**Практическая работа 01. 06.09.2024 (урок 3)**

1) import sys

import math

eps = 1

while 1 + eps != 1:

eps /= 2

print(f"Машинный нуль для 1: {eps \* 2}")

eps = 1

while eps != 0:

eps /= 2

print(f"Машинный нуль для положительной степени двойки: {eps}")

eps = 1

while eps != 0:

eps \*= 2

print(f"Машинный нуль для отрицательной степени двойки: {eps}")

print(f"Минимальное число с плавающей точкой: {sys.float\_info.min}")

print(f"Максимальное число с плавающей точкой: {sys.float\_info.max}")

print(f"Точность чисел с плавающей точкой: {sys.float\_info.epsilon}")

2) def decimal\_to\_binary\_integer(n):

"""Перевод целого десятичного числа в двоичную систему счисления."""

if n < 0:

return '-' + decimal\_to\_binary\_integer(-n)

elif n == 0:

return '0'

binary = ''

while n > 0:

binary = str(n % 2) + binary

n //= 2

return binary

def decimal\_to\_binary\_fraction(fraction, precision=10):

"""Перевод правильной десятичной дроби в двоичную систему счисления."""

binary\_fraction = '.'

count = 0

while fraction > 0 and count < precision:

fraction \*= 2

if fraction >= 1:

binary\_fraction += '1'

fraction -= 1

else:

binary\_fraction += '0'

count += 1

return binary\_fraction

def decimal\_to\_binary(decimal\_number):

"""Перевод целого числа и дроби в двоичную систему счисления."""

if '.' in decimal\_number:

integer\_part, fractional\_part = decimal\_number.split('.')

integer\_part = int(integer\_part)

fractional\_part = float('0.' + fractional\_part)

else:

integer\_part = int(decimal\_number)

fractional\_part = 0.0

binary\_integer = decimal\_to\_binary\_integer(integer\_part)

binary\_fraction = decimal\_to\_binary\_fraction(fractional\_part)

if fractional\_part > 0:

return f"{binary\_integer}{binary\_fraction}"

else:

return binary\_integer

decimal\_number = input("Введите десятичное число (целое или дробное): ")

binary\_representation = decimal\_to\_binary(decimal\_number)

print(f"Двоичное представление: {binary\_representation}")

3) l = 5.15

w = 4.33

delta = 0.01

S = l \* w

relative\_error = (delta / l) + (delta / w)

delta\_S = S \* relative\_error

print(f"Площадь комнаты: {S:.4f} м²")

print(f"Абсолютная погрешность площади: ±{delta\_S:.4f} м²")

4) import math

number = 23

sqrt\_value = math.sqrt(number)

max\_relative\_error = 0.01

max\_absolute\_error = max\_relative\_error \* sqrt\_value

def calculate\_decimal\_places(max\_error):

d = 0

while True:

if 0.5 \* 10\*\*(-d) < max\_error:

break

d += 1

return d

decimal\_places = calculate\_decimal\_places(max\_absolute\_error)

print(f"Квадратный корень из {number} ≈ {sqrt\_value:.{decimal\_places}f}")

print(f"Для достижения относительной погрешности не более {max\_relative\_error}, необходимо {decimal\_places} десятичных знаков.")

5) import math

e = math.e

e\_rounded\_5 = round(e, 5)

e\_rounded\_6 = round(e, 6)

absolute\_error\_5 = abs(e - e\_rounded\_5)

absolute\_error\_6 = abs(e - e\_rounded\_6)

relative\_error\_5 = absolute\_error\_5 / e

relative\_error\_6 = absolute\_error\_6 / e

print(f"Число e: {e:.11f}")

print(f"Округление до 5 знаков: {e\_rounded\_5:.5f}")

print(f"Абсолютная погрешность (5 знаков): {absolute\_error\_5:.11f}")

print(f"Относительная погрешность (5 знаков): {relative\_error\_5:.11f}")

print(f"\nОкругление до 6 знаков: {e\_rounded\_6:.6f}")

print(f"Абсолютная погрешность (6 знаков): {absolute\_error\_6:.11f}")

print(f"Относительная погрешность (6 знаков): {relative\_error\_6:.11f}")