

ФГАОУ ВО "МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Лабораторная работа №3

Организация циклов

Вариант №11

по дисциплине:

Основы программирования

Выполнил

студент 1 курса
группы 211-321
Журавлев Д.А.

Проверил

Никишина И.Н.

МОСКВА 2021

Постановка задачи

Задание 1

Вычислить и вывести на экран или в файл в виде таблицы значения функции, заданной графически, на интервале от $X_{нач}$ до $X_{кон}$ с шагом dx . Интервал и шаг должны быть такими что бы проверить все ветви программы.

Задание 2

Для 10 выстрелов координаты которых задаются генератором случайных чисел, вывести текстовые сообщения о попадании в мишень

Задание 3

Вычислить и вывести на экран в виде таблицы значения функции заданной с помощью ряда Тейлора, на интервале от $X_{нач}$ до $X_{кон}$ с шагом dx и с точностью ϵ .

$$. \operatorname{Arth} = \sum_{n=0}^{\infty} \frac{1}{(2 \cdot n + 1) \cdot x^{2 \cdot n + 1}} = \frac{1}{x} + \frac{1}{3 \cdot x^3} + \frac{1}{5 \cdot x^5} + \dots, \quad |x| > 1.$$

Теоретическая часть

Для математических вычислений в Python имеются как встроенные, так и дополнительные функции и методы. Для применения дополнительных математических функций необходимо использовать модуль `math`, который подключается с помощью инструкции:

```
import math
```

Для ввода данных используется инструкция `input()`, которая возвращает строку. Введенные значения должны быть преобразованы к числовому формату перед использованием в арифметических выражениях.

Для предотвращения появления ошибок при преобразовании из-за неправильного ввода, а так же предотвращения ошибок из-за деления на ноль используется инструкция `try-except`.

Для вычисления всех значений функции в заданном интервале используется цикл `while`. Циклы позволяют выполнять блок кода до тех пор пока условие цикла не станет ложным.

Вывод данных выполняется инструкцией `print()`, с возможностью форматирования данных.

Задание 1

Описание программы

Программа написана на алгоритмическом языке Python 3.9.1, реализована в среде OS Windows 10 и состоит из частей, отвечающих за ввод данных, их преобразования к численному формату, вычисления и представления итоговых данных на экране монитора.

Описание алгоритма

1. Обернем все тело программы в `try - except` для контроля ошибок приведения и деления на ноль.
2. Ввести значения аргументов `xs xe st`, привести их к типу `float`.
3. Выводим шапку таблицы.
4. В цикле вычисляем(функцией `func()` из прошлой работы) и выводим получившееся значение функции, задав формат.
5. Увеличиваем счетчик цикла `xs` на шаг `st`.
6. Выводим последнюю строку таблицы.

Описание входных и выходных данных

Входные данные поступают с клавиатуры, а выходные – выводятся на монитор для просмотра. Входные имеют тип `float`. Выходные данные являются таблицей значений функции выведенной в терминал.

Листинг программы

```
from math import *

def func(x):
    if x <= -2:
        y = -x - 2
    if x > -2 and x < -1:
        y = sqrt(1 - pow(x + 1, 2))
    if x >= -1 and x <= 1:
        y = 1
    if x > 1 and x < 2:
        y = -2 * (x - 2) - 1
    if x >= 2:
        y = -1
    return y

try:
    xs = float(input("Input x start: "))
    xe = float(input("Input x end : "))
    st = float(input("Input step : "))

    print("      Xbeg = {} Xend = {} \n      Dx = {}".format(xs, xe, st))
    print("+-----+-----+")
    print("+      X      +      Y      +")
    print("+-----+-----+")
    while (xs <= xe):
        y = func(xs)
        print("I{0:8.2f}      I{1:8.2f}      I".format(xs, y))
        xs += st
    print("+-----+-----+")
except:
    print("Value error")
```

Результат работы программы

```
Input x start: -3
Input x end   : 3
Input step    : 0.5
      Xbeg = -3.0 Xend = 3.0
      Dx = 0.5
```

```
+-----+-----+
+      X      +      Y      +
+-----+-----+
I   -3.00   I   1.00   I
I   -2.50   I   0.50   I
I   -2.00   I   0.00   I
I   -1.50   I   0.87   I
I   -1.00   I   1.00   I
I   -0.50   I   1.00   I
I    0.00   I   1.00   I
I    0.50   I   1.00   I
I    1.00   I   1.00   I
I    1.50   I   0.00   I
I    2.00   I  -1.00   I
I    2.50   I  -1.00   I
I    3.00   I  -1.00   I
+-----+-----+
```

Задание 2

Описание программы

Программа написана на алгоритмическом языке Python 3.9.1, реализована в среде OS Windows 10 и состоит из частей, отвечающих за ввод данных, их преобразования к численному формату, вычