \\ \sin 2x - x^2, & x \geq 5. \end{cases} \quad (1)$$

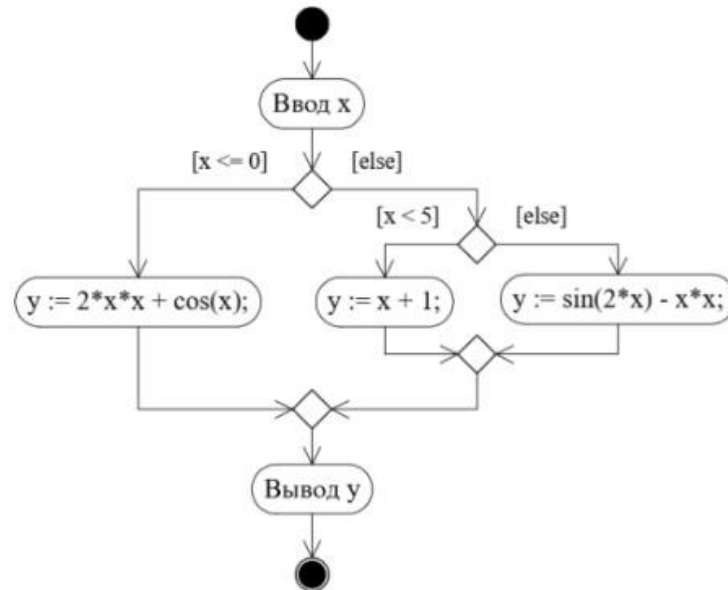


Рисунок 1 – UML-диаграмма деятельности

### Код:

```
#!/usr/bin/env python3 #
-*- coding: utf-8 -*-
import math
if __name__ ==
'__main__':
x = float(input("Value of x? "))
if x <= 0:
y = 2 * x * x + math.cos(x)
elif x < 5:
y = x + 1
else:
y = math.sin(x) - x * x
print(f"y = {y}")
```

```
/Users/svetik/.conda/envs/Laba5/bin/python
Value of x? -1
y = 2.5403023058681398

Process finished with exit code 0
```

Рисунок 2 – Результат выполнения программы при  $x = -1$

```
/Users/svetik/.conda/envs/Laba5/bin/python3
Value of x? -1
y = 4.0

Process finished with exit code 0
```

Рисунок 3 – Результат выполнения программы при  $x = 3$

```
/Users/svetik/.conda/envs/Laba5/bin/python3
Value of x? 3
y = -36.27941549819893

Process finished with exit code 0
```

Рисунок 4 – Результат выполнения программы при  $x = 6$

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

*Решение:* Составим UML-диаграмму деятельности вычисления значения функции (рис. 4.6).

