**МИНИCTEPCTBO 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»**

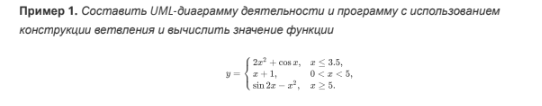
|  |
| --- |
| Выполнил студент группы ПИЖ-б-о-21-1 |
| Халимендик Я. Д. « » 2022г. |
| Подпись студента\_\_\_\_\_\_\_\_\_\_\_ |
| Работа защищена « »\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г. |
| Проверил Воронкин Р.А. \_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

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:



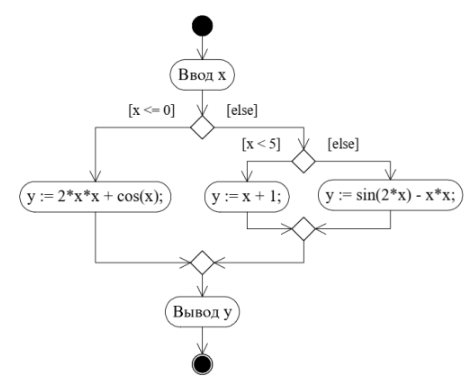


Рисунок 1 – UML-диаграмма деятельности

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}")

****

Рисунок 2 – Результат выполнения программы при х = -1



Рисунок 3 – Результат выполнения программы при х = 3



Рисунок 4 – Результат выполнения программы при х = 6



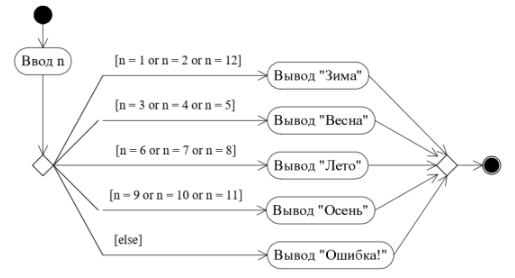


Рисунок 5 – UML-диаграмма деятельности

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)



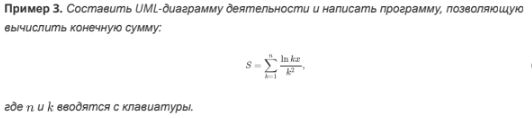
Рисунок 6 – Результат выполнения программы при х = 5



Рисунок 7 – Результат выполнения программы при х = 9



Рисунок 8 – Результат выполнения программы при х = -9



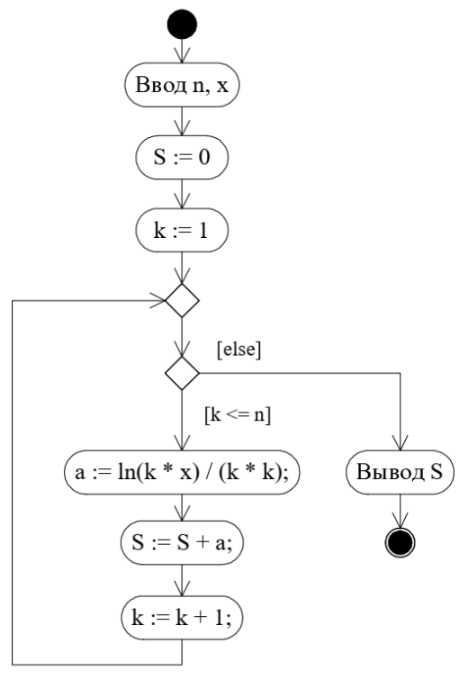


Рисунок 9 – UML-диаграмма деятельности

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}")

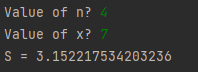


Рисунок 10 – Результат выполнения программы при n = 4, х = 7

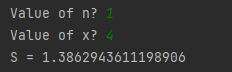
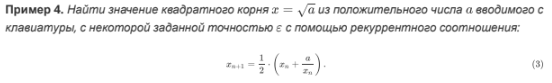


