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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.2**
**дисциплины «Программирование на Python»**
Вариант №2

Выполнила:
Беседина Инга Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и вычислительная техника», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических наук, доцент, доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 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 и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

## Ход работы

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

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

```python
#!/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}")
```

Рисунок 1. Программа вычисления значения функции

```
Value of x? 4
y = 5.0
```

Рисунок 2. Результат выполнения программы

**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.*

```python
#!/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)
```

Рисунок 3. Программа решения поставленной задачи

```
Введите номер месяца: 10
Осень
```

Рисунок 4. Результат выполнения программы

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

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

*где $n$ и $k$ вводятся с клавиатуры.*

```python
#!/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}")
```

Рисунок 5. Программа решения поставленной задачи

```
Value of n? 10
Value of x? 3
S = 2.321096518079936
```

Рисунок 6. Результат выполнения программы

**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:*

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

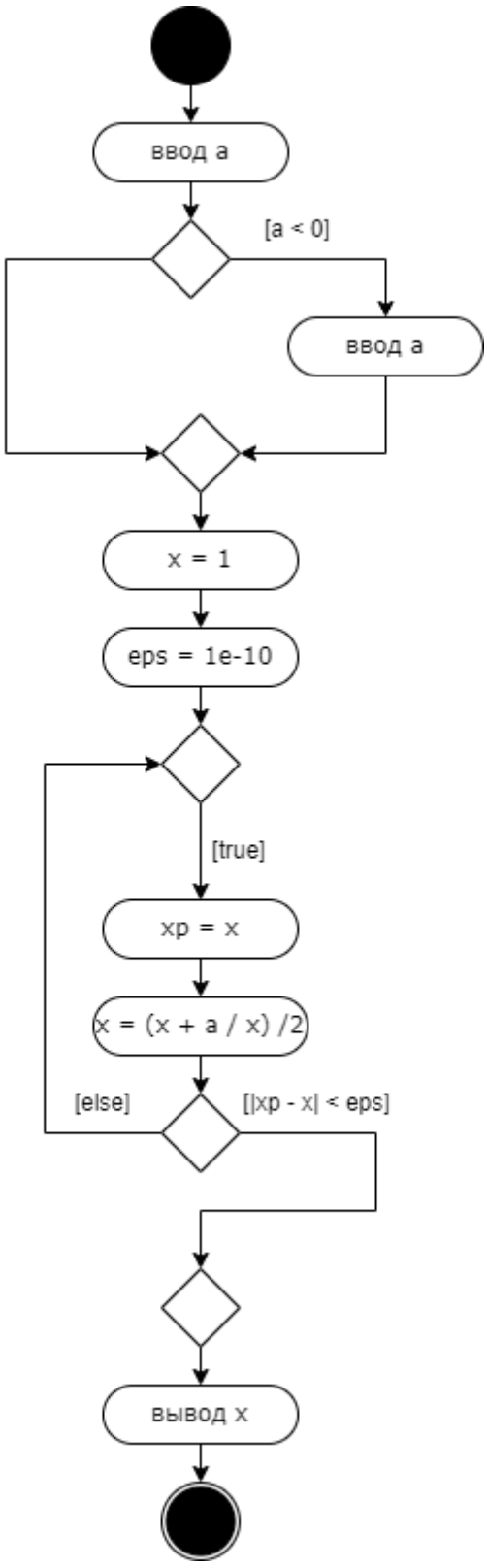



Рисунок 7. UML-диаграмма для примера 4

```python
#!/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)}")
```

Рисунок 8. Программа для решения поставленной задачи

```
cripts/python.exe c:/Rep/Python_LR_2_2/Project/example4.py
Value of a? 36
x = 6.0
X = 6.0
PS C:\Rep\Python_LR_2_2\Project> & c:/Rep/Python_LR_2_2/Project/venv/Scripts/python.exe c:/Rep/Python_LR_2_2/Project/example4.py
Value of a? 5
x = 2.23606797749979
X = 2.23606797749979
PS C:\Rep\Python_LR_2_2\Project>
```

Рисунок 9. Результат выполнения программы

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

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

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

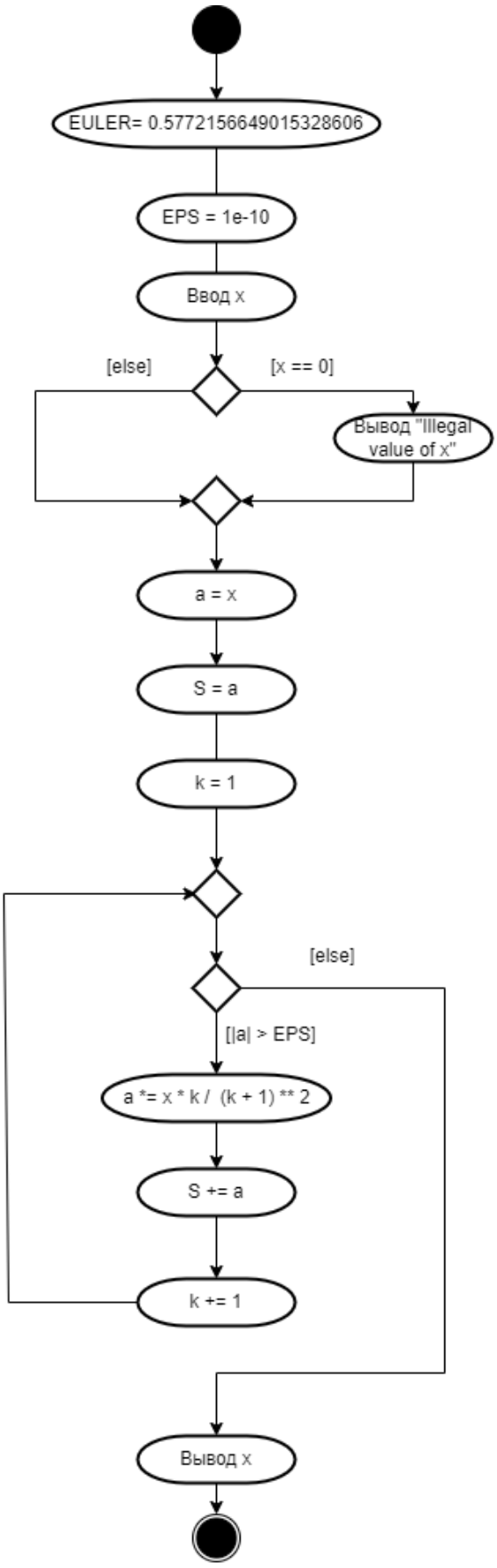



Рисунок 10. UML-диаграмма для примера 5

```python
#!/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}")
```

Рисунок 11. Программа для решения поставленной задачи

```
Value of x? 5
Ei(5.0) = 40.18527535579794
```

Рисунок 12. Результат выполнения программы

2. Дано число $m$ ($1 \leq m \leq 12$).Определить, сколько дней в месяце с номером $m$.

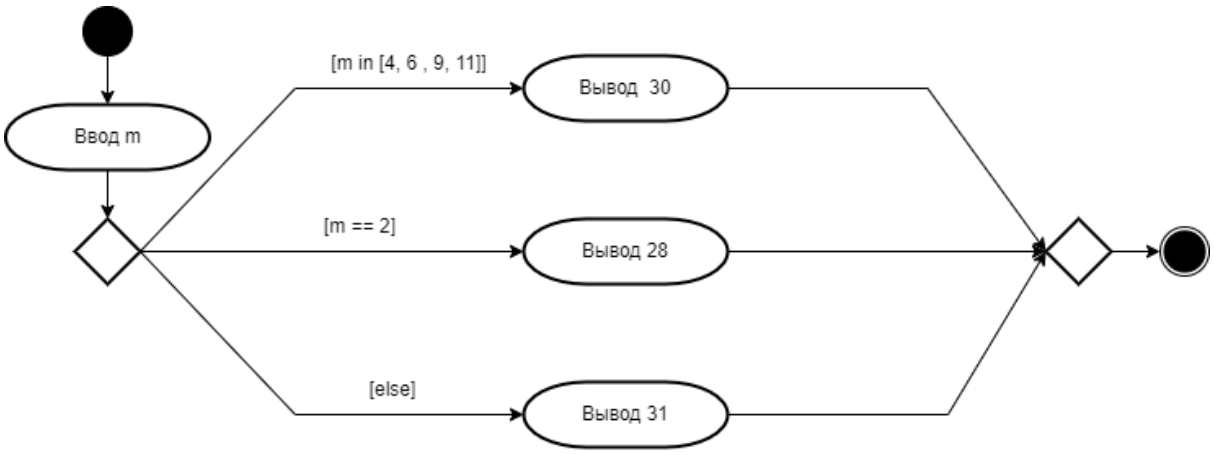



Рисунок 13. UML-диаграмма для индивидуального задания 1

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


if __name__ == '__main__':
    m = int(input("Value of m: "))
    if m in [4, 6, 9, 11]:
        print("30")
    elif m == 2:
        print("28")
    else:
        print("31")
```

Рисунок 14. Программа

