

**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
Коновалова В.Н. « » 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.

Ход работы:

Учебные задачи:

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

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

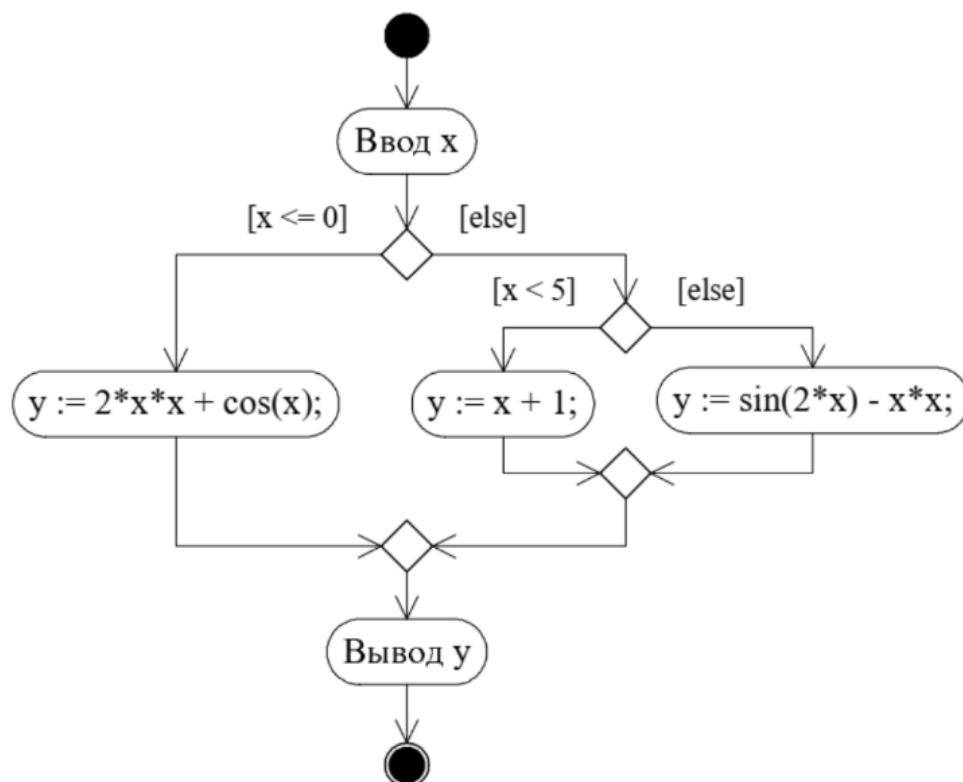


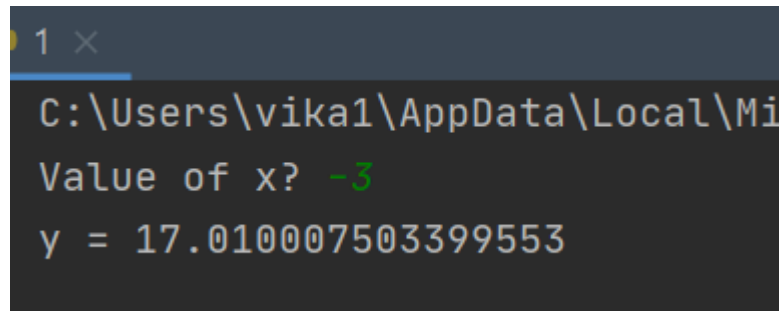
Рисунок 1 – UML-диаграмма деятельности

Код:

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

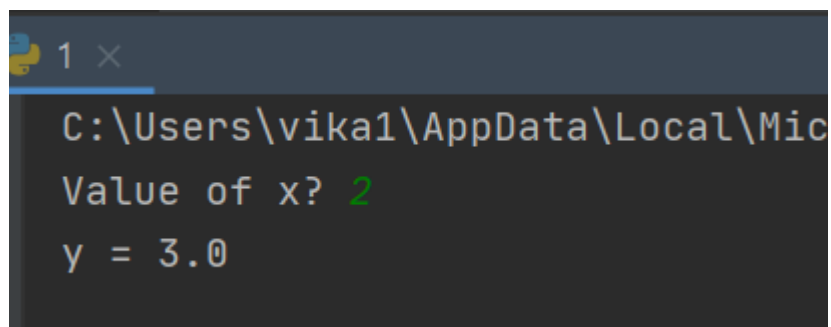
if __name__ == '__main__':
```

```
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}")
```



A screenshot of a Python terminal window. The title bar shows a file icon, the number '1', and a close button. The terminal path is 'C:\Users\vika1\AppData\Local\Mi'. The prompt 'Value of x?' is followed by the input '-3'. The output is 'y = 17.010007503399553'.

Рисунок 2 – Результат выполнения программы при $x = -3$



A screenshot of a Python terminal window. The title bar shows a file icon, the number '1', and a close button. The terminal path is 'C:\Users\vika1\AppData\Local\Mi'. The prompt 'Value of x?' is followed by the input '2'. The output is 'y = 3.0'.

Рисунок 3 – Результат выполнения программы при $x = 2$

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.

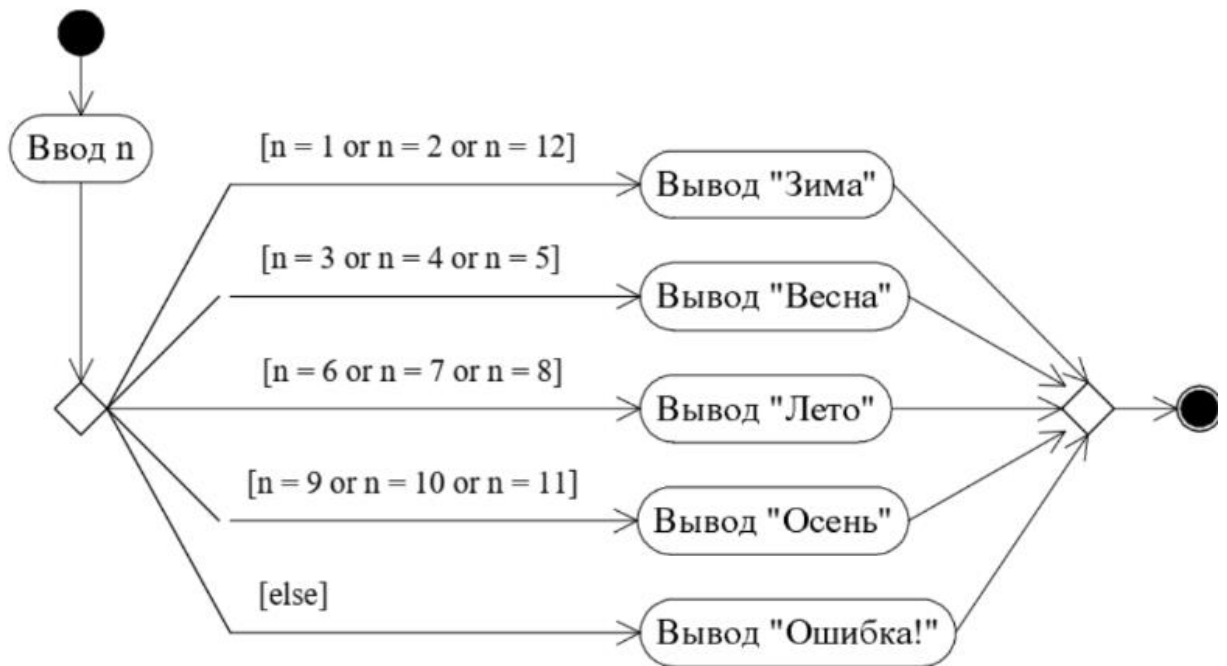


Рисунок 4 – UML-диаграмма деятельности

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)
  
```

```
2 x
C:\Users\vika1\AppData\Local\Microsoft
Введите номер месяца: 3
Весна
```

Рисунок 5 – Результат выполнения программы при $x = 3$

```
2 x
C:\Users\vika1\AppData\Local\Microsoft\Windows
Введите номер месяца: 13
Ошибка!
```

Рисунок 6 – Результат выполнения программы при $x = 13$

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

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

где n и k вводятся с клавиатуры.

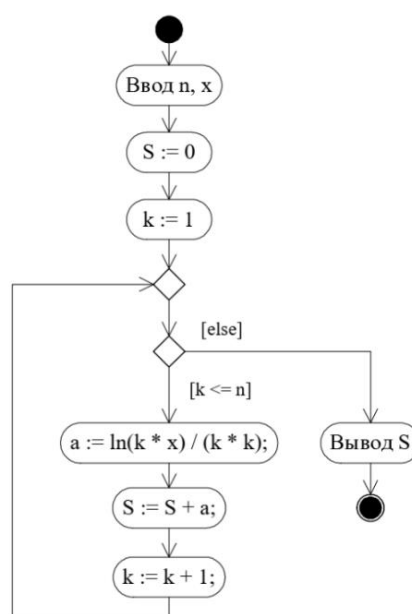


Рисунок 7 – UML-диаграмма деятельности

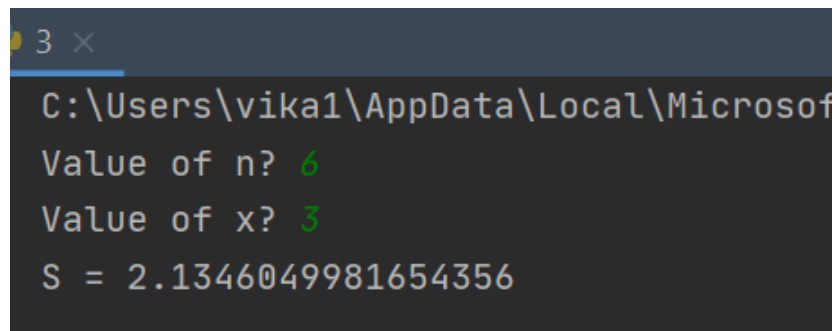
Код:

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



```
3 x
C:\Users\vika1\AppData\Local\Microsoft
Value of n? 6
Value of x? 3
S = 2.1346049981654356
```

Рисунок 8 – Результат выполнения программы при $n = 6$, $x = 3$

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

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

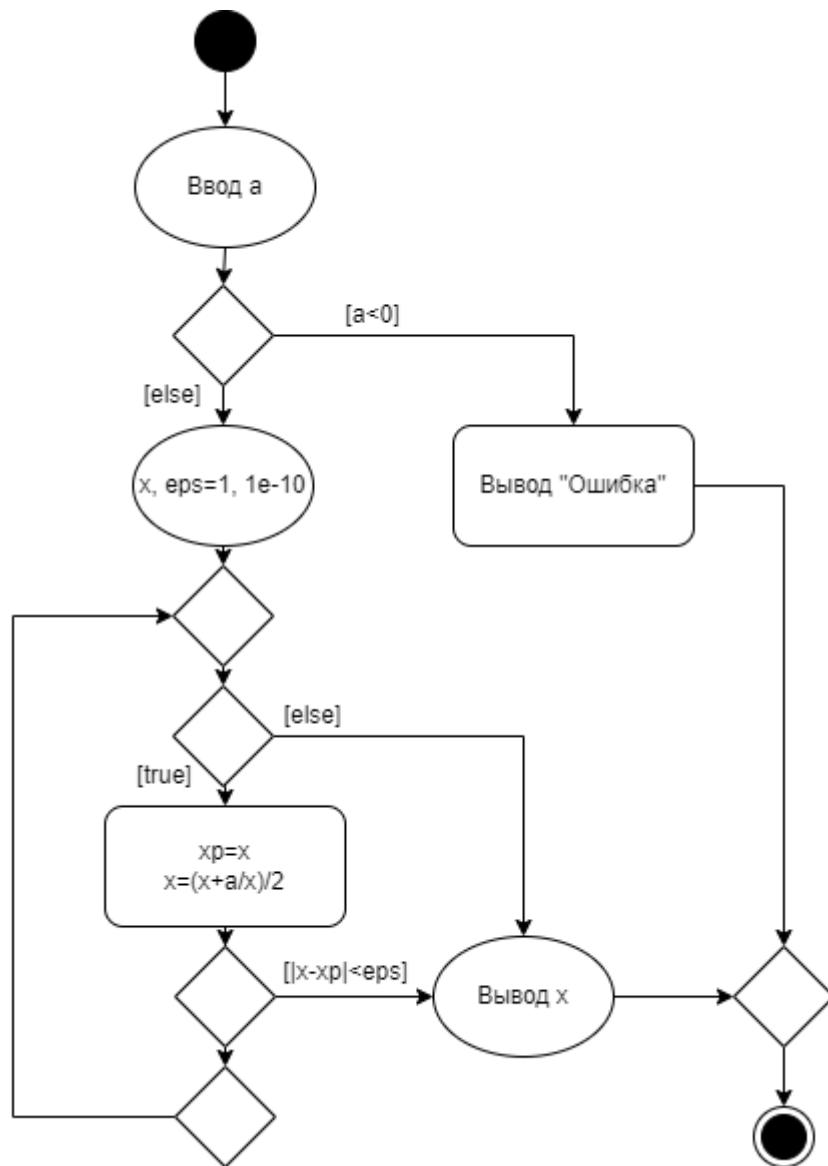


Рисунок 9– UML-диаграмма деятельности

Код:

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

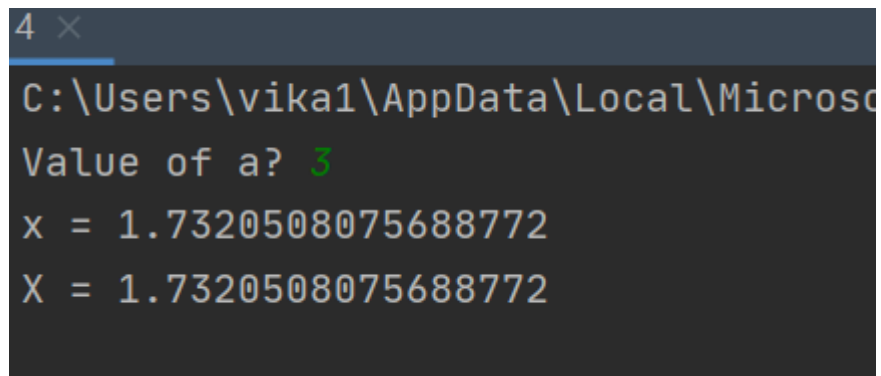
if __name__ == '__main__':
    a = float(input("Value of a? "))
    if a < 0:
        print("Illegal value of a", file=sys.stderr)
        exit(1)
    x, eps = 1, 1e-10
    while True:
        xp = x
        x = (x + a / x) / 2
```

```

if math.fabs(x - xp) < eps:
    break

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

```



```

4 x
C:\Users\vika1\AppData\Local\Microsoft
Value of a? 3
x = 1.7320508075688772
X = 1.7320508075688772

```

Рисунок 10 – Результат выполнения программы при $a = 3$

Пример 5. Вычислить значение специальной (интегральной показательной) функции

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

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

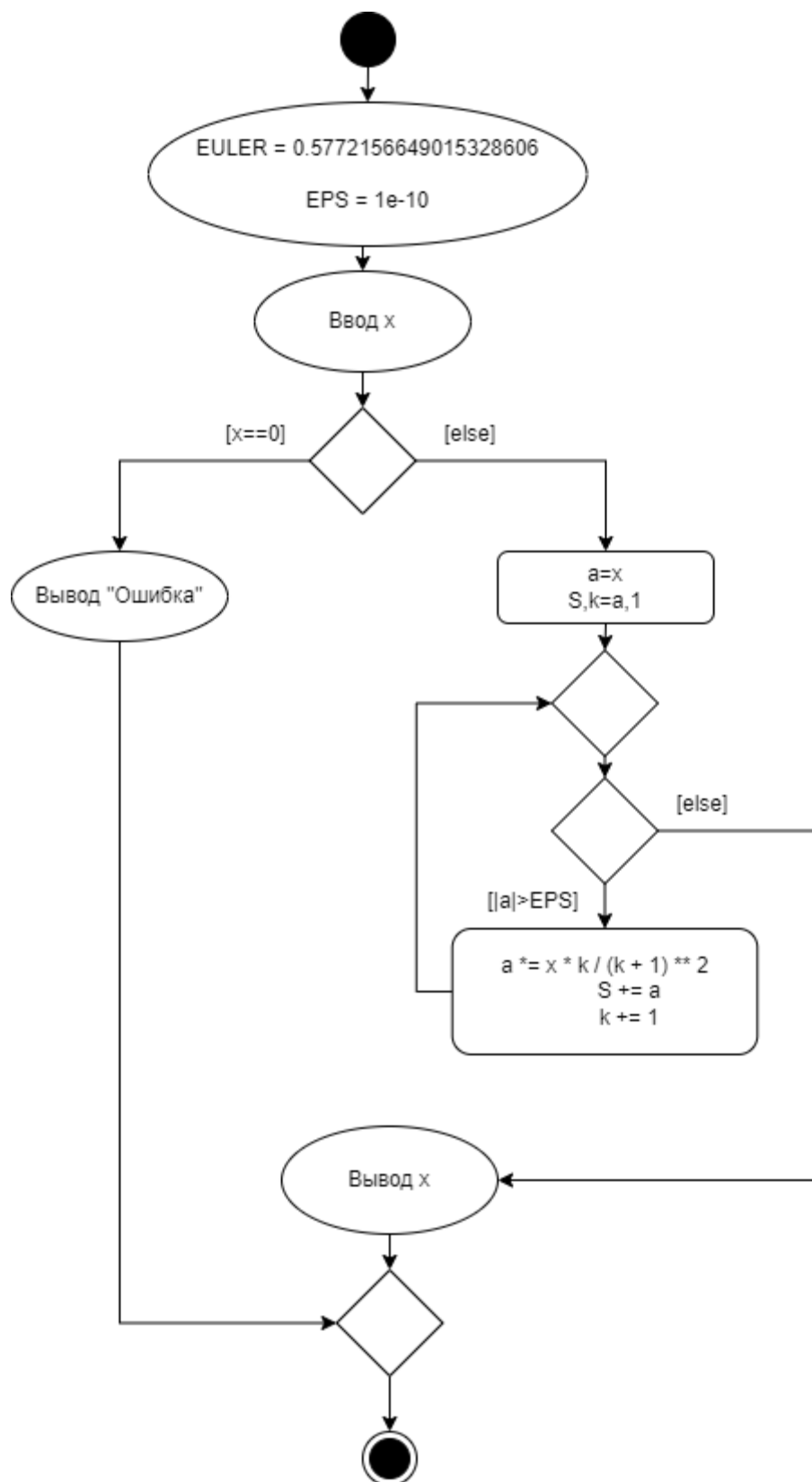


Рисунок 11 – UML-диаграмма деятельности

Код:

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

import math
```

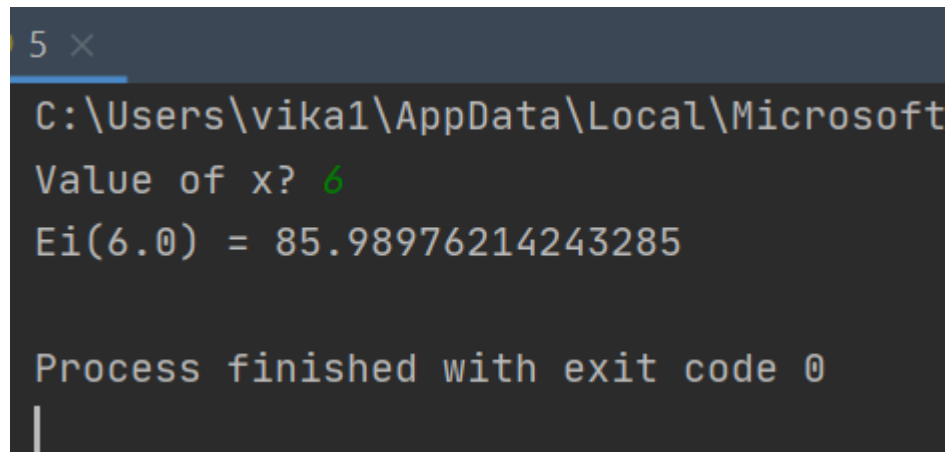
```

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}")

```



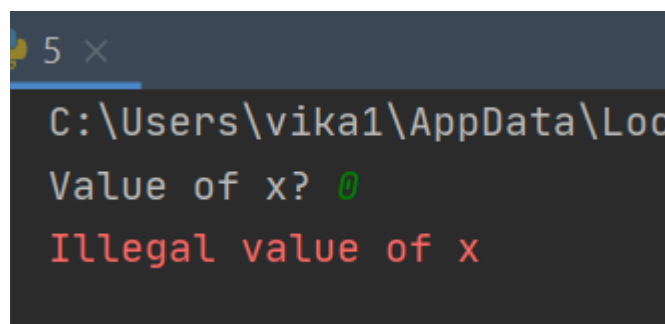
```

5 x
C:\Users\vika1\AppData\Local\Microsoft
Value of x? 6
Ei(6.0) = 85.98976214243285

Process finished with exit code 0

```

Рисунок 12 – Результат выполнения программы при $x = 6$



```

5 x
C:\Users\vika1\AppData\Loc
Value of x? 0
Illegal value of x

```

Рисунок 13 – Результат выполнения программы при $x = 0$

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

Задание 1.

13. Студенты убирают урожай помидоров. При сборе до 50 кг в день работа оплачивается из расчёта 30 коп. за 1 кг; при сборе от 50 до 75 кг в день - 50 коп. за 1 кг; при сборе от 75 до 90 кг в день - 65 коп. за 1 кг; при сборе свыше 90 кг в день - 70 коп. за 1 кг плюс 20 руб. премия. Студент собрал X кг за день. Определить его заработок.

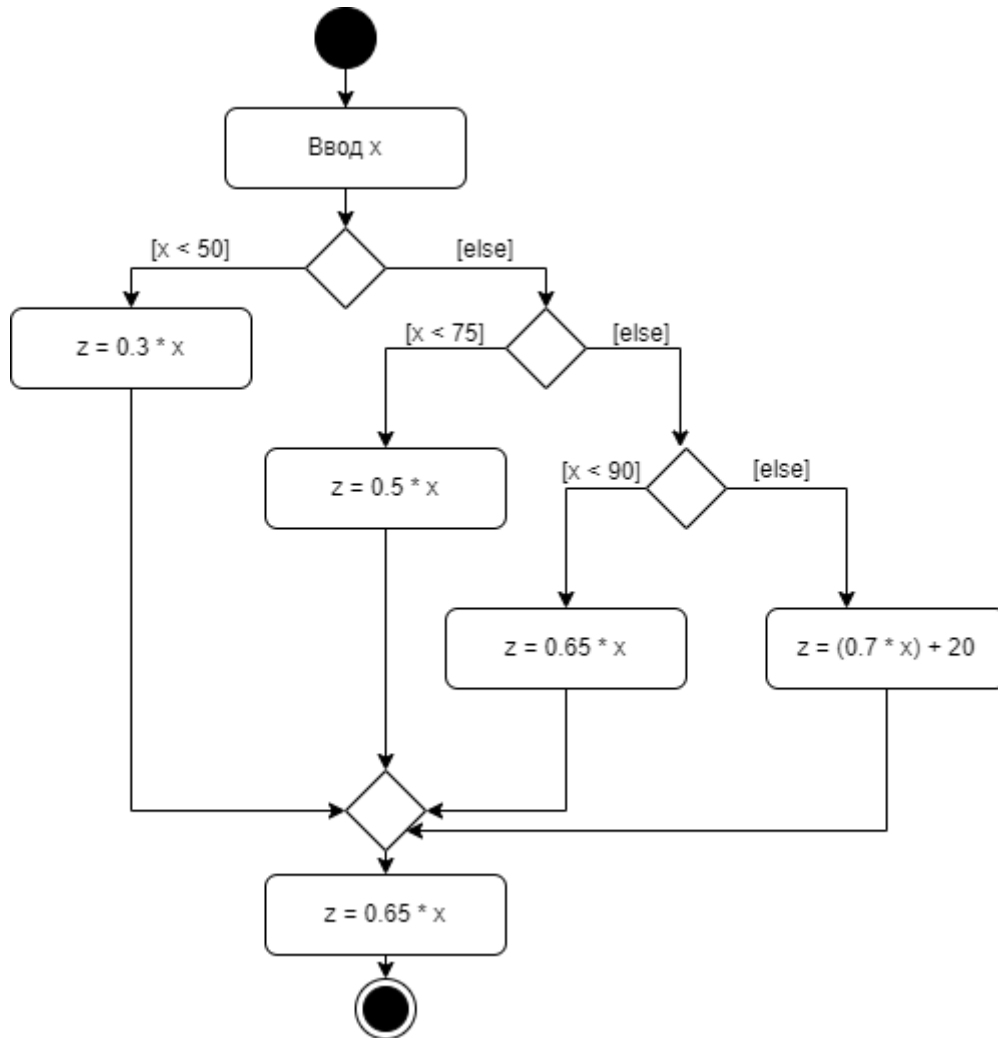


Рисунок 14 – UML-диаграмма деятельности

Код:

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

if __name__ == '__main__':
    x = float(input("Value of x? "))
    if x < 50:
        z = 0.3 * x
    elif x < 75:
        z = 0.5 * x
    elif x < 90:
        z = 0.65 * x
    else:
        z = (0.7 * x) + 20
    print("Salary: ", z)
```

```
1id x
C:\Users\vika1\AppData\Local\Microsoft
Value of x? 75
Salary: 48.75

Process finished with exit code 0
```

Рисунок 15– Результат выполнения программы при $x = 75$

```
1id x
C:\Users\vika1\AppData\Local\Micros
Value of x? 100
Salary: 90.0

Process finished with exit code 0
```

Рисунок 16 – Результат выполнения программы при $x = 100$

Задание 2.

13. Составить программу, выясняющую делится ли натуральное число x нацело на натуральное число y .

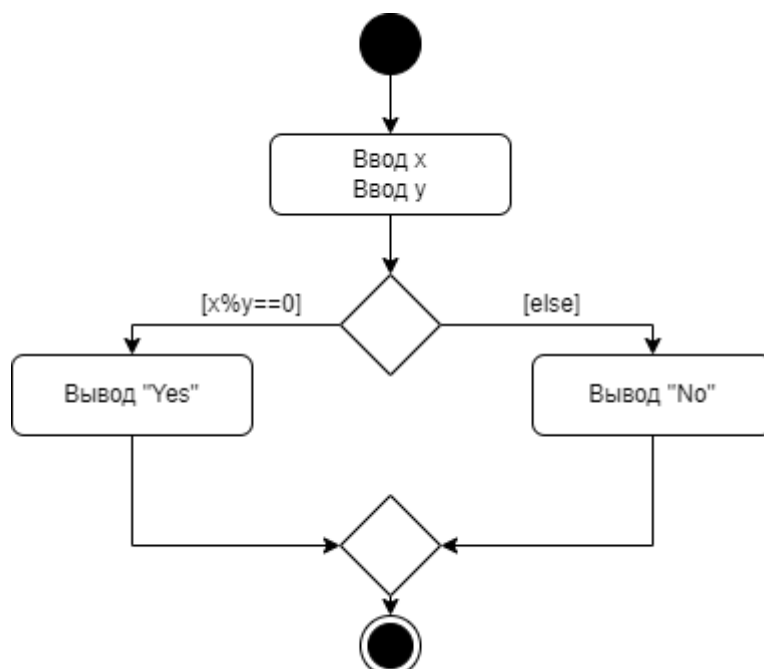
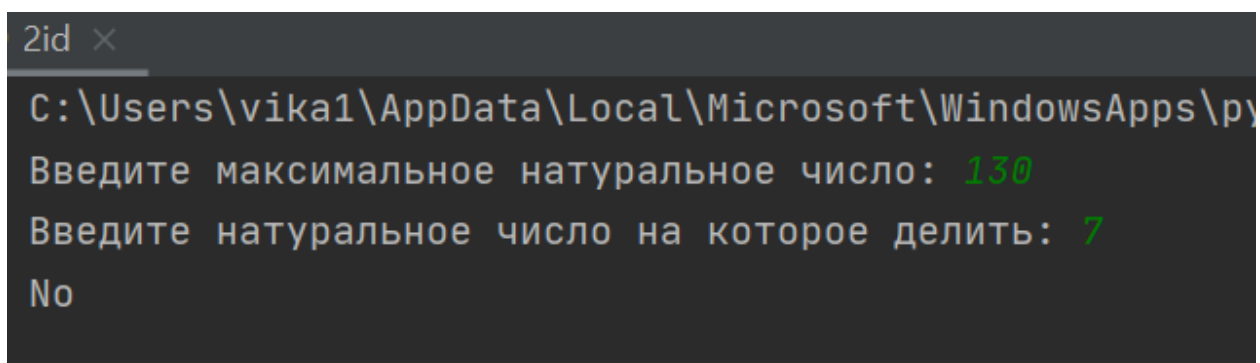


Рисунок 17 – UML-диаграмма деятельности

Код:

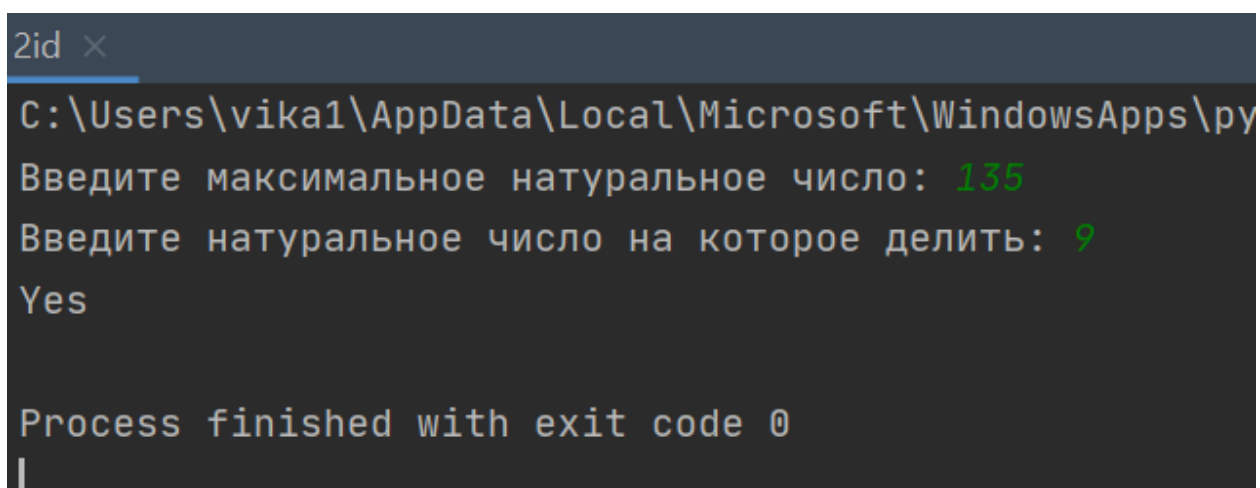
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    x = int(input("Введите максимальное натуральное число: "))
    y = int(input("Введите натуральное число на которое делить: "))
    if x % y == 0:
        print("Yes")
    else:
        print("No")
```



```
2id x
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\py
Введите максимальное натуральное число: 130
Введите натуральное число на которое делить: 7
No
```

Рисунок 18– Результат выполнения программы



```
2id x
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\py
Введите максимальное натуральное число: 135
Введите натуральное число на которое делить: 9
Yes

Process finished with exit code 0
```

Рисунок 19– Результат выполнения программы

Задание 3.

13. Дано натуральное число n . Получить все его натуральные делители.

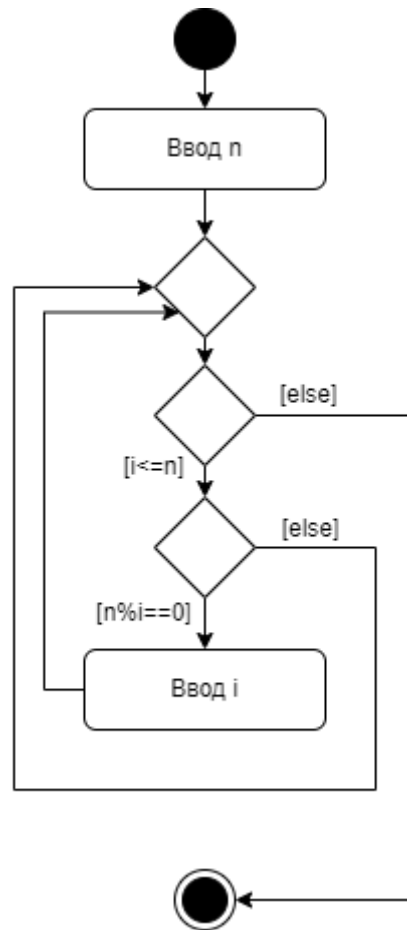


Рисунок 20 – UML-диаграмма деятельности

Код:

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

if __name__ == '__main__':
    n = int(input("Введите натуральное число: "))
    for i in range(1, n + 1):
        if n % i == 0:
            print(i)
```

```

3id x
C:\Users\vika1\AppData\Local\Microsoft\WindowsApp
Введите натуральное число: 138
1
2
3
6
23
46
69
138

Process finished with exit code 0
  
```

Рисунок 21– Результат выполнения программы при n = 138

Задание повышенной сложности

Составить UML-диаграмму деятельности, программу и произвести вычисления вычисление значения специальной функции по ее разложению в ряд с точностью $\varepsilon = 10^{-10}$, аргумент функции x вводится с клавиатуры. Номер варианта необходимо получить у преподавателя.

3. Интегральный гиперболический синус:

$$\text{Shi}(x) = \int_0^x \frac{\text{sh } t}{t} dt = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)(2n+1)!}.$$

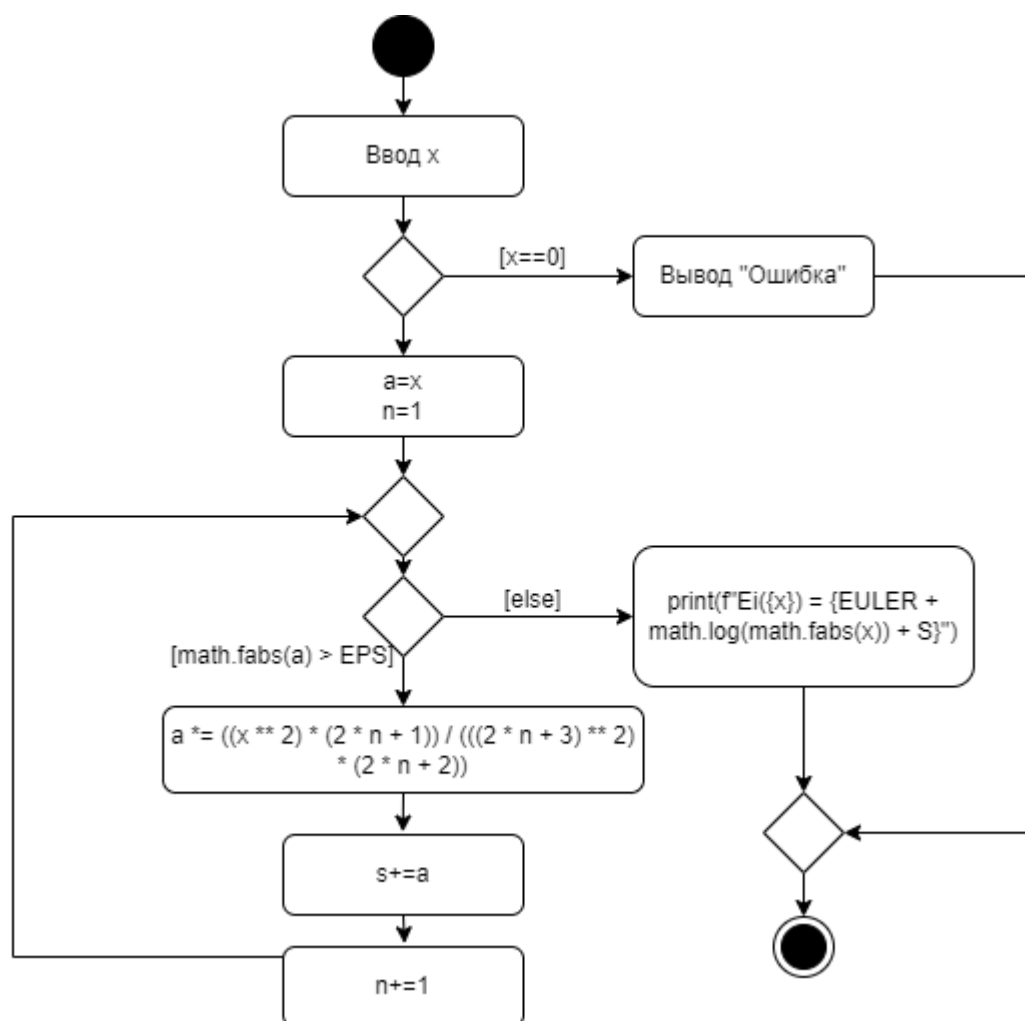


Рисунок 22 – UML-диаграмма деятельности

Код:

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

```

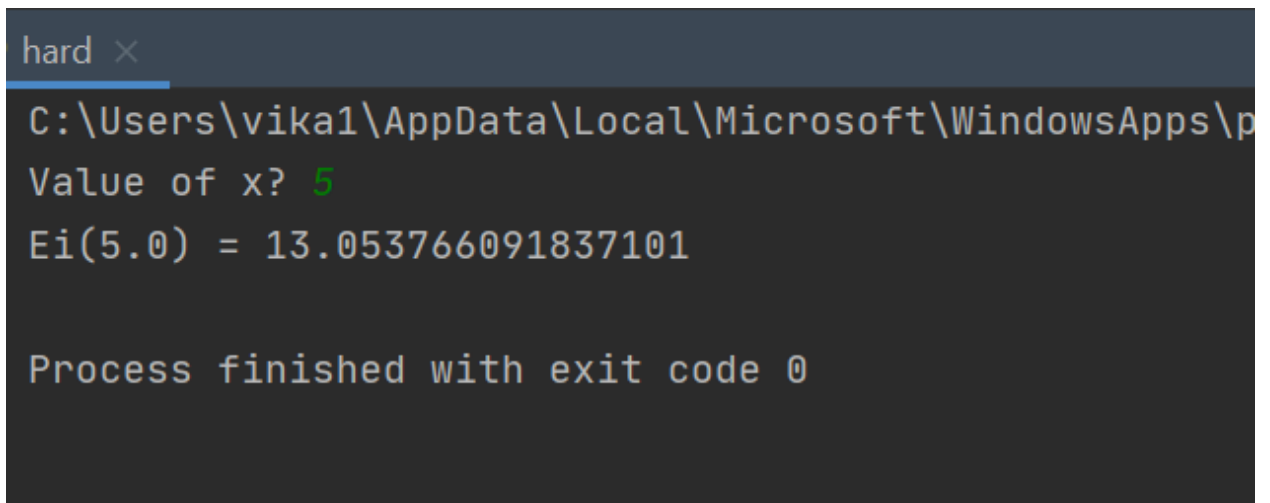
import sys
import math

# 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
    while math.fabs(a) > EPS:
        a *= ((x ** 2) * (2 * n + 1)) / (((2 * n + 3) ** 2) * (2 * n + 2))
        S += a
        n += 1

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

```



```

hard x
C:\Users\vika1\AppData\Local\Microsoft\WindowsApps\p
Value of x? 5
Ei(5.0) = 13.053766091837101

Process finished with exit code 0

```

Рисунок 23– Результат выполнения программы при $x = 5$

Вопросы для защиты работы:

1. Для чего нужны диаграммы деятельности UML?

Диаграммы деятельности представляют собой графическое представление рабочих процессов поэтапных действий и действий с поддержкой выбора, итерации и параллелизма. Они описывают поток управления целевой системой, такой как исследование сложных бизнес-правил и операций, а также описание прецедентов и бизнес-процессов. В

UML диаграммы деятельности предназначены для моделирования как вычислительных, так и организационных процессов.

2. Что такое состояние действия и состояние деятельности?

Состояние действия — это состояние, внутренняя активность которого является действием.

Состояние деятельности — это состояние, внутренняя активность которого является деятельностью.

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?

Алгоритм 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?

Линейный алгоритм - алгоритм, все этапы которого выполняются однократно и строго последовательно.

Разветвляющийся алгоритм - алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из нескольких возможных шагов.

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

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

7. Какие операторы сравнения используются в Python?

If, elif, else

8. Что называется простым условием? Приведите примеры.

Простым условием называется выражение, составленное из двух арифметических выражений или двух текстовых величин.

Пример: `x == y`

9. Что такое составное условие? Приведите примеры.

Составное условие – логическое выражение, содержащее несколько простых условий объединенных логическими операциями. Это операции `not`, `and`, `or`.

Пример: `(a == b or c == d)`

10. Какие логические операторы допускаются при составлении сложных условий?

`not`, `and`, `or`.

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

Может.

12. Какой алгоритм является алгоритмом циклической структуры?

Циклический алгоритм — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными.

13. Типы циклов в языке Python.

В Python есть 2 типа циклов: - цикл `while`, - цикл `for`.

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

Функция `range` генерирует серию целых чисел, от значения `start` до `stop`, указанного пользователем. Мы можем использовать его для цикла `for` и обходить весь диапазон как список

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

Параметры функции:

`start` - с какого числа начинается последовательность.

По умолчанию-0

`stop` - до какого числа продолжается последовательность чисел. Указанное число не включается в диапазон

`step` - с каким шагом растут числа. По умолчанию 1

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?

Могут.

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

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

18. Для чего нужен оператор `break`?

Оператор `break` предназначен для досрочного прерывания работы цикла.

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

Оператор `continue` используется только в циклах. В операторах `for`, `while` оператор `continue` запускает цикл заново, при этом код, расположенный после данного оператора, не выполняется.

20. Для чего нужны стандартные потоки `stdout` и `stderr`?

Ввод и вывод распределяется между тремя стандартными потоками: `stdin` — стандартный ввод (клавиатура), `stdout` — стандартный вывод (экран), `stderr` — стандартная ошибка (вывод ошибок на экран)

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

Указать в `print(..., file=sys.stderr)`.

22. Каково назначение функции `exit`?

Функция `exit()` модуля `sys` - выход из Python.