Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Условные операторы и циклы в языке Python»

ОТЧЕТ
по лабораторной работе №5
дисциплины
«Основы программной инженерии»

Выполнил:
Мизин Глеб Егорович
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения

_____
(подпись)

Проверил:

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2022 г.

Проработка примеров из лабораторной работы:

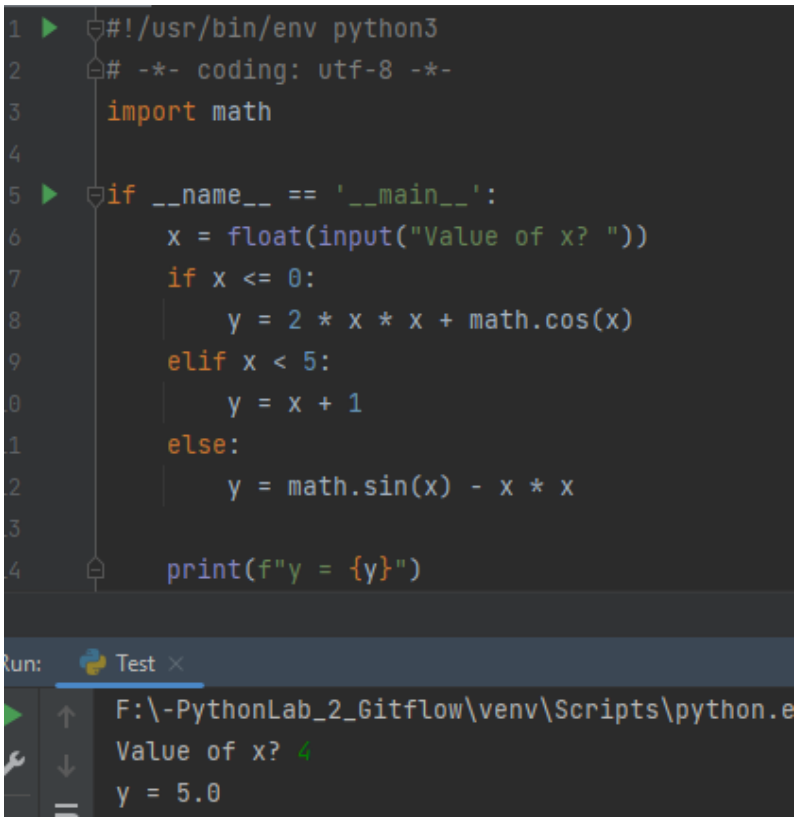


Рисунок 1 – Пример №1

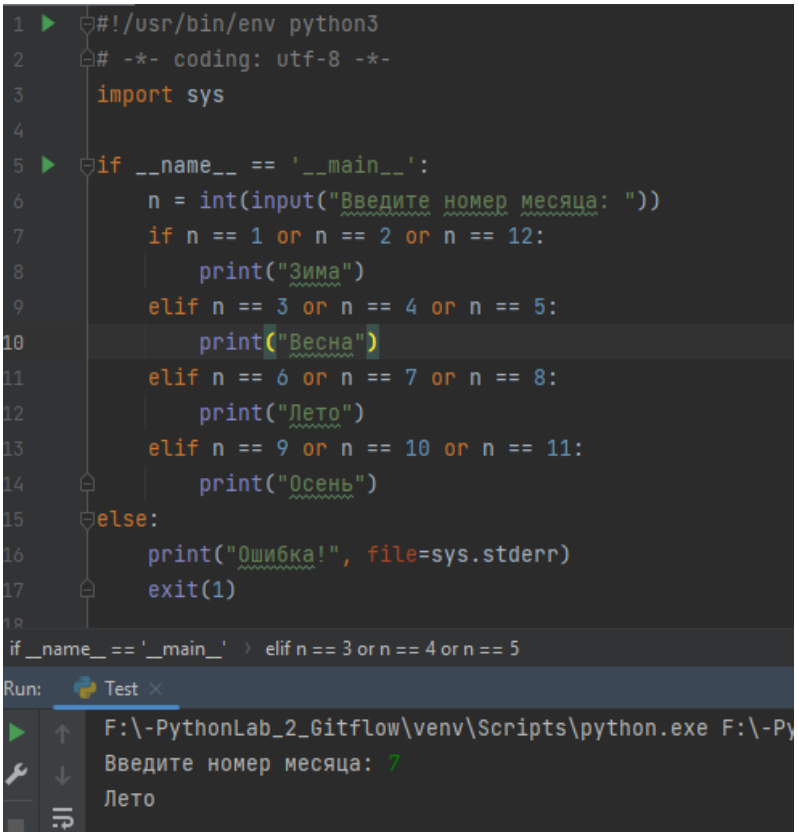


Рисунок 2 – Пример №2

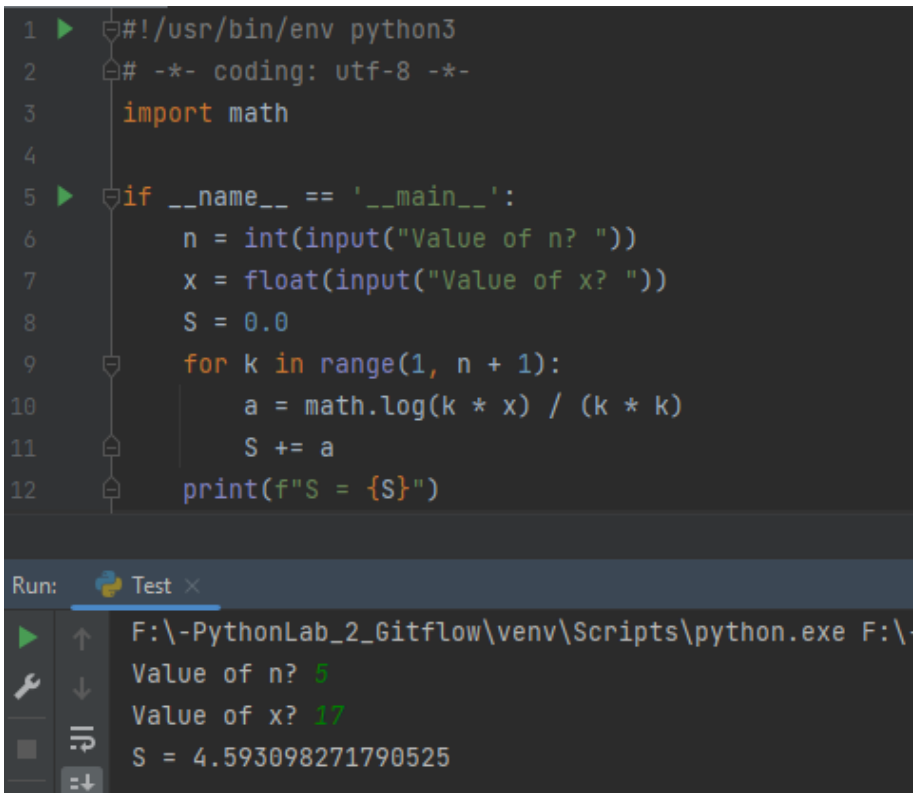


Рисунок 3 – Пример №3

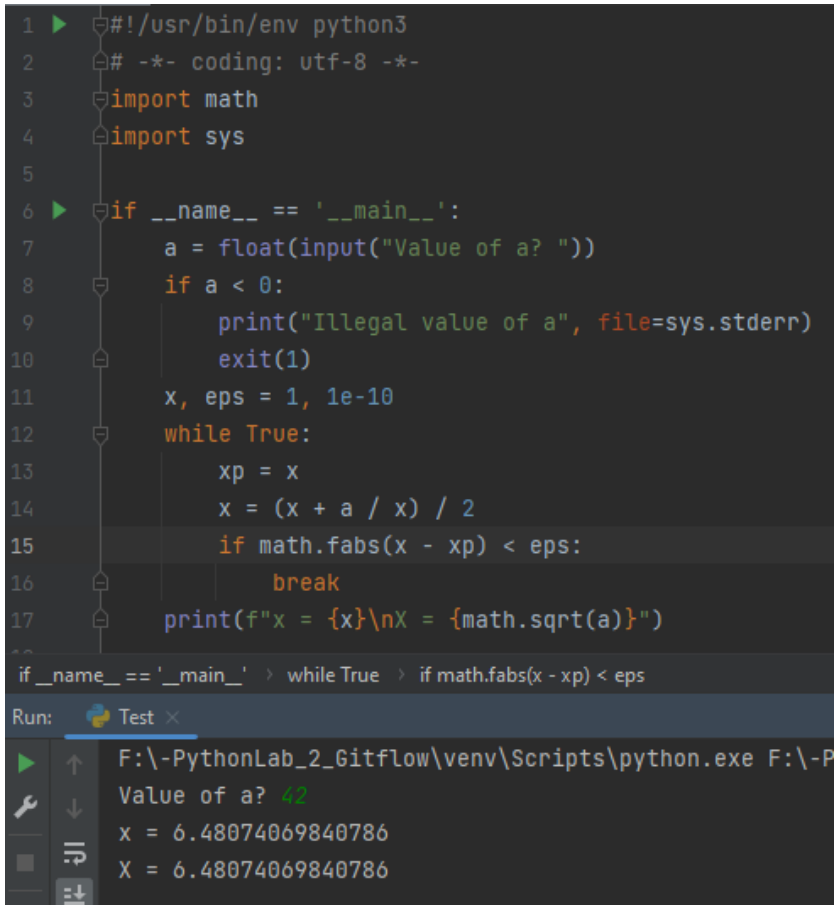


Рисунок 4.1 – Пример №4

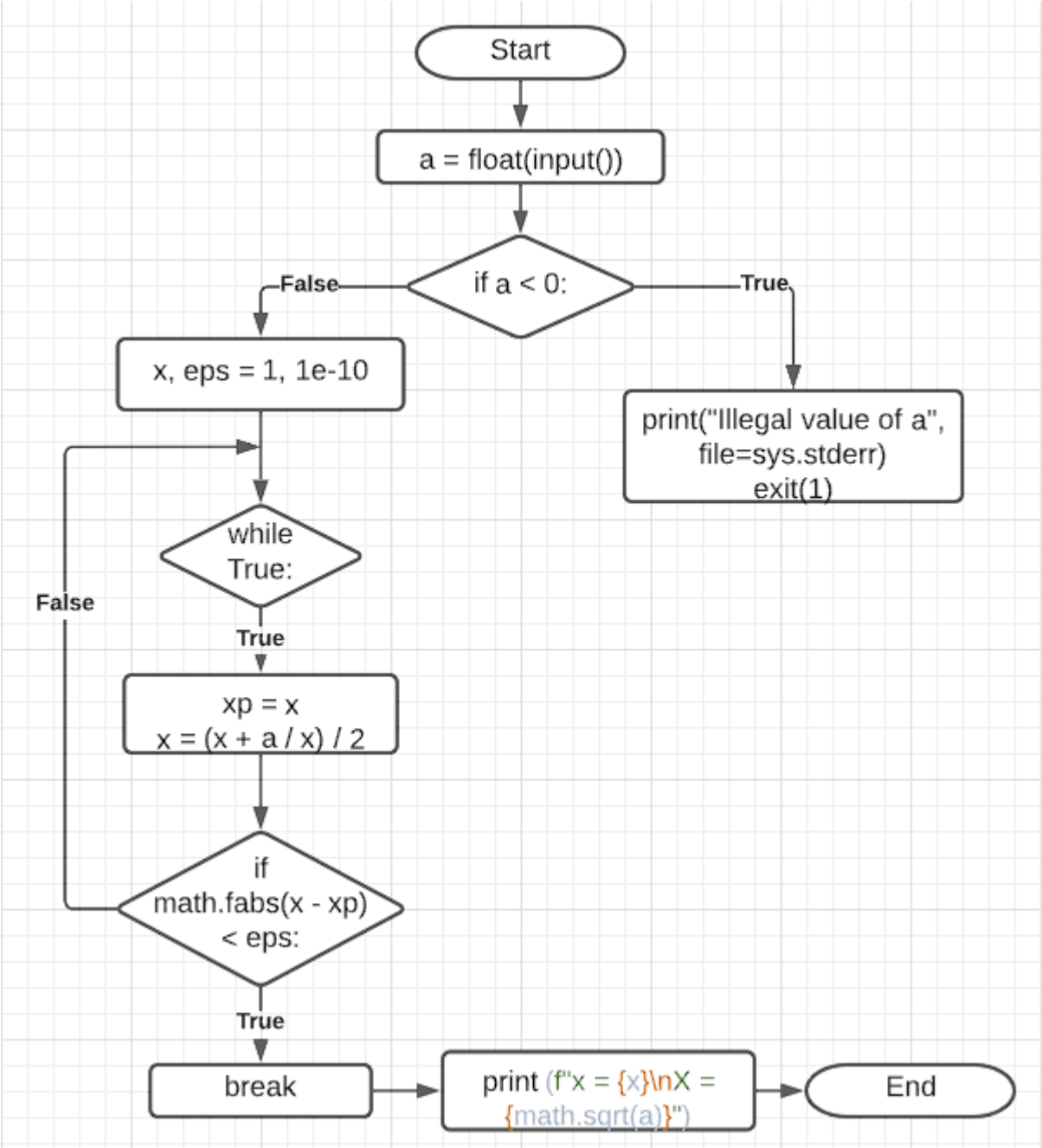


Рисунок 4.2 – UML диаграмма примера №4

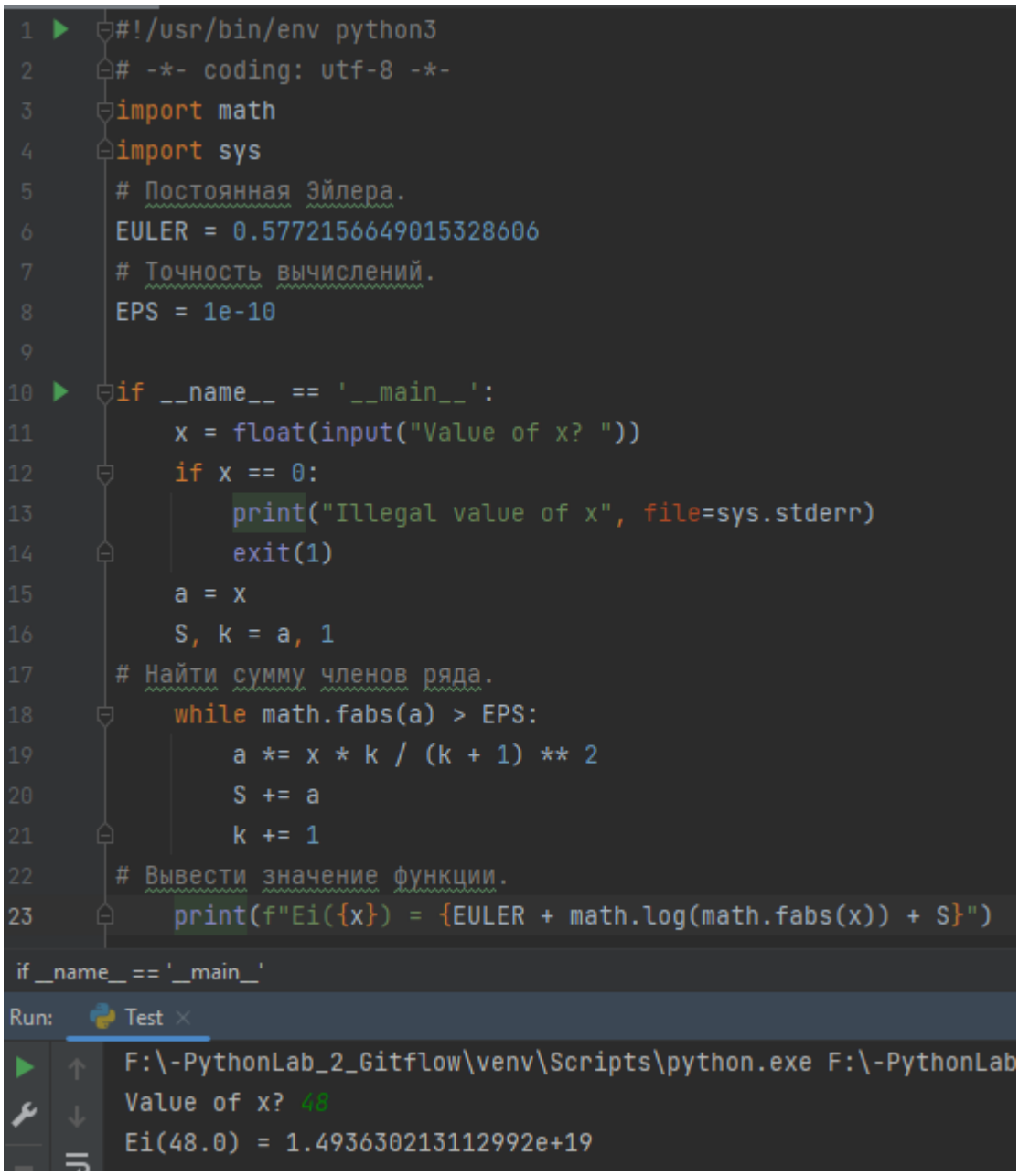


Рисунок 5.1 – Пример №5

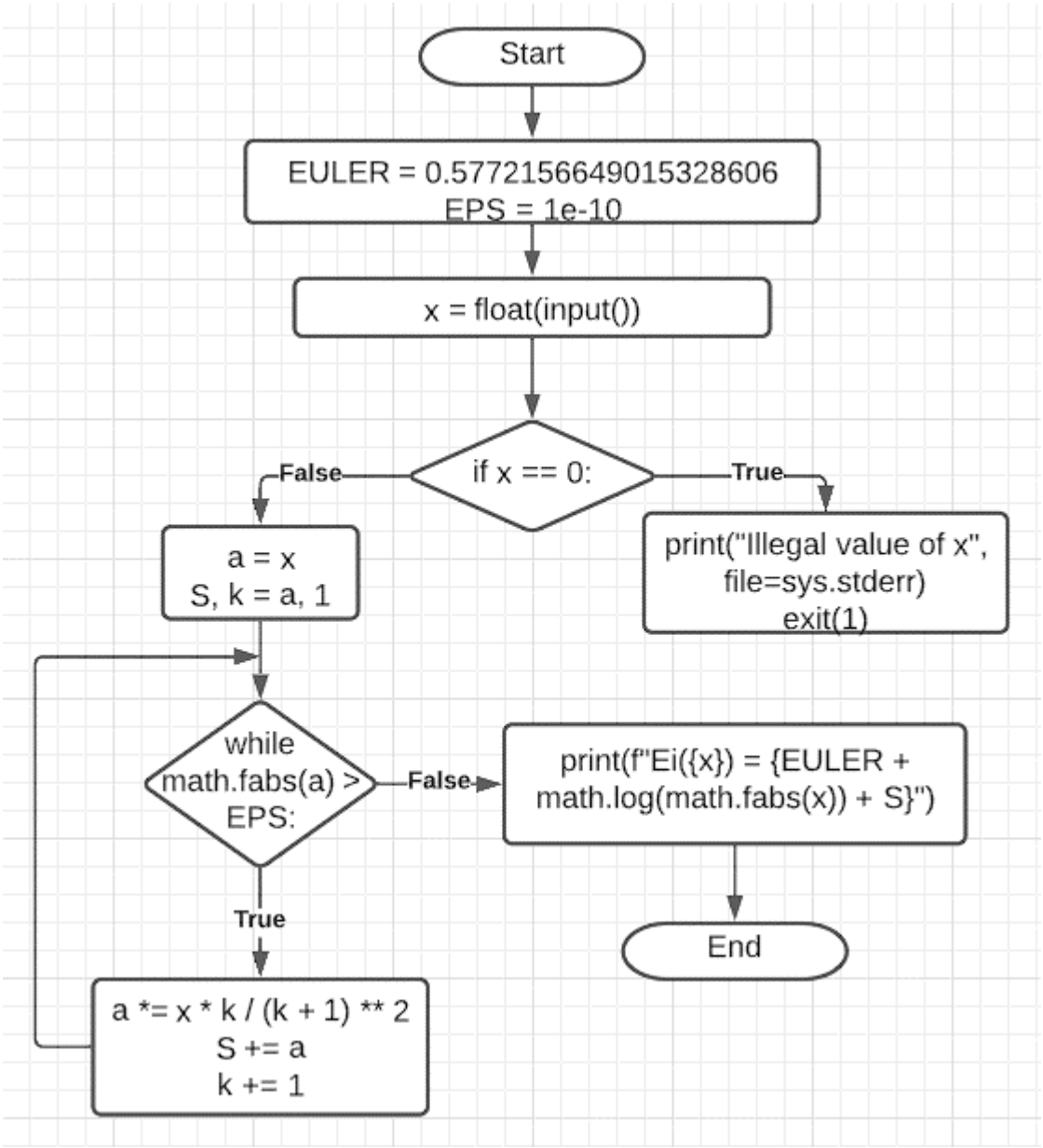


Рисунок 5.2 – UML диаграмма примера №5

Задание№1: дано целое число C такое, что |C| <9. Вывести это число в словесной форме, учитывая его знак.

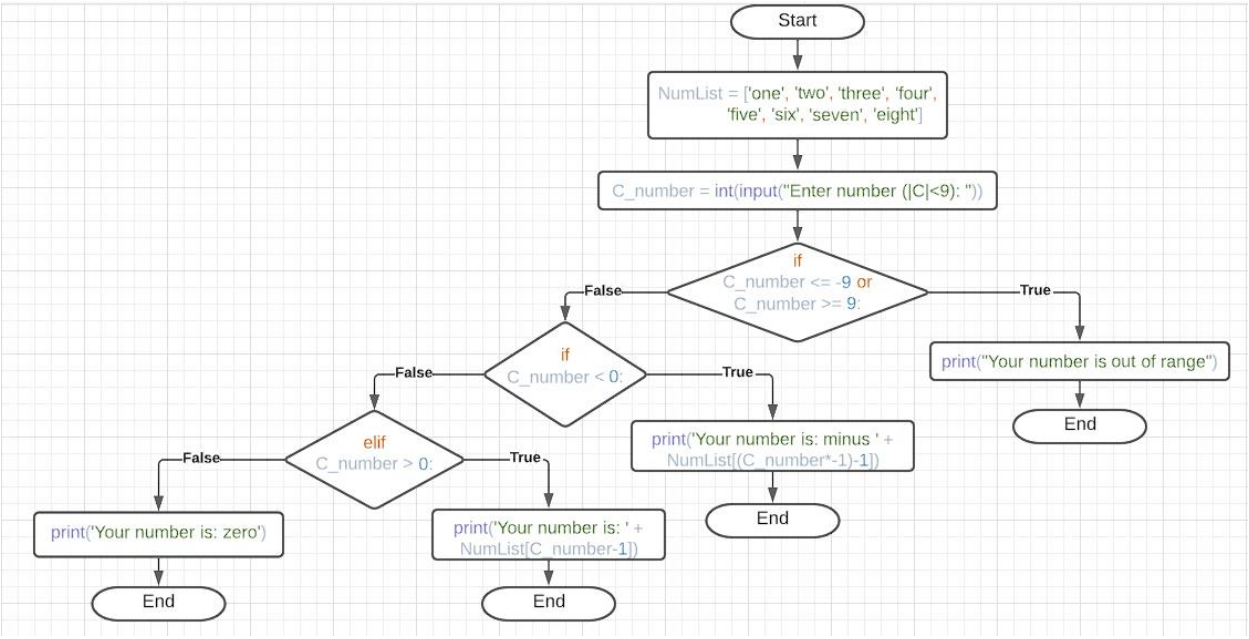


Рисунок 1 – UML диаграмма первого задания

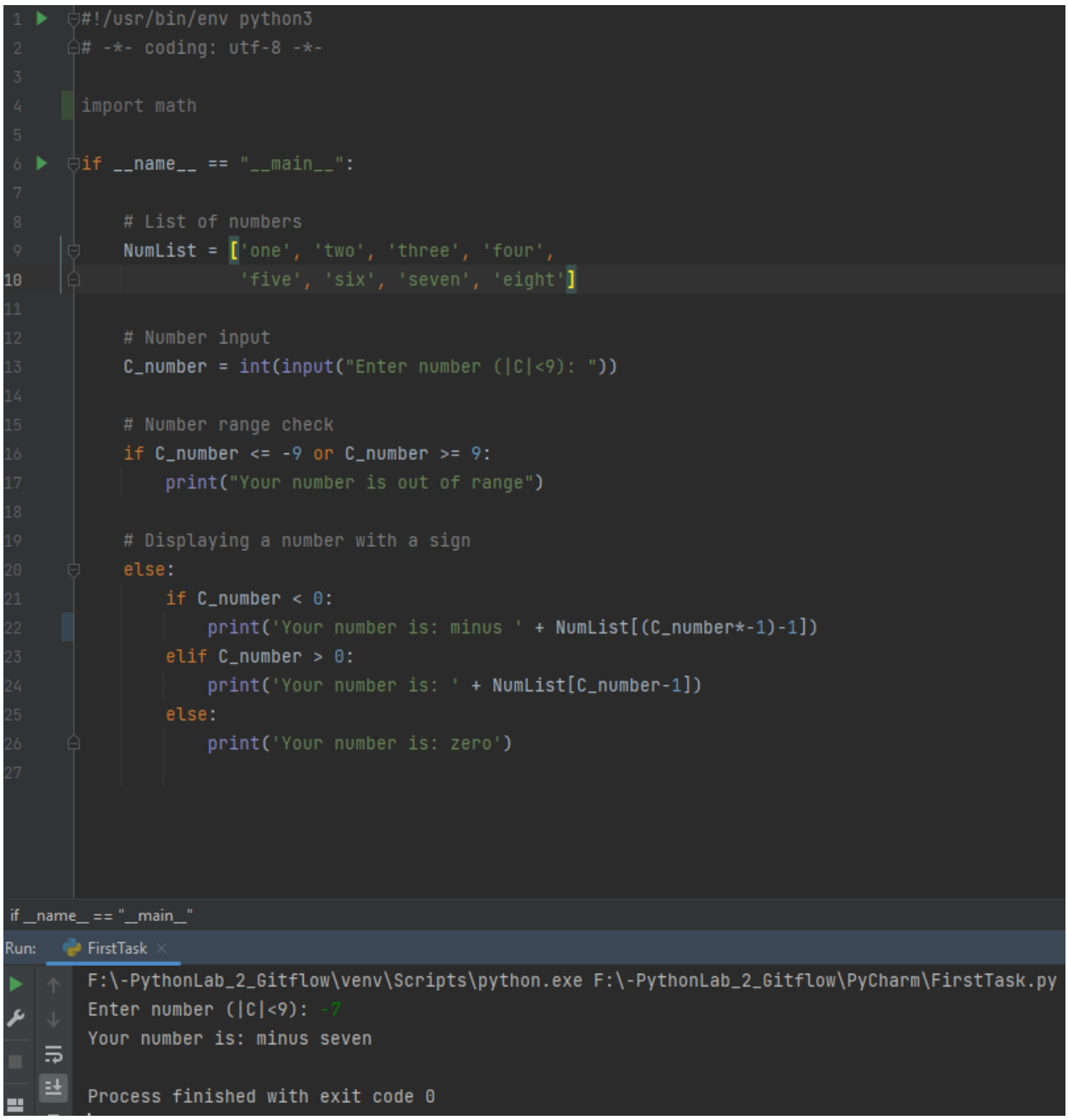


Рисунок 2 – Код и результат работы программы №1

Задание №2: какая из точек A (a1, a2) или B (b1, b2) находится дальше от начала координат?

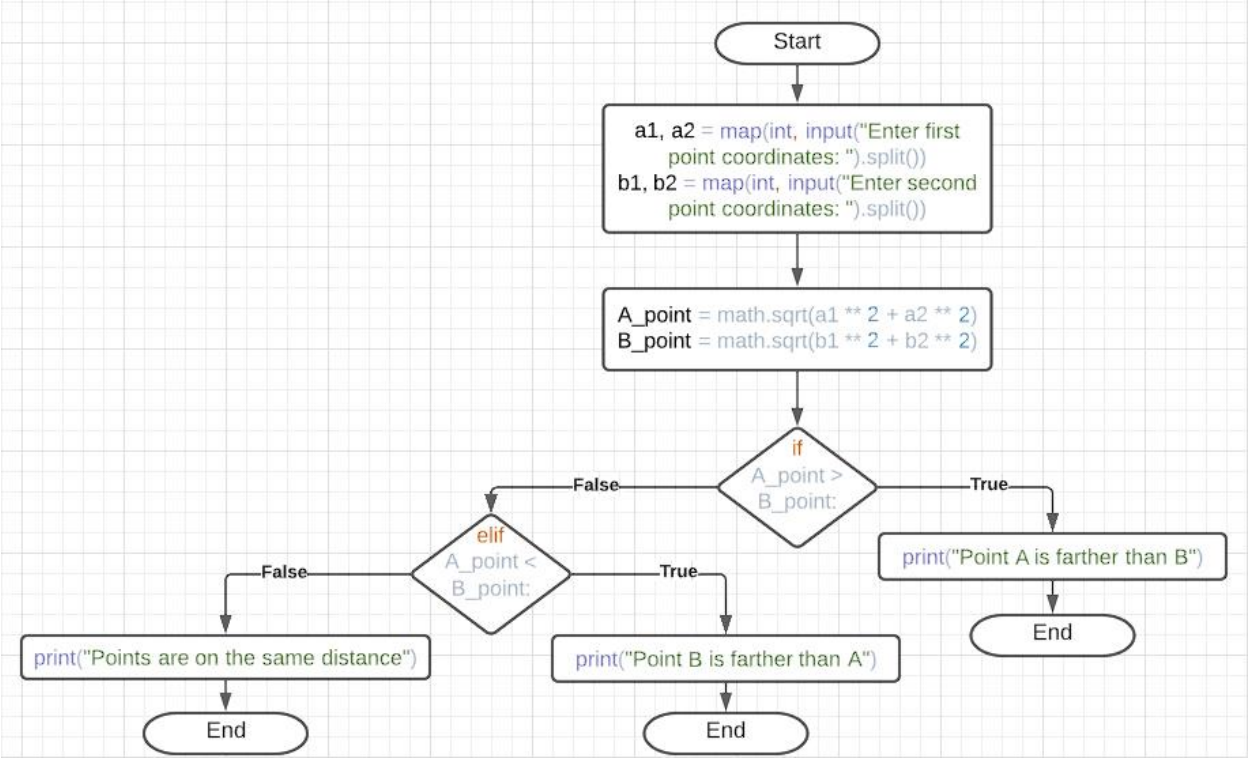


Рисунок 3 – UML диаграмма второго задания

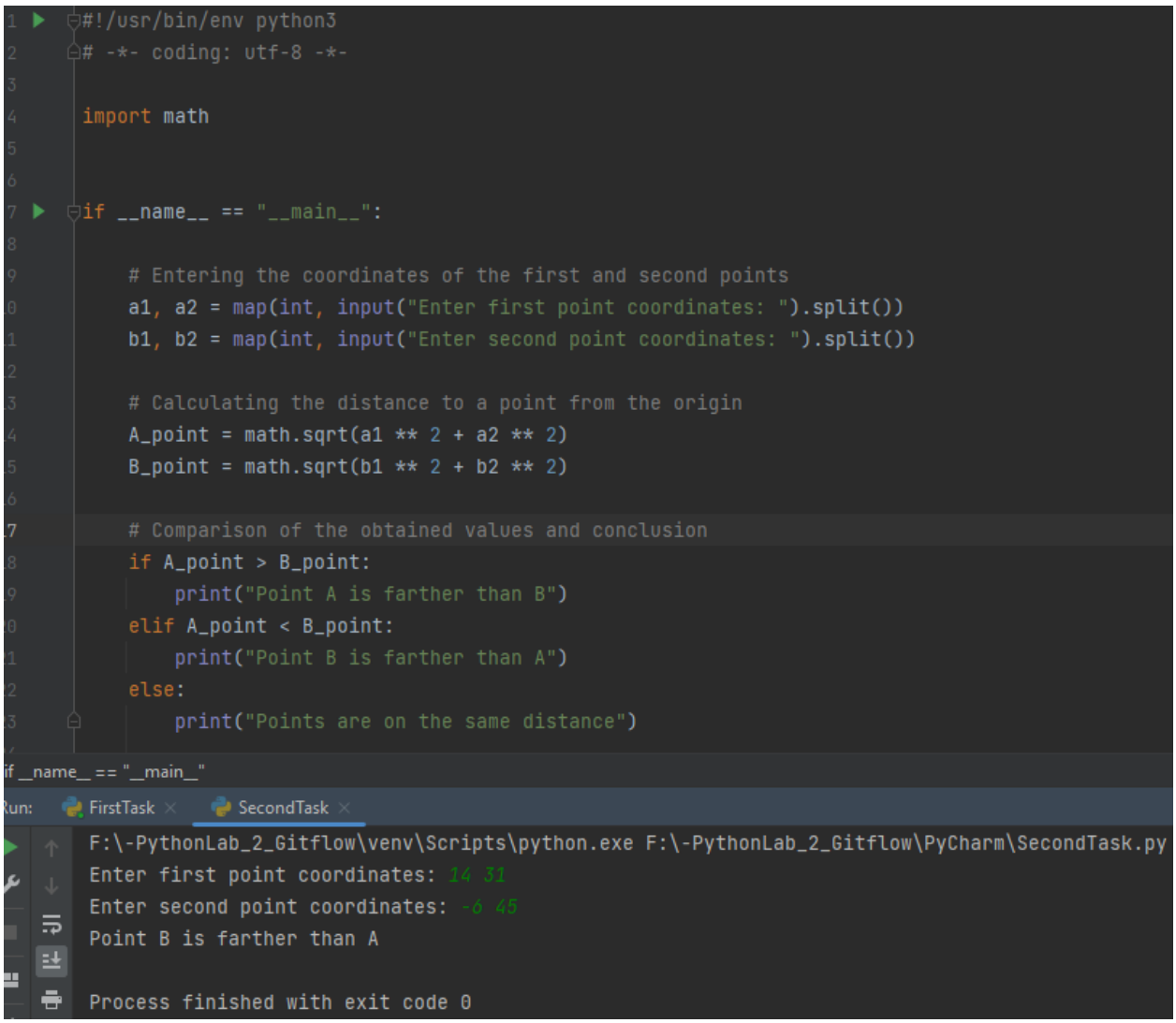


Рисунок 4 – Код и результат работы программы №2

Задание №3: у гусей и кроликов вместе 64 лапы. Сколько могло быть кроликов и гусей (указать все сочетания, которые возможны).

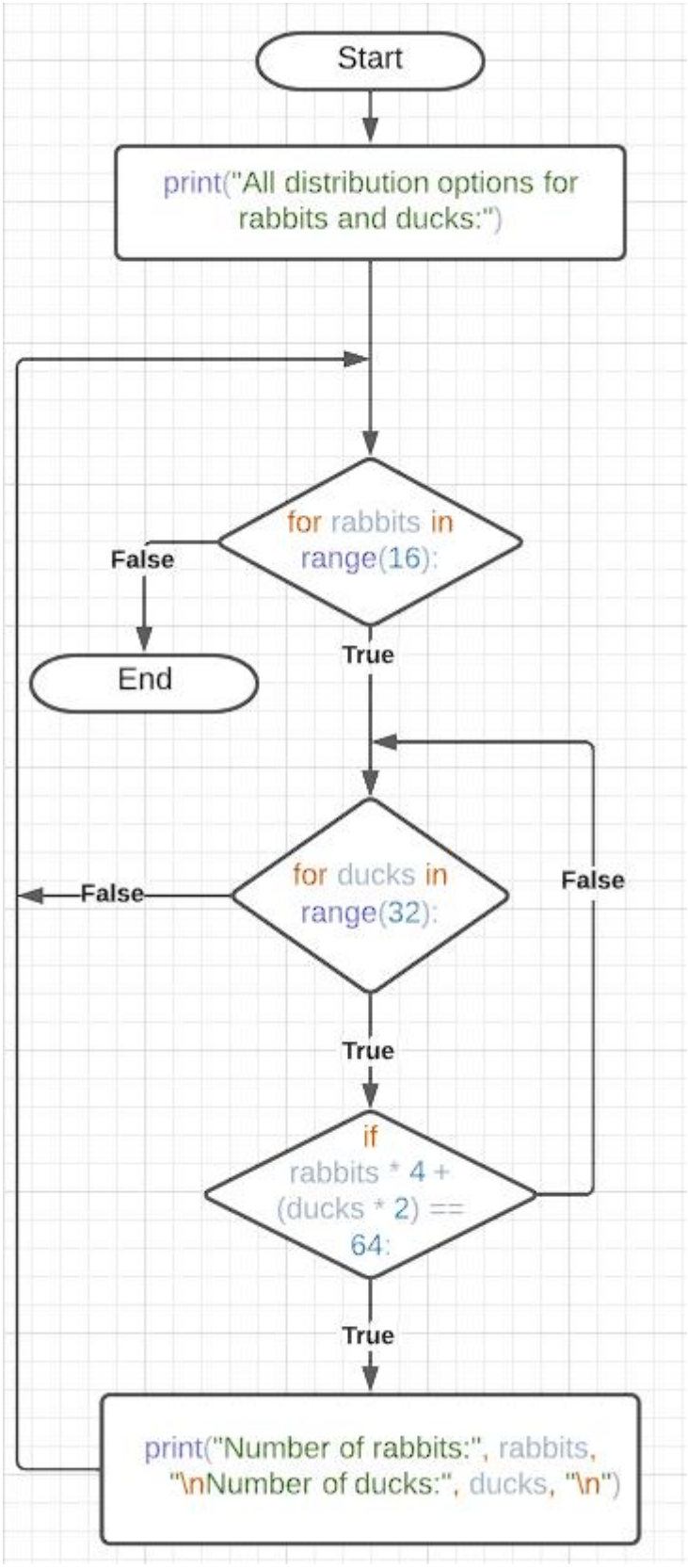
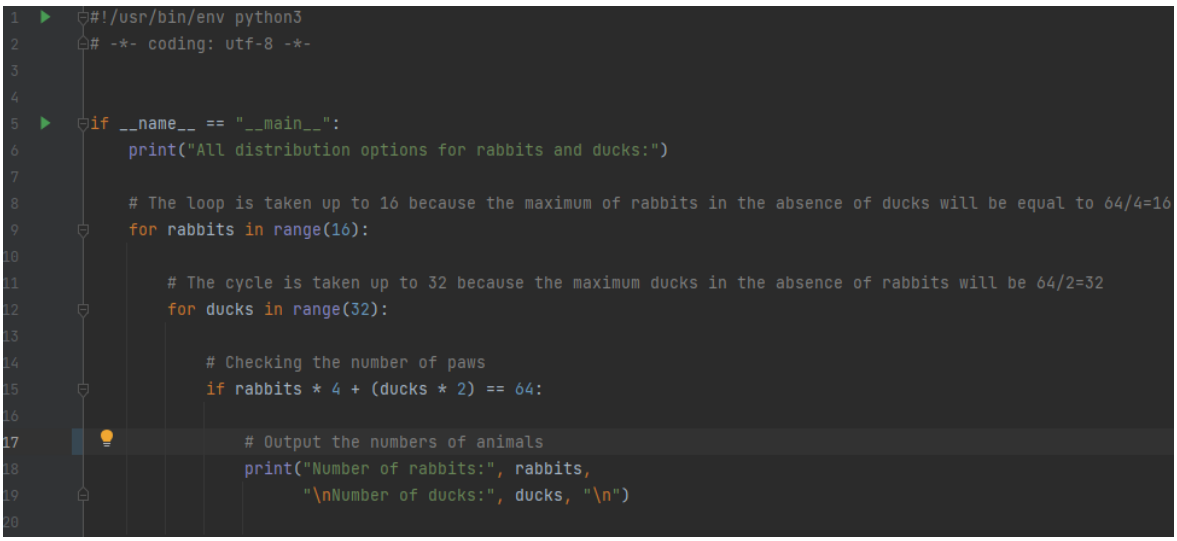


Рисунок 5 – UML диаграмма третьего задания



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


if __name__ == "__main__":
    print("All distribution options for rabbits and ducks:")

    # The loop is taken up to 16 because the maximum of rabbits in the absence of ducks will be equal to 64/4=16
    for rabbits in range(16):

        # The cycle is taken up to 32 because the maximum ducks in the absence of rabbits will be 64/2=32
        for ducks in range(32):

            # Checking the number of paws
            if rabbits * 4 + (ducks * 2) == 64:

                # Output the numbers of animals
                print("Number of rabbits:", rabbits,
                      "\nNumber of ducks:", ducks, "\n")

```

Рисунок 6 – Код программы №3

```
All distribution options for rabbits and ducks:
Number of rabbits: 1
Number of ducks: 30

Number of rabbits: 2
Number of ducks: 28

Number of rabbits: 3
Number of ducks: 26

Number of rabbits: 4
Number of ducks: 24

Number of rabbits: 5
Number of ducks: 22
```

Рисунок 7.1 – Вывод программы №3

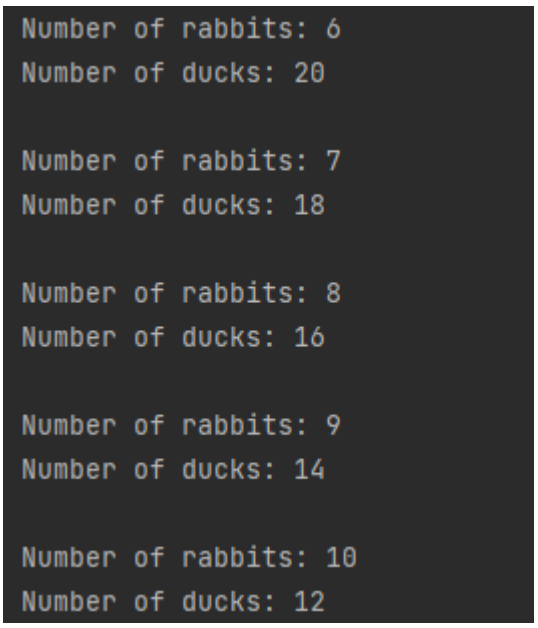


Рисунок 7.2 – Вывод программы №3

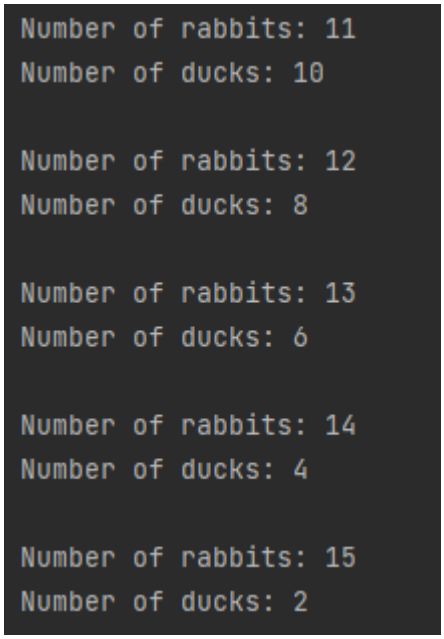


Рисунок 7.3 – Вывод программы №3

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 вводится с клавиатуры.

$$\mathrm{Si}(x) = \int_0^x \frac{\sin t}{t}\,dt = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)(2n+1)!}.$$

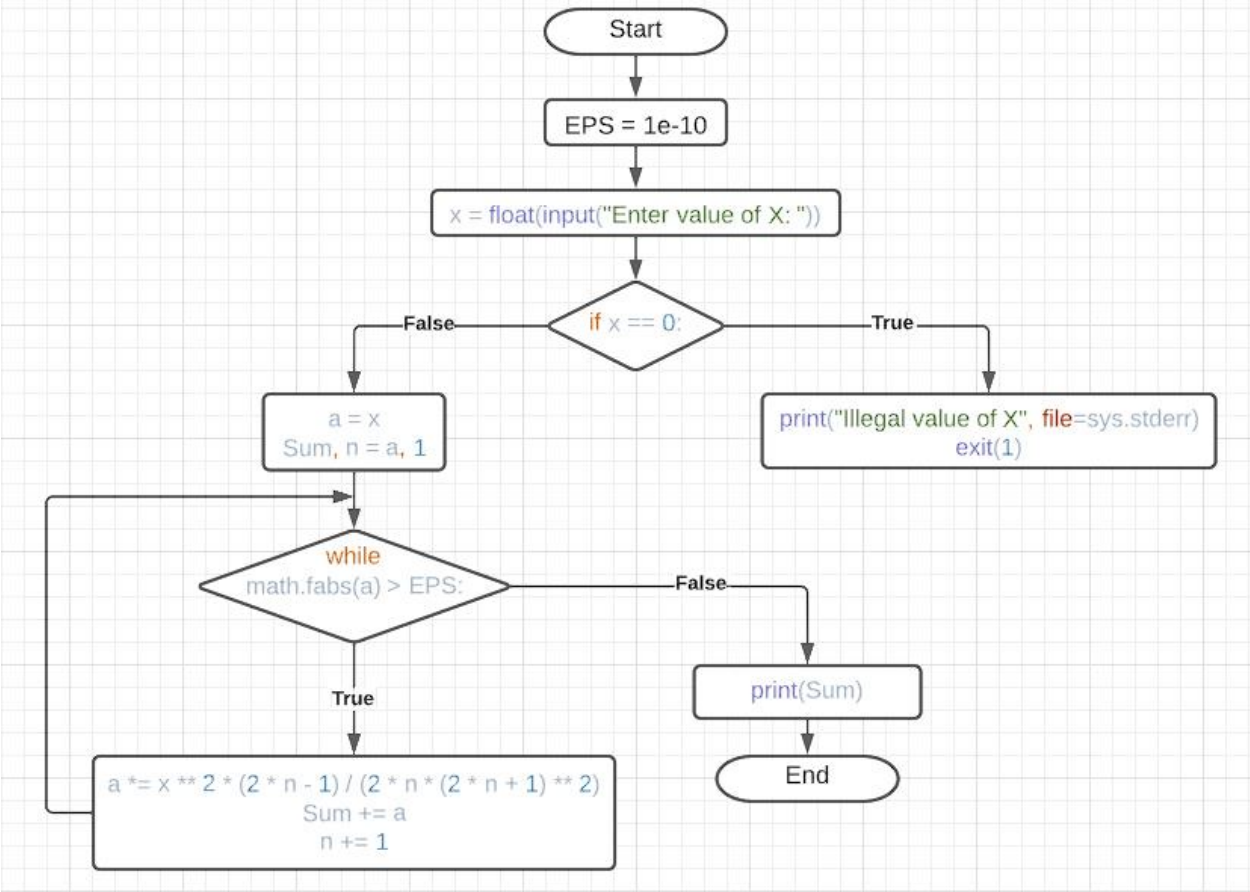


Рисунок 8 – UML диаграмма задания повышенной сложности

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

import math
import sys

# Calculation accuracy.
EPS = 1e-10

if __name__ == "__main__":
    x = float(input("Enter value of X: "))
    if x == 0:
        print("Illegal value of X", file=sys.stderr)
        exit(1)

    a = x
    Sum, n = a, 1

    # Find the sum of the terms of the series.
    while math.fabs(a) > EPS:
        a *= x ** 2 * (2 * n - 1) / (2 * n * (2 * n + 1) ** 2)
        Sum += a
        n += 1

    # Function value output.
    print(Sum)
```

Рисунок 9 – Код программы

```
Enter value of X: 6
42.995061112445526

Process finished with exit code 0
```

Рисунок 10 – Вывод программы

Контрольные вопросы

1. Для чего нужны диаграммы деятельности UML?

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. Что такое состояние действия и состояние деятельности?

Состояния действия не могут быть подвергнуты декомпозиции. Кроме того, они атомарны. Это значит, что внутри них могут происходить различные события, но выполняемая в состоянии действия работа не может быть прервана. Обычно предполагается, что длительность одного состояния действия занимает неощутимо малое время

Состояния деятельности могут быть подвергнуты дальнейшей декомпозиции, вследствие чего выполняемую деятельность можно представить с помощью других диаграмм деятельности. Состояния деятельности не являются атомарными, то есть могут быть прерваны. Предполагается, что для их завершения требуется заметное время.

3. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?

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?

Наличием условных операторов

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

Команда, которая выполняется только при каком-либо условии

7. Какие операторы сравнения используются в Python?

оператор < , «меньше»;
оператор <= , «меньше или равно;
оператор == , «равно»;
оператор != , «не равно»;
оператор > , «больше»;
оператор >= , «больше или равно».

8. Что называется простым условием? Приведите примеры.

Простым условием (отношением) называется выражение, составленное из двух арифметических выражений или двух текстовых величин (иначе их еще называют операндами), связанных одним из знаков приведенных в ответе на 7 вопрос

9. Что такое составное условие? Приведите примеры.

Составное условие – логическое выражение, содержащее несколько простых условий, объединенных логическими операциями.

10. Какие логические операторы допускаются при составлении сложных условий?

Логическое И, логическое ИЛИ, логическое отрицание

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

Да

12. Какой алгоритм является алгоритмом циклической структуры?

Алгоритм циклической структуры — это алгоритм, в котором происходит многократное повторение одного и того же участка программы. Такие повторяемые участки вычислительного процесса называются циклами.

13. Типы циклов в языке Python.

В Python есть два вида циклов: for и while

14. Назовите назначение и способы применения функции range

Функция range возвращает неизменяемую последовательность чисел в виде объекта range. Синтаксис функции:

range(stop)

range(start, stop[, step])

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(0, 15, 2)

16. Могул ли быть циклы вложенными?

Вложенный цикл - цикл который выполняется внутри другого цикла. Обычно вложенные циклы используются для работы с двумя измерениями.

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

Пример бесконечного цикла:

```
a = 0
while a == 0:
    print("A")
```

Выйти из такого цикла можно при помощи оператора break

18. Для чего нужен оператор break ?

Оператор break предназначен для досрочного прерывания работы цикла while.

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

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

20. Для чего нужны стандартные потоки stdout и stderr?

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.

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

По умолчанию функция print использует поток stdout. Для того, чтобы использовать поток stderr необходимо передать его в параметре file функции print.

22. Каково назначение функции exit ?

Если в процессе выполнения программы произошли ошибки, программа должна передать операционной системе код возврата отличный от нуля. В Python завершить программу и передать операционной системе заданный код возврата можно посредством функции exit.