```
PS C:\Rep\Python_LR_2_2\Project> & c:/Rep/
Value of m: 8
31
Value of m: 6
30
(venv) PS C:\Rep\Python_LR_2_2\Project> &
11.py
Value of m: 2
28
(venv) PS C:\Rep\Python_LR_2_2\Project>
```

Рисунок 15. Результат выполнения программы

Даны действительные числа $x$ и $y$. Найти $U = \max^2(x^2\,y, x\,y^2) + \min^2(x - y, x + 2y)$. Для минимума и максимума использовать условный оператор `if`.

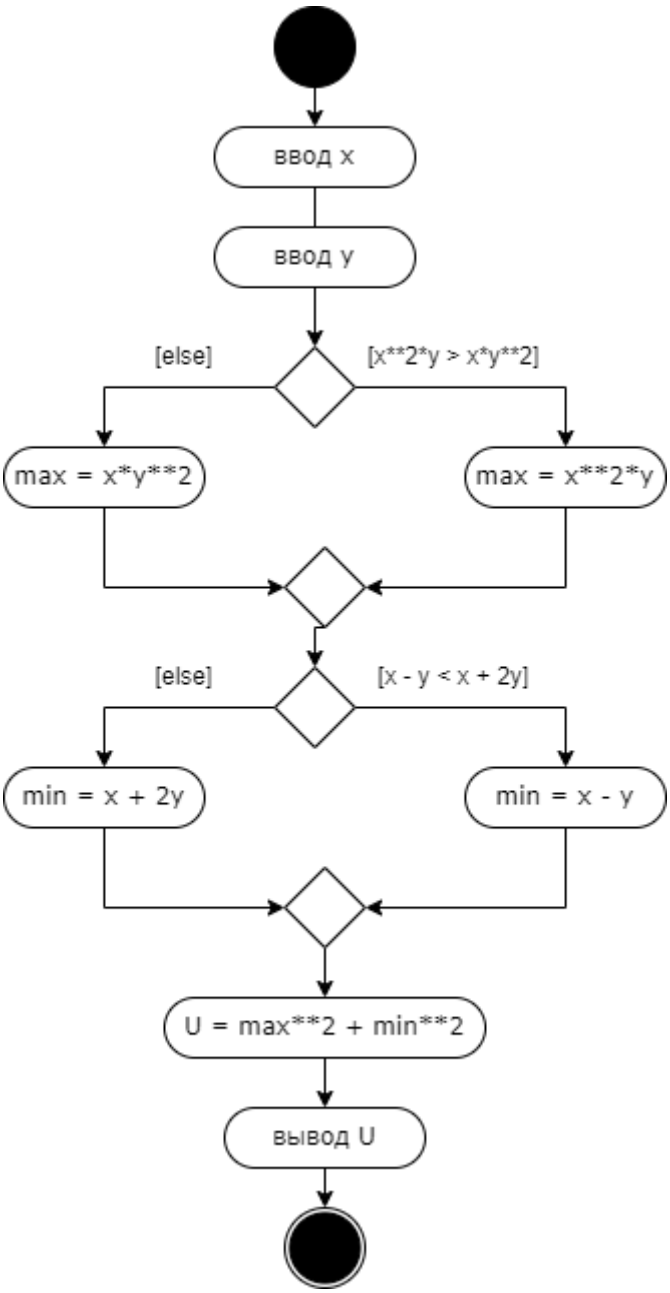
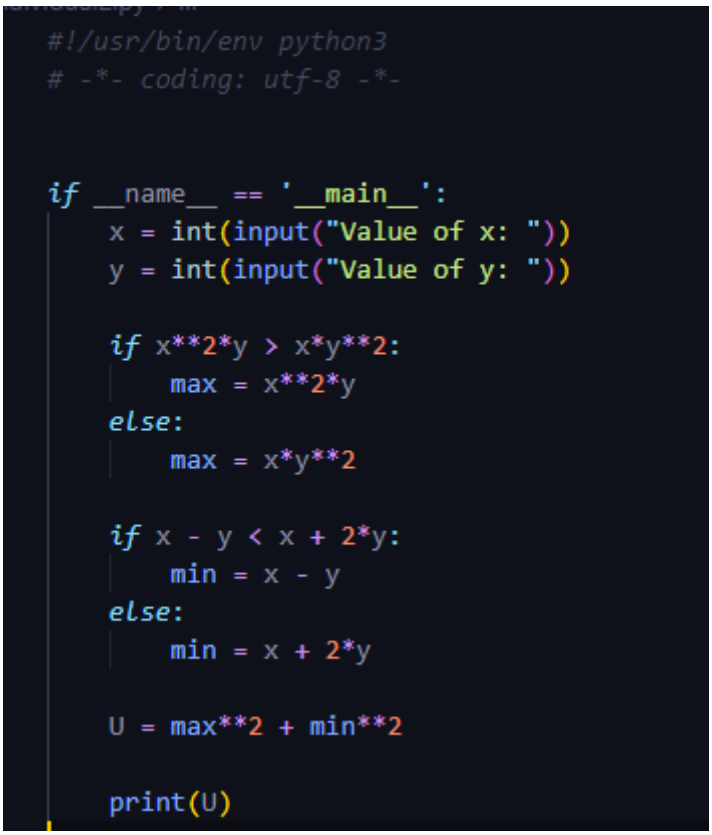



Рисунок 16. UML-диаграмма для индивидуального задания 2



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


if __name__ == '__main__':
    x = int(input("Value of x: "))
    y = int(input("Value of y: "))

    if x**2*y > x*y**2:
        max = x**2*y
    else:
        max = x*y**2

    if x - y < x + 2*y:
        min = x - y
    else:
        min = x + 2*y

    U = max**2 + min**2

    print(U)
```

Рисунок 17. Программа

```
Value of x: 4
Value of y: 3
2305
```

Рисунок 18. Результат выполнения программы

2. Найти сумму целых положительных чисел, больших 20, меньших 100 и кратных 3.

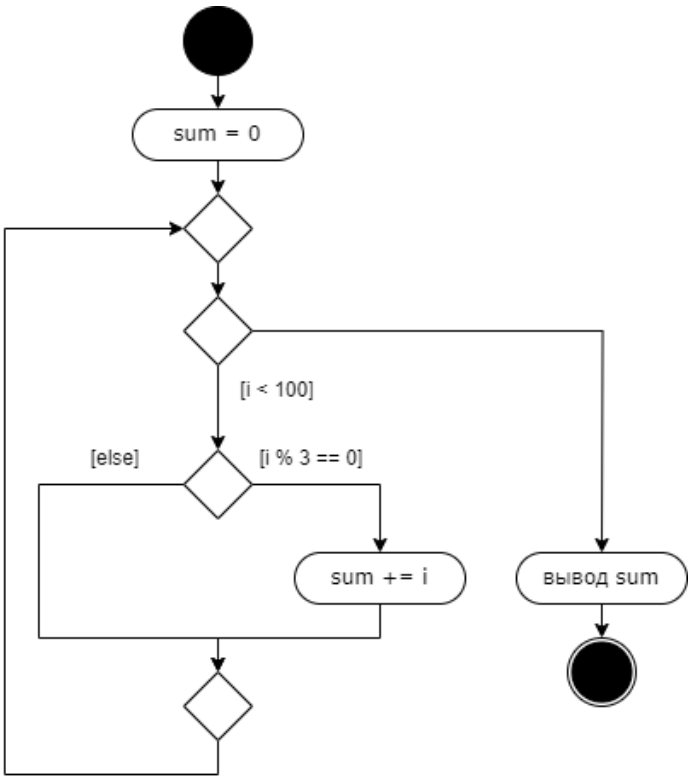



Рисунок 19. UML-диаграмма для индивидуального задания 3

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


if __name__ == '__main__':
    sum = 0

    for i in range(21, 100):
        if i % 3 == 0:
            sum += i

    print(sum)
```

Рисунок 20. Программа

```
(venv) PS C:\Rep\Python_LR_2_2\Project>
l3.py
1620
```

Рисунок 21.  Результат выполнения программы

2. Интегральный косинус:

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

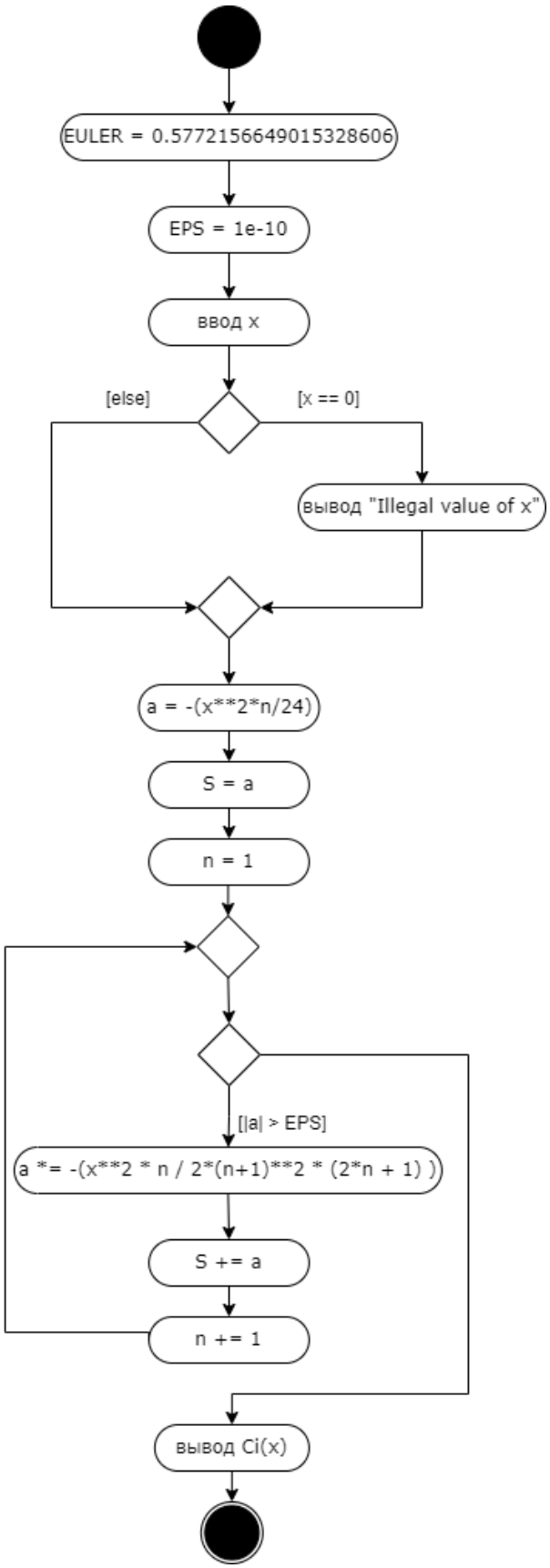



Рисунок 22. UML-диаграмма для индивидуального задания 4

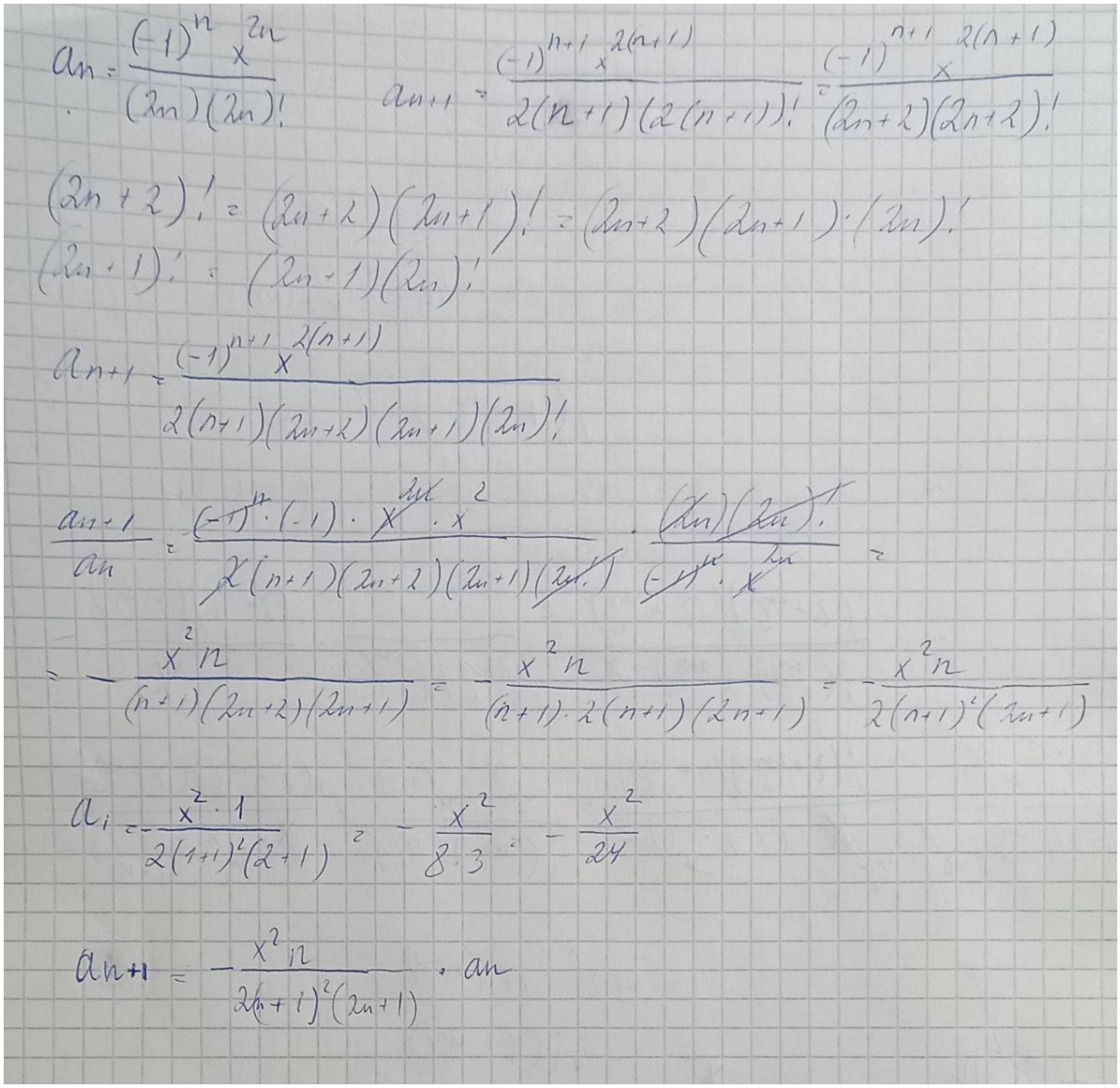



Рисунок 23. Выражения, необходимые для вычисления значения рекуррентного соотношения

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


import math
import sys


EULER = 0.5772156649015328606
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**2/24)
    S = a
    n = 1

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

    print(f"Ci({x}) = {EULER + math.log(math.fabs(x)) + S}")
```

Рисунок 24. Программа для решения поставленной задачи

```
Value of x? 0.01
Ci(0.01) = -4.027958685264474
```

Рисунок 25. Результат выполнения программы

Контрольные вопросы

1. 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

2. Состояние действия в UML диаграммах деятельности представляет собой мгновенное действие, которое выполняется в рамках

процесса. Это может быть любая операция, функция или шаг, который происходит в определенный момент времени. Состояние деятельности в UML диаграммах деятельности представляет собой состояние, в котором процесс находится во время выполнения определенной деятельности. Это может быть любое состояние, в котором процесс находится в процессе выполнения определенной задачи или операции

3. Стрелки для обозначения переходов между состояниями или действиями. Стрелка указывает направление потока управления от одного состояния или действия к другому. Ромбы (или ромбовидные фигуры) для обозначения ветвлений. Ромб указывает на место в диаграмме, где происходит разветвление потока управления на два или более возможных пути выполнения. Условные обозначения (например, условные выражения или логические операторы) для обозначения условий, которые определяют направление переходов или ветвлений

4. 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

5. В программе разветвляющейся структуры имеется один или несколько условных операторов. Для программной реализации условия используется логическое выражение

6. Условный оператор - это конструкция в программировании, которая позволяет выполнять определенные действия в зависимости от выполнения определенного условия. В зависимости от того, истинно ли условие, выполняется определенный блок кода. Существуют различные формы условных операторов: Оператор if-else: позволяет выполнить определенный блок кода, если условие истинно, и другой блок кода, если условие ложно. Оператор switch: позволяет выбрать один из нескольких вариантов выполнения кода, в зависимости от значения выражения. Тернарный оператор: это оператор, который позволяет сократить запись

условного выражения до одной строки кода, включая условие, значение при истинном условии и значение при ложном условии

7. Операторы сравнения: > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), != (не равно), == (равно)

8. Простым условием называется выражение, которое использует один из операторов сравнения для сравнения двух значений и возвращает булево значение. x == 5 (проверяет, равно ли значение переменной x числу 5); y > 10 (проверяет, больше ли значение переменной y числа 10)

9. Составное условие - это выражение, которое использует логические операторы (и, или, не) для объединения нескольких простых условий. (x > 5) and (y < 10) (проверяет, больше ли значение переменной x числа 5 и меньше ли значение переменной y числа 10); (z == 100) or (w != 0) (проверяет, равно ли значение переменной z числу 100 или не равно ли значение переменной w нулю)

10. and (логическое и), or (логическое или), not (логическое не)

11. Да, оператор ветвления в Python (if, elif, else) может содержать внутри себя другие ветвления. Например, внутри блока if можно использовать другой блок if или блок elif для дополнительной проверки условий

12. Алгоритм циклической структуры - это алгоритм, который содержит циклы или повторяющиеся операции. Например, цикл for или цикл while являются примерами циклических структур в программировании. Эти циклы позволяют выполнять определенные действия несколько раз, пока выполняется определенное условие

13. Цикл for используется для выполнения определенного блока кода определенное количество раз. Он может быть использован для перебора элементов в последовательности, таких как списки, кортежи, строки и др. Цикл while выполняет блок кода, пока указанное условие истинно. Он может быть использован, когда количество итераций заранее неизвестно

14. Функция range используется для создания последовательности чисел в определенном диапазоне. Она может принимать один, два или три

аргумента. Если указан только один аргумент, то range(n) создаст последовательность чисел от 0 до n-1. Если указаны два аргумента, то range(start, stop) создаст последовательность чисел от start до stop-1. Если указаны три аргумента, то range(start, stop, step) создаст последовательность чисел от start до stop-1 с шагом step

15. range(15, -1, -2)

16. Циклы могут быть вложенными

17. Бесконечный цикл образуется, когда условие выхода из цикла не выполняется, и цикл продолжает выполняться бесконечно. Для того чтобы выйти из бесконечного цикла, можно использовать команду прерывания цикла, такую как break

18. Оператор break используется для прерывания выполнения цикла при выполнении определенного условия. Это позволяет избежать бесконечного выполнения цикла и перейти к выполнению следующих инструкций в программе

19. Оператор continue используется внутри цикла для пропуска текущей итерации и перехода к следующей итерации. Это позволяет пропустить выполнение определенной части кода внутри цикла и перейти к следующей итерации, если выполняется определенное условие. Оператор continue обычно используется, когда нужно пропустить выполнение определенной части кода внутри цикла, но продолжить выполнение цикла с следующей итерации

20. Стандартные потоки stdout и stderr в Python используются для вывода информации и ошибок соответственно. Stdout (стандартный вывод) используется для вывода обычных сообщений, результатов вычислений и любой другой информации, которую вы хотите видеть в консоли или записать в файл. Stderr (стандартный поток ошибок) используется для вывода сообщений об ошибках, предупреждений и другой информации об ошибках, которая не должна попадать в стандартный вывод

21. Для вывода информации в стандартный поток ошибок (stderr) в Python можно использовать модуль sys:

```
import sys
sys.stderr.write("Это сообщение будет выведено в стандартный поток ошибок\n")
```

22. Функция exit() используется для завершения выполнения программы. При вызове этой функции программа завершается, и управление возвращается операционной системе. Можно передать опциональный аргумент, который будет использоваться в качестве кода завершения программы. Например, вызов exit(1) указывает на то, что программа завершилась с ошибкой, а вызов exit(0) указывает на успешное завершение программы

**Вывод:** в ходе выполнения лабораторной работы были приобретены навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры, освоены операторы языка Python версии 3.x if, while, for, break и continue, позволяющие реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.