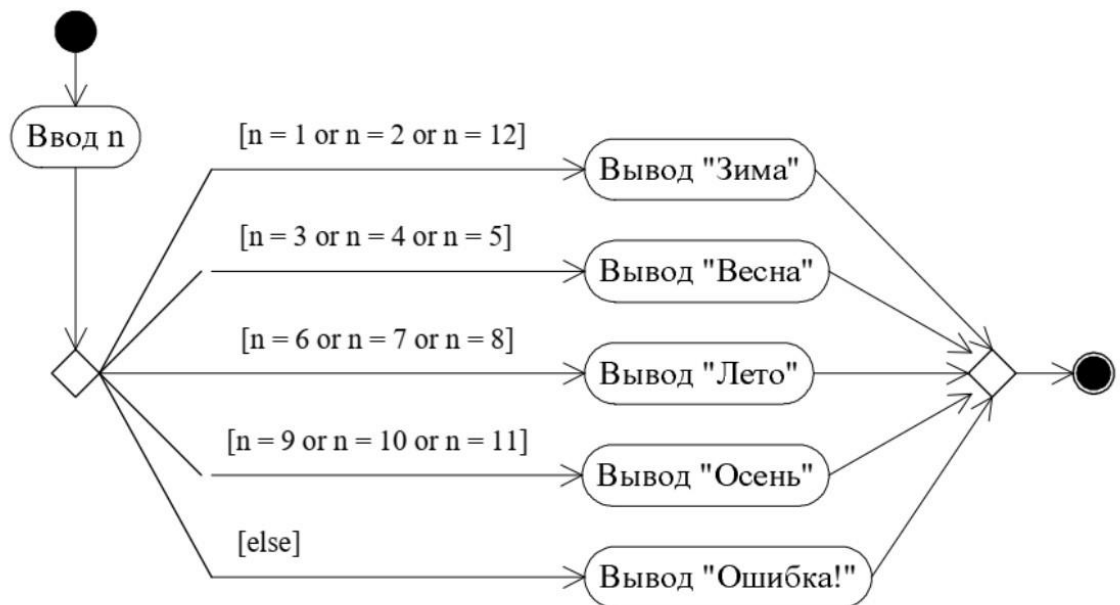


Рисунок 5 – UML-диаграмма деятельности

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys
if __name__ ==
'__main__':
    n = int(input("Введите номер месяца: "))
    if n == 1 or n == 2 or n ==
12:
        print("Зима")
    elif n ==
3 or n == 4 or n == 5:
        print("Весна")
    elif n ==
6 or n == 7 or n == 8:
        print("Лето")
    elif n == 9
or n == 10 or n == 11:
        print("Осень")
    else:
        print("Ошибка!", file=sys.stderr)
    exit(1)
```

```
/Users/svetik/.conda/envs/Laba5/bin
Введите номер месяца: 8
Лето

Process finished with exit code 0
```

Рисунок 6 – Результат выполнения программы при x = 8

```
Введите номер месяца: 12
Зима

Process finished with exit code 0
```

Рисунок 7 – Результат выполнения программы при x = 12

```
Введите номер месяца: 32
Ошибка!

Process finished with exit code 1
```

Рисунок 8 – Результат выполнения программы при x = 32

**Пример 3.** Составить UML-диаграмму деятельности и написать программу, позволяющую вычислить конечную сумму:

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

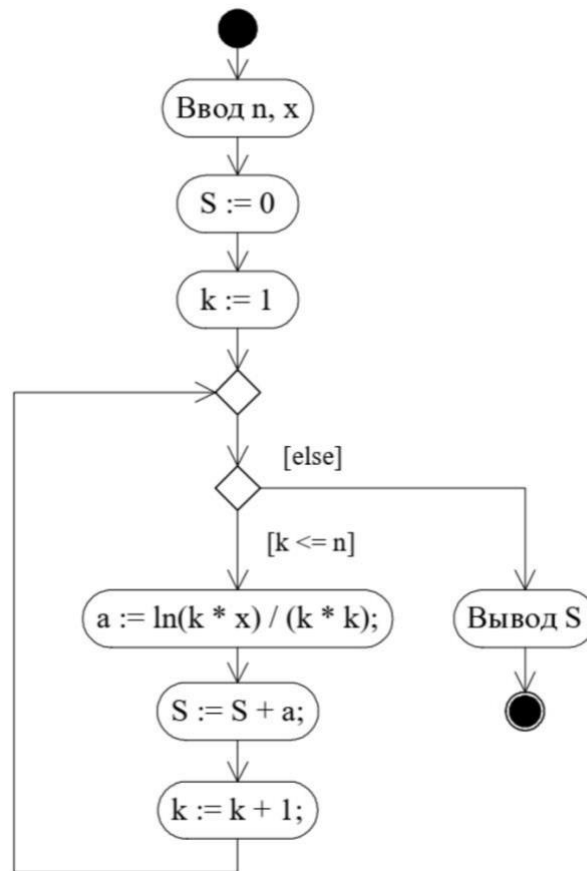


Рисунок 9 – UML-диаграмма деятельности

Код:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math
if __name__ ==
'__main__':
    n = int(input("Value of n? "))
    x = float(input("Value of x? "))

    S = 0.0
    for k in range(1, n + 1):
        a = math.log(k * x) / (k * k)
        S += a
        print(f"S = {S}")
```

```
Value of n? 5
Value of x? 7
S = 3.294431456662813
```

Рисунок 10 – Результат выполнения программы при n = 5, x = 7

```
Value of n? 1
Value of x? 4
S = 1.3862943611198906
```

