

**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

Халимендик Я. Д. « » 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

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

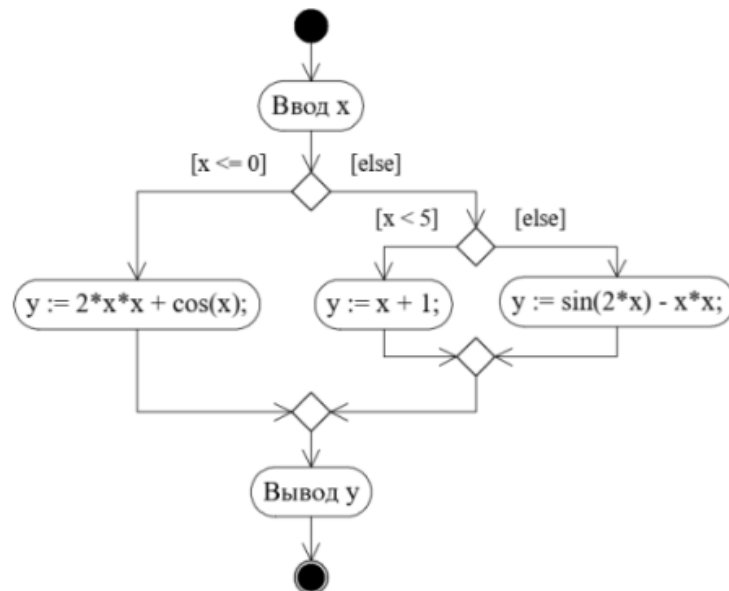


Рисунок 1 – UML-диаграмма деятельности

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}")
```

```
Value of x? -1  
y = 2.5403023058681398
```

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

```
Value of x? 3  
y = 4.0
```

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

```
Value of x? 6  
y = -36.27941549819893
```

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.

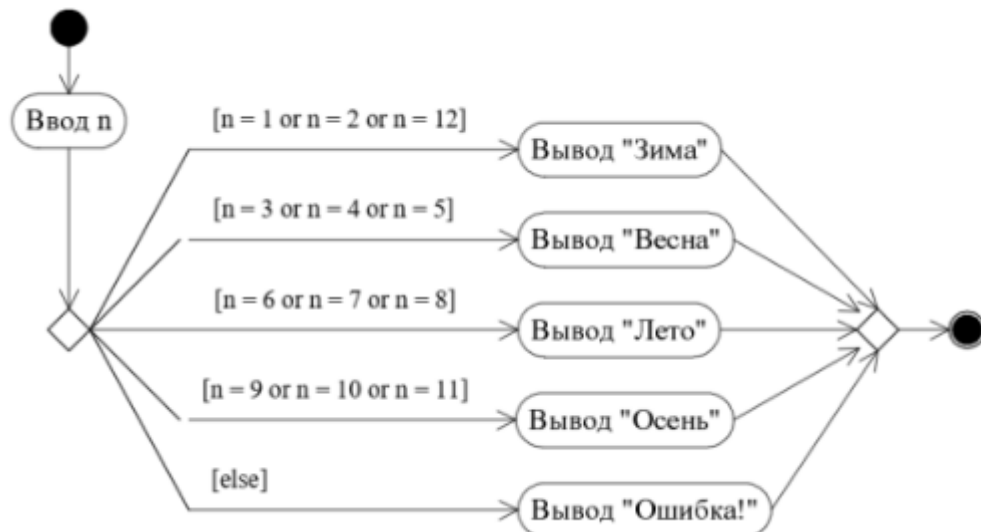


Рисунок 5 – UML-диаграмма деятельности

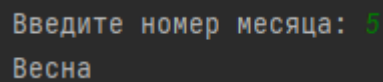
Код:

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

import sys

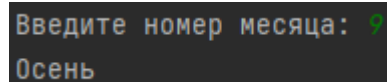
if __name__ == '__main__':
    n = int(input("Введите номер месяца: "))

    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)
```



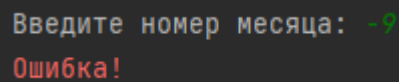
Введите номер месяца: 5
Весна

Рисунок 6 – Результат выполнения программы при $x = 5$



Введите номер месяца: 9
Осень

Рисунок 7 – Результат выполнения программы при $x = 9$



Введите номер месяца: -9
Ошибка!

Рисунок 8 – Результат выполнения программы при $x = -9$

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.

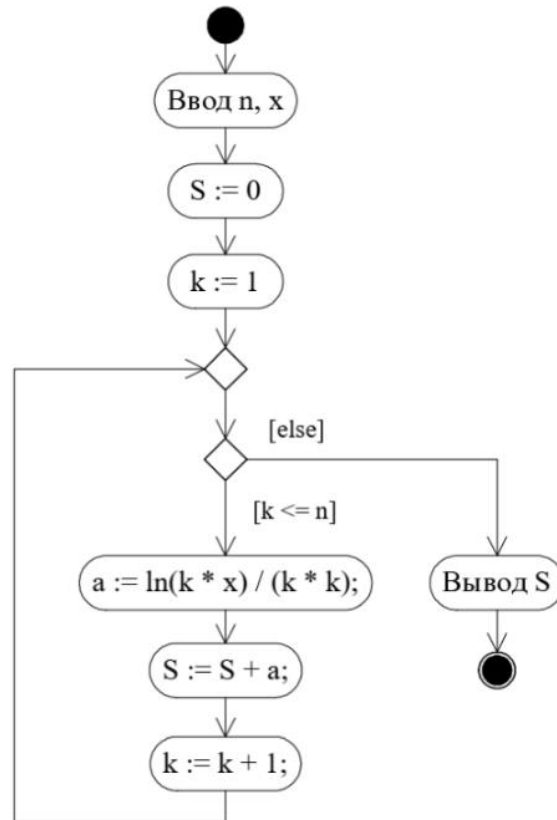


Рисунок 9 – UML-диаграмма деятельности

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? 4
Value of x? 7
S = 3.152217534203236
```

Рисунок 10 – Результат выполнения программы при n = 4, 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 ϵ с помощью рекуррентного соотношения:

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

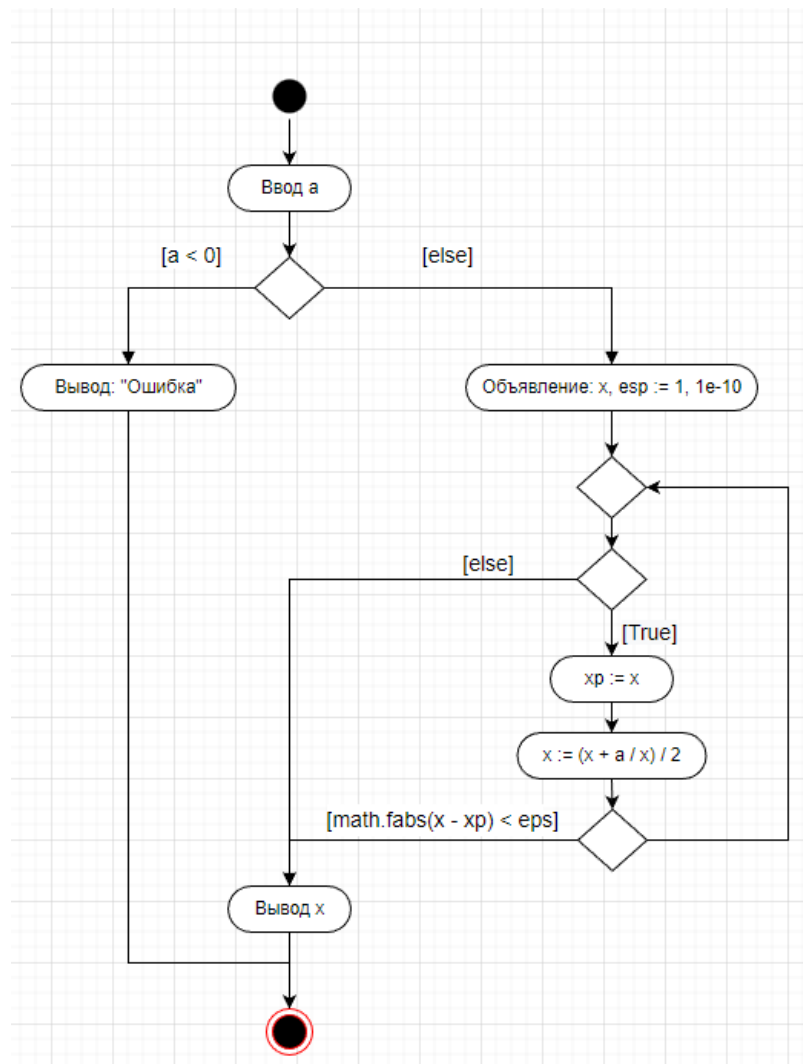


Рисунок 12 – UML-диаграмма деятельности

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? 5
x = 2.23606797749979
X = 2.23606797749979
```

Рисунок 14 – Результат выполнения программы при $a = 5$

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 $\varepsilon = 10^{-10}$, аргумент x вводится с клавиатуры.

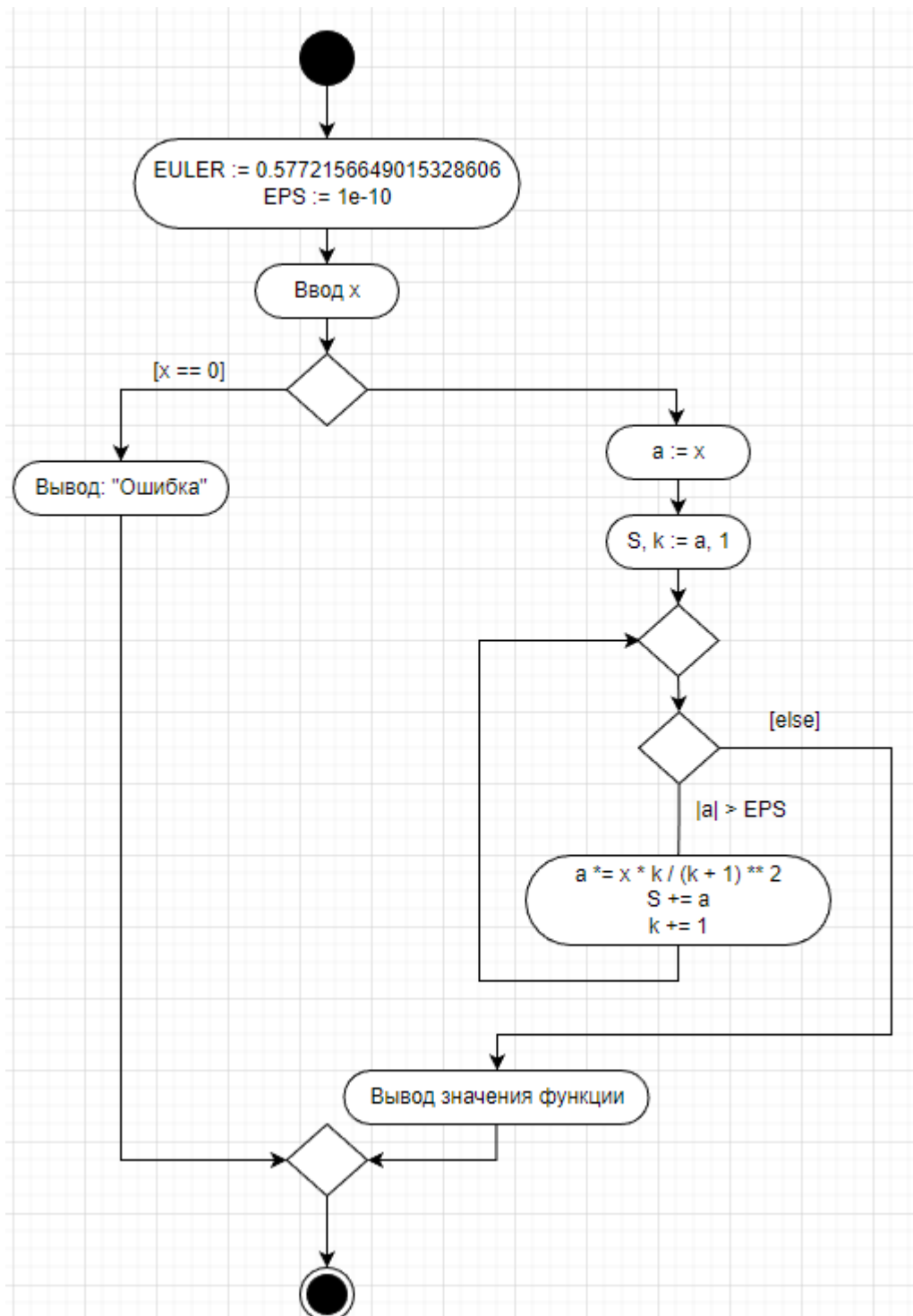


Рисунок 15 – UML-диаграмма деятельности

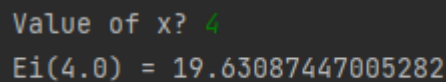
Код:

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

import math
import sys

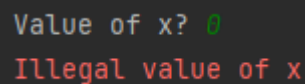
# Постоянная Эйлера.
EULER = 0.5772156649015328606
# Точность вычислений.
EPS = 1e-10

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? 0
Illegal value of x

Рисунок 17 – 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.

Вариант 6(32). Дано целое число c такое, что $|c| < 9$. Вывести это число в словесной форме, учитывая его знак.

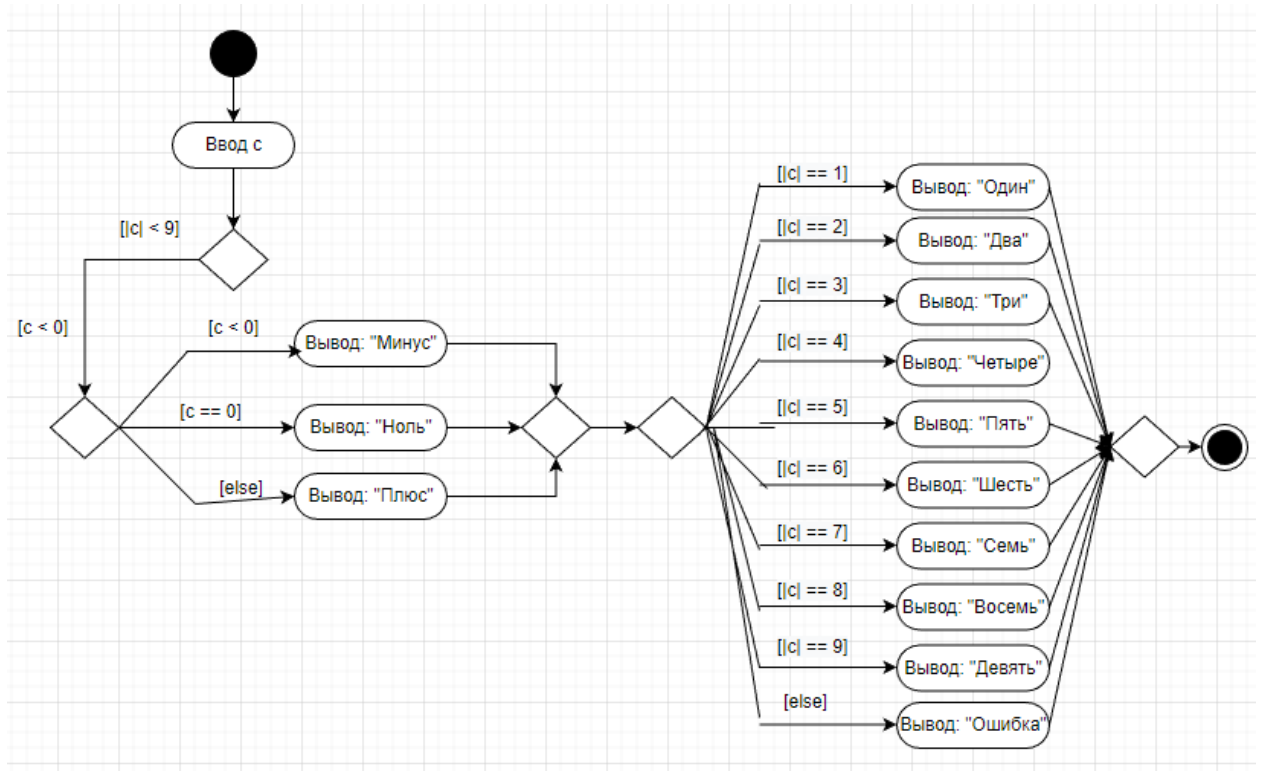


Рисунок 18 – UML-диаграмма деятельности

Код:

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

if __name__ == '__main__':
    c = float(input("Value of c? "))
    if math.fabs(c) < 9:
        if c < 0:
            print("Минус")
        elif c == 0:
            print("Ноль")
        else:
            print("Плюс")
    if math.fabs(c) == 1:
        print("один")
    elif math.fabs(c) == 2:
        print("два")
    elif math.fabs(c) == 3:
        print("три")
    elif math.fabs(c) == 4:
        print("четыре")
    elif math.fabs(c) == 5:
        print("пять")
    elif math.fabs(c) == 6:
        print("шесть")
```

```
elif math.fabs(c) == 7:  
    print("семь")  
elif math.fabs(c) == 8:  
    print("восемь")  
elif math.fabs(c) == 9:  
    print("девять")  
else:  
    print("Ошибка")
```

```
Value of c? -5  
Минус  
пять
```

Рисунок 19– Результат выполнения программы при $x = -5$

```
Value of c? 1  
Плюс  
один
```

Рисунок 20 – Результат выполнения программы при $x = 1$

```
Value of c? 10  
Ошибка
```

Рисунок 21 – Результат выполнения программы при $x = 10$

Задание 2.

Вариант 11. Определить, есть ли среди трёх заданных чисел чётные.

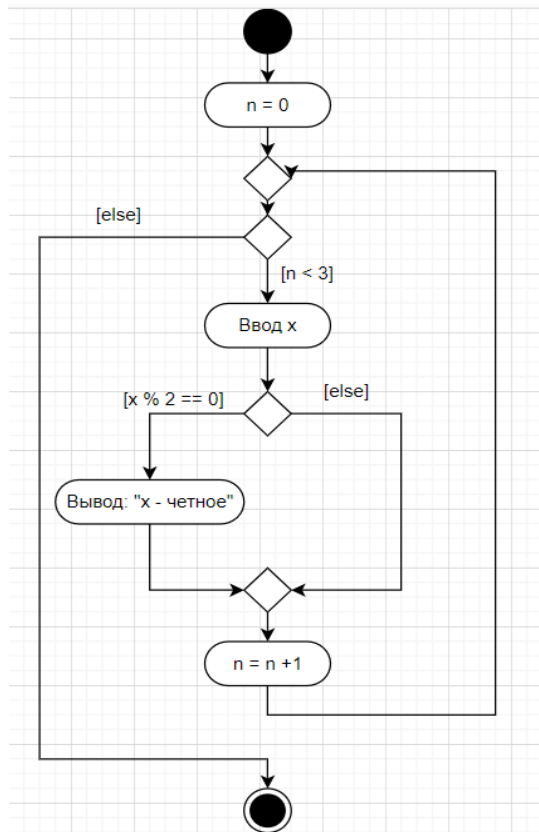


Рисунок 22 – UML-диаграмма деятельности

Код:

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

if __name__ == '__main__':
    n = 0
    while n < 3:
        x = float(input("Enter x: "))
        if x % 2 == 0:
            print(f"{x} is even")
        n = n + 1
```

```
Enter x: 1
Enter x: 6
6.0 is even
Enter x: 3
```

Рисунок 23 – Результат выполнения программы

Задание 3.

Вариант 10(32). Сколько можно купить быков, коров и телят, платя за быка 10 р., за корову – 5 р., а за теленка – 0,5 р., если на 100 р. надо купить 100 голов скота?

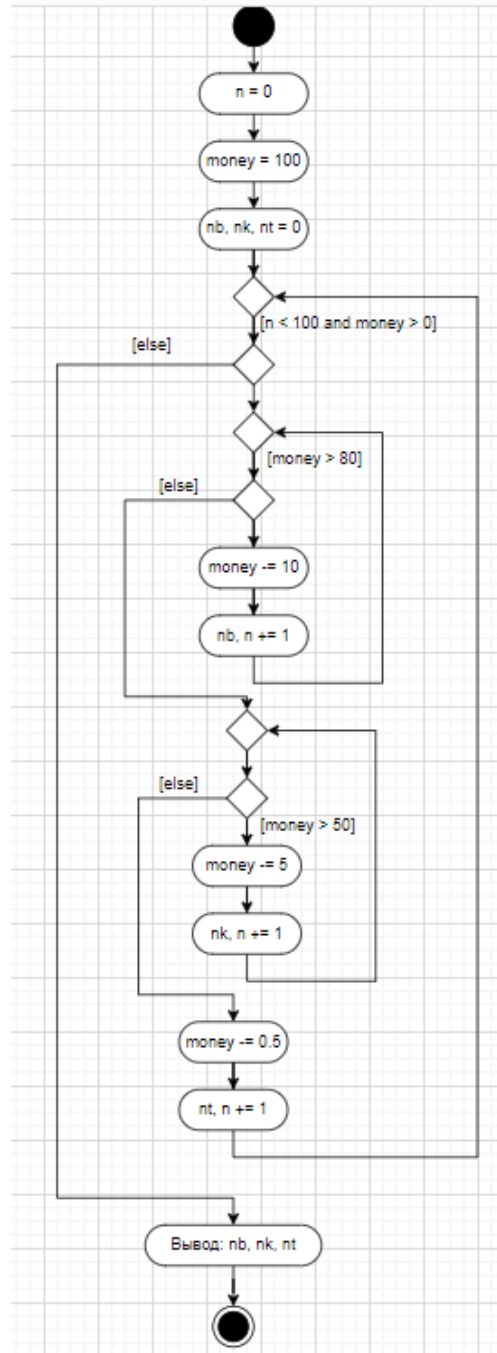


Рисунок 24 – UML-диаграмма деятельности

Код:

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

if __name__ == '__main__':
    n = 0
    money = 100
    nb = 0
    nk = 0
    nt = 0
    while n < 100 and money > 0:
        while money > 80:
            money -= 10
            nb += 1
            n += 1
        while money > 50:
            money -= 5
            nk += 1
            n += 1
        money -= 0.5
        nt += 1
        n += 1
    print(f"You can bay {nb} bull(s), {nk} cow(s) and {nt} calve(s)")
```

```
You can bay 2 bull(s), 6 cow(s) and 92 calve(s)
```

Рисунок 25 – 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 содержится 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.