Рисунок 11 – Результат выполнения программы при n = 1, х = -4



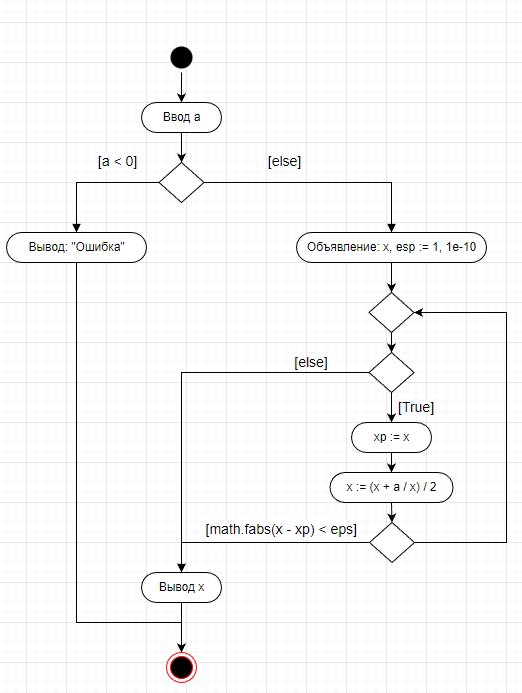


Рисунок 12 – UML-диаграмма деятельности

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)}")

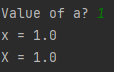


Рисунок 13 – Результат выполнения программы при а = 1

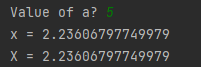


Рисунок 14 – Результат выполнения программы при а = 5

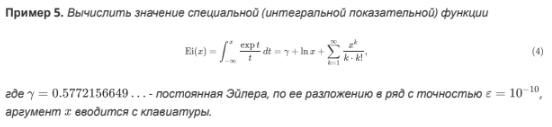


Рисунок 15 – UML-диаграмма деятельности

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}")



Рисунок 16 – Результат выполнения программы при х = 4



Рисунок 17 – Результат выполнения программы при х = 0

10. Для примеров 4 и 5 постройте UML-диаграмму деятельности. Для построения диаграмм деятельности использовать веб-сервис Google <https://www.diagrams.net/>.

11. Выполните индивидуальные задания, согласно своему варианту. Для заданий повышенной сложности номер варианта должен быть получен у преподавателя.

12. Приведите в отчете скриншоты работы программ и UML-диаграммы деятельности решения индивидуальных заданий.

13. Зафиксируйте сделанные изменения в репозитории.

14. Выполните слияние ветки для разработки с веткой main / master.

15. Отправьте сделанные изменения на сервер GitHub.

16. Отправьте адрес репозитория GitHub на электронный адрес преподавателя.

Задание 1 Решить задачу согласно варианту, составить UML-диаграмму деятельности и программу с использованием конструкций ветвления. Номер варианта необходимо получить у преподавателя.

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

Задание 2 Решить задачу согласно варианту, составить UML-диаграмму деятельности и программу с использованием конструкций ветвления. Номер варианта необходимо получить у преподавателя.

Вариант 9(32). Найти координаты точки пересечения прямых заданных уравнениями a1x + b1y + c1 = 0 и a2x + b2y + c2 = 0, либо сообщить совпадают, параллельны или не существуют.

Задание 3 Составить UML-диаграмму деятельности и программу с использованием конструкций цикла для решения задачи. Номер варианта необходимо получить у преподавателя.

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?

Вопросы для защиты работы:

1. Для чего нужны диаграммы деятельности UML?

2. Что такое состояние действия и состояние деятельности?

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

4. Какой алгоритм является алгоритмом разветвляющейся структуры?

5. Чем отличается разветвляющийся алгоритм от линейного?

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

7. Какие операторы сравнения используются в Python?

8. Что называется простым условием? Приведите примеры.

9. Что такое составное условие? Приведите примеры.

10. Какие логические операторы допускаются при составлении сложных условий?

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

12. Какой алгоритм является алгоритмом циклической структуры?

13. Типы циклов в языке Python.

14. Назовите назначение и способы применения функции range .

15. Как с помощью функции range организовать перебор значений от 15 до 0 с шагом 2?

16. Могул ли быть циклы вложенными?

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

18. Для чего нужен оператор break?

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

20. Для чего нужны стандартные потоки stdout и stderr?

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

22. Каково назначение функции exit?