Рисунок 11 – Результат выполнения программы при n = 1, x = 4

**Пример 4.** Найти значение квадратного корня  $x = \sqrt{a}$  из положительного числа  $a$  вводимого с клавиатуры, с некоторой заданной точностью  $\varepsilon$  с помощью рекуррентного соотношения:

$$x_{n+1} = \frac{1}{2} \cdot \left( x_n + \frac{a}{x_n} \right). \quad (3)$$

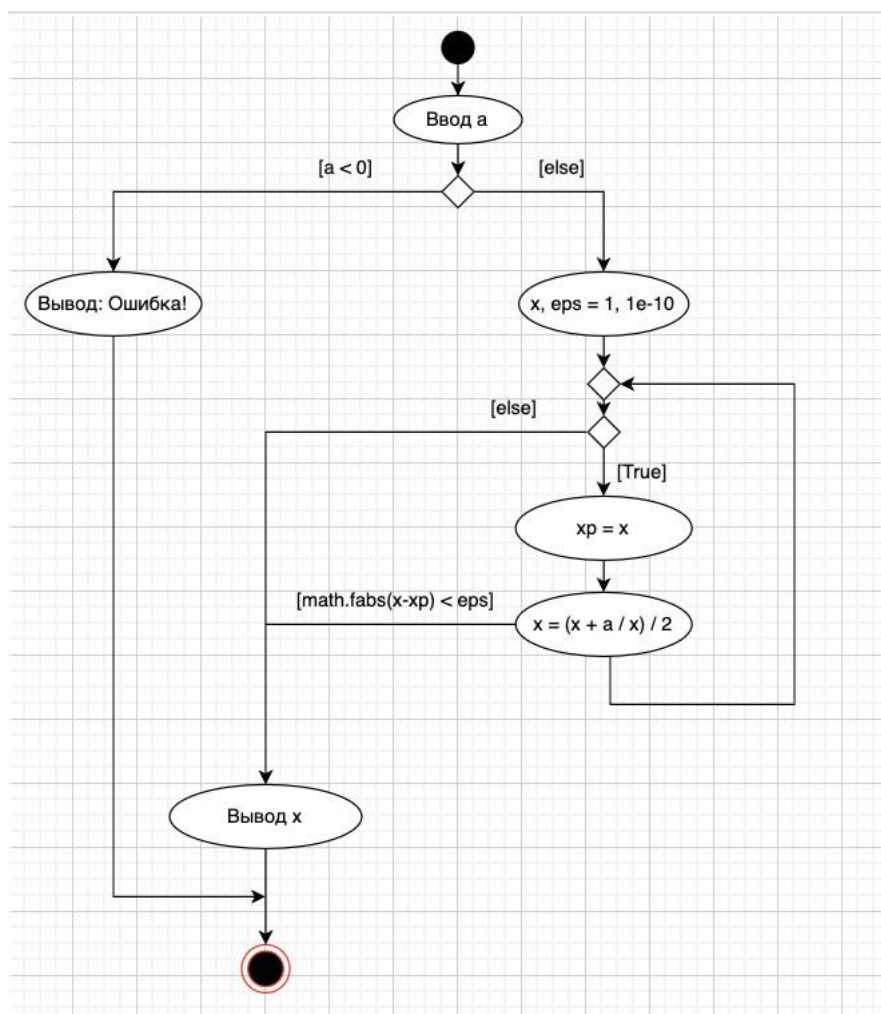
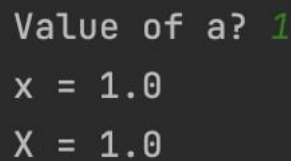


Рисунок 12 – UML-диаграмма деятельности

Код:

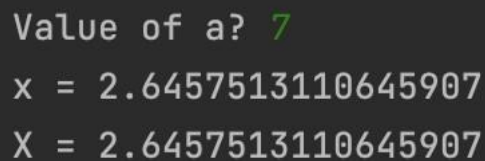
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import math import sys
if __name__ ==
'__main__':
    a = float(input("Value of a? "))    if a < 0:
print("Illegal value of a", file=sys.stderr)
exit(1)    x, eps = 1, 1e-10    while True:
xp = x
    x = (x + a / x) / 2
if math.fabs(x - xp) < eps:
    break

print(f"x = {x}\nX = {math.sqrt(a)}")
```



Value of a? 1  
x = 1.0  
X = 1.0

Рисунок 13 – Результат выполнения программы при a = 1



Value of a? 7  
x = 2.6457513110645907  
X = 2.6457513110645907

Рисунок 14 – Результат выполнения программы при a = 7

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

$$\text{Ei}(x) = \int_{-\infty}^x \frac{\exp t}{t} dt = \gamma + \ln x + \sum_{k=1}^{\infty} \frac{x^k}{k \cdot k!}, \quad (4)$$

где  $\gamma = 0.5772156649 \dots$  - постоянная Эйлера, по ее разложению в ряд с точностью  $\epsilon = 10^{-10}$ , аргумент  $x$  вводится с клавиатуры.

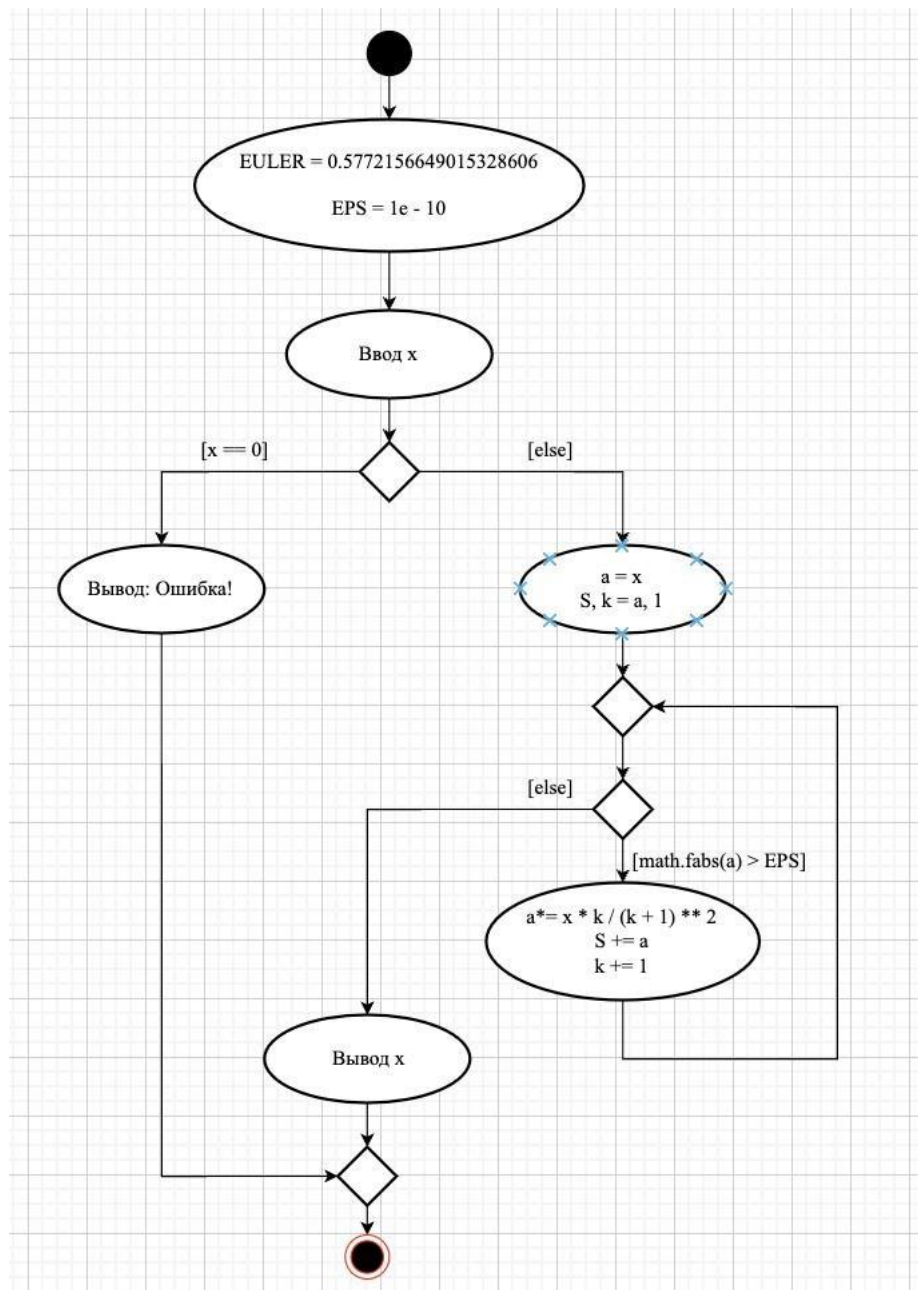


Рисунок 15 – UML-диаграмма деятельности

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, k = a, 1
    # Найти сумму членов ряда.
while math.fabs(a) > EPS:
    a *= x * k / (k + 1) ** 2
S += a      k += 1
    # Вывести значение функции.
    print(f"Ei({x}) = {EULER + math.log(math.fabs(x)) + S}")

```

```

Value of x? 4
Ei(4.0) = 19.63087447005282

```

Рисунок 16 – Результат выполнения программы при  $x = 4$

```

Value of x? -1
Ei(-1.0) = -0.21938393439629178

```

Рисунок 17 – Результат выполнения программы при  $x = -1$

**Индивидуальное задание:** решить задачу согласно варианту, составить UML-диаграмму деятельности и программу с использованием конструкций ветвления. Номер варианта необходимо получить у преподавателя.

#### Задание 1.

Вариант 9.

Вводится число экзаменов  $N \leq 20$ . Напечатать фразу «Мы успешно сдали N экзаменов», согласовав слово «экзамен» с числом N.

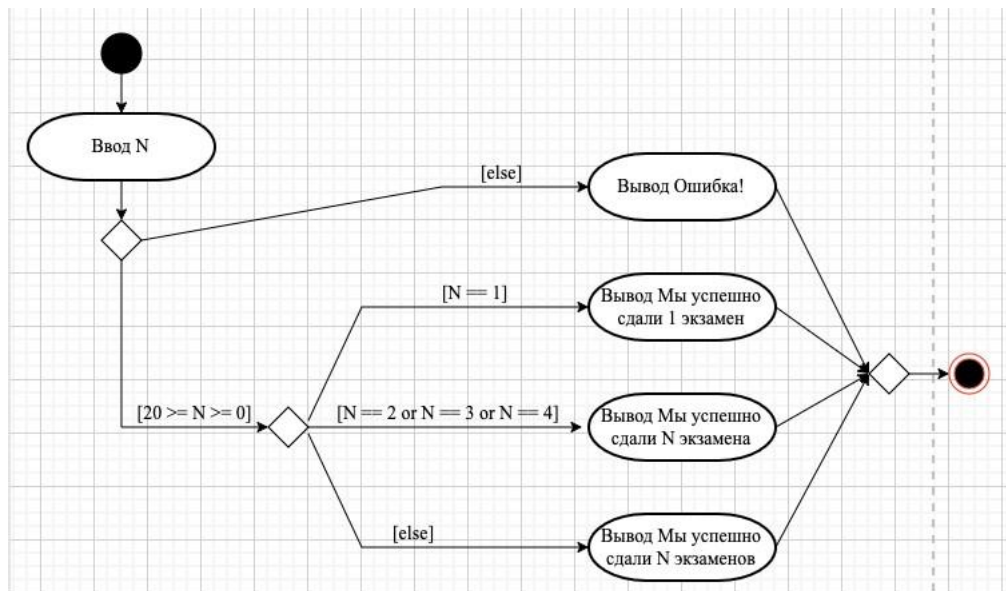


Рисунок 18 – UML-диаграмма деятельности

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ ==
'__main__':
    N = int(input("Value of N? "))
if 20 >= N >= 0:
    if N ==
1:
        print(f"Мы успешно сдали {N} экзамен")
    elif N == 2 or N == 3 or N == 4:
        print(f"Мы успешно сдали {N} экзамена")
    else:
        print(f"Мы успешно сдали {N} экзаменов")
else:
    print("Ошибка!")
  
```

```

/Users/svetik/.conda/envs/Laba5/bin/py
Value of N? 21|
Ошибка!

Process finished with exit code 0
  
```

Рисунок 19– Результат выполнения программы при N = 21

```

Value of N? 3
Мы успешно сдали 3 экзамена

Process finished with exit code 0
  
```

Рисунок 20 – Результат выполнения программы при N = 3

```
Value of N? 10
Мы успешно сдали 10 экзаменов
Process finished with exit code 0
```

Рисунок 21 – Результат выполнения программы при N = 10

## Задание 2.

### Вариант 9

Найти координаты точки пересечения прямых заданных уравнениями  $a_1x + b_1y + c_1 = 0$  и  $a_2x + b_2y + c_2 = 0$ , либо сообщить совпадают, параллельны или не существуют.

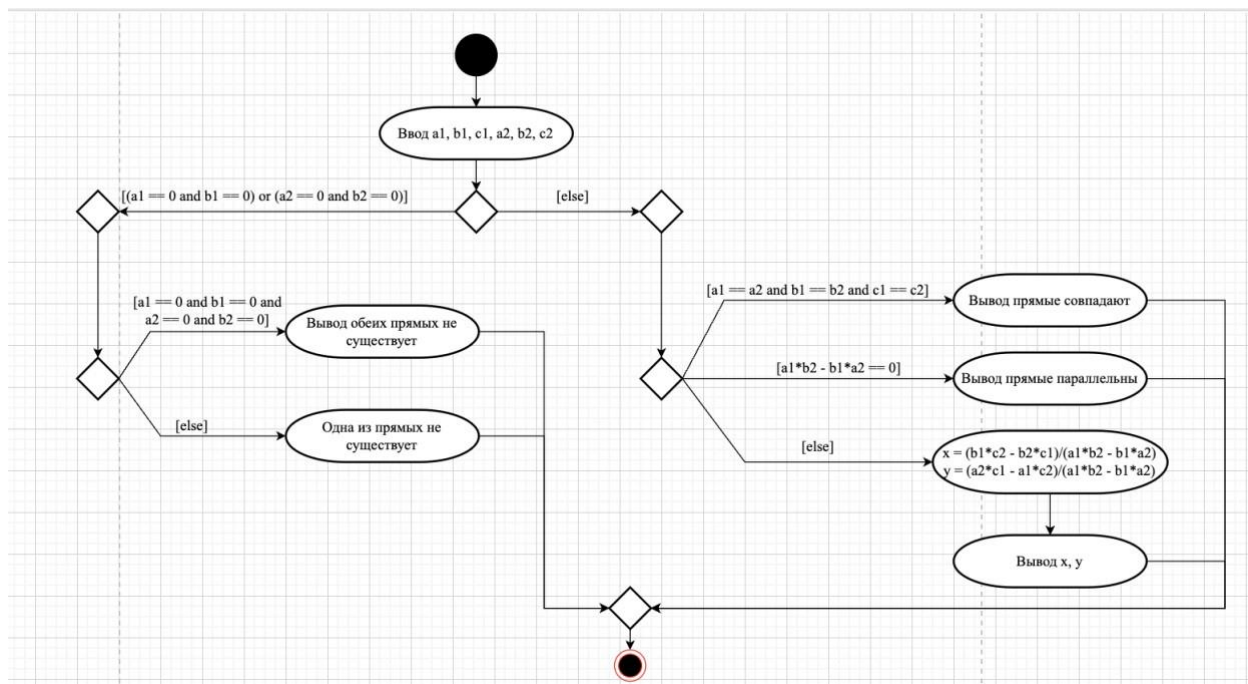


Рисунок 22 – UML-диаграмма деятельности

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ ==
'__main__':
    a1, b1, c1 = map(int, input("Введите координаты первой прямой: ").split('
'))
    a2, b2, c2 = map(int, input("Введите координаты второй прямой ").split('
'))
    if (a1 == 0 and b1 == 0) or (a2 == 0 and b2 ==
0):
        if a1 == 0 and b1 == 0 and a2 == 0 and b2
== 0:
            print("Обе прямые не существуют")
        else:
            print("Одна из прямых не существует")
    else:
        if a1 == a2 and b1 == b2 and c1 ==
c2:
            print("Прямые совпадают")
    elif a1*b2 - b1*a2 == 0:
        print("Прямые параллельны")
    else:
        x = (b1*c2 - b2*c1)/(a1*b2 - b1*a2)
        y = (a2*c1 - a1*c2)/(a1*b2 - b1*a2)
        print(f"Координаты точки пересечения: x = {x}, y = {y}")
```

```
/Users/svetik/.conda/envs/Laba5/bin/python
Введите координаты первой прямой: 0 0 1
Введите координаты второй прямой 1 2 3
Одна из прямых не существует
```

Рисунок 23 – Результат выполнения программы

```
Введите координаты первой прямой: 0 0 4
Введите координаты второй прямой 0 0 6
Обе прямые не существуют
```

Рисунок 24 – Результат выполнения программы

```
svetik/.conda/envs/Laba5/bin/python /Users/svetik/Desktop/Laba5/LR_2.2/
координаты первой прямой: 1 6 3
координаты второй прямой 4 6 1
аты точки пересечения: x = 0.6666666666666666, y = -0.6111111111111112

finished with exit code 0
```

Рисунок 25 – Результат выполнения программы при N = 10  
Задание 3.

Вариант 9.

Если к сумме цифр двузначного числа прибавить квадрат этой суммы, то снова получится это двузначное число. Найти все эти числа.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    for x in range(10, 100, 1):
        sum = x // 10 + x % 10
        if x == sum + sum**2:
            print(f"Полученные числа: {x}")
```

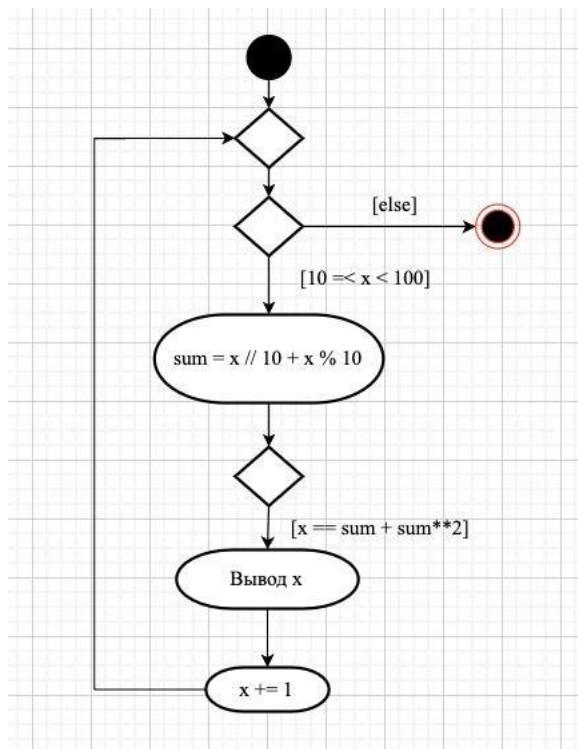


Рисунок 26 – UML-диаграмма деятельности

```
/Users/svetik/.conda/envs/Laba5/bin/
Полученные числа: 12
Полученные числа: 42
Полученные числа: 90

Process finished with exit code 0
```

Рисунок 27 – Результат выполнения программы

**Задание повышенной сложности:**

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)n!}.$$

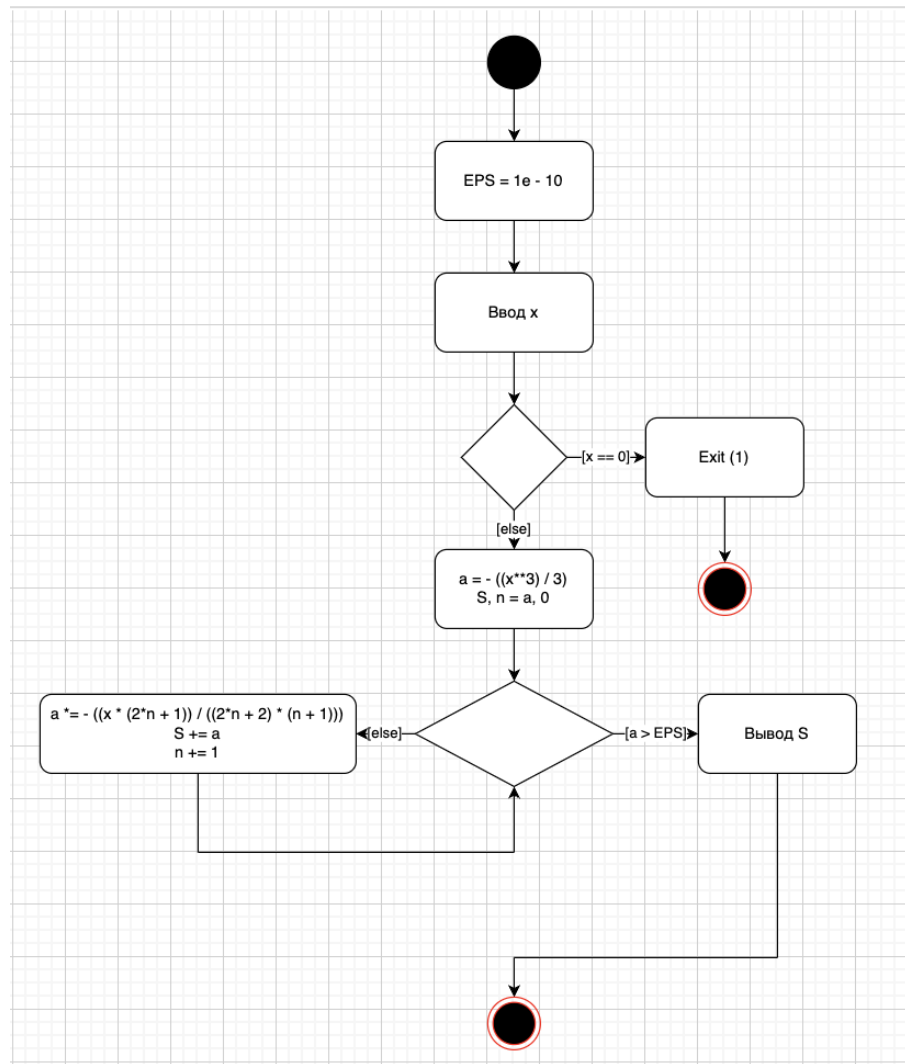


Рисунок 28 – UML-диаграмма деятельности

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import math
import sys

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**3) / 3)
    S, n = a, 0

    while math.fabs(a) > EPS:
        a *= - ((x * (2*n + 1)) / ((2*n + 2) * (n + 1)))
        S += a
        n += 1

    print(f"erf(x) = ", S)

```

```
/usr/local/bin/python3.11 /Users/sveti
Value of x? 5
erf(x) = -11.251935067162147

Process finished with exit code 0
```

Рисунок 29 – Результат выполнения программы

### Вопросы для защиты работы:

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

Дает возможность визуально представить алгоритм программы.

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

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

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

#### 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?

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?

В линейном алгоритме все этапы выполняются однократно и строго последовательно, в то время как в разветвляющемся содержащий хотя бы одно условие, в результате проверки которого совершается переход на один из нескольких возможных шагов.

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

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

Условные операторы: if, while, for.

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

If, elif, else

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

Условия, в которых выполняется только одна логическая операция, например  $x > 5$ ,  $a == 0$ .

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

Условия, в которых выполняется больше одной логической операции, например  $x > 5$  or  $a == 0$ .

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

Or, and, not.

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

Да.

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

Это вид алгоритма, в процессе выполнения которого одно или несколько действий нужно повторить несколько раз.



**13. Типы циклов в языке Python.** while, for.

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

Функция range(start, stop[, step]) генерирует серию целых чисел, от значения start до stop, с шагом step 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. Могут ли быть циклы вложенными?**

Да

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

Если условие выхода из него никогда не выполняется.

Чтобы выйти необходимо использовать оператор break.

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

Для выхода из цикла.

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

Оператор continue используется только в циклах. Он выполняет пропуск оставшейся части кода тела цикла и переходит к следующей итерации цикла.

**20. Для чего нужны стандартные потоки stdout и stderr? stdout — стандартный вывод (экран) stderr — стандартная ошибка (вывод ошибок на экран)**

**21. Как в Python организовать вывод в стандартный поток stderr? `print(..., file=sys.stderr)`.**

**22. Каково назначение функции `exit`?**

Завершить программу и передать операционной системе заданный код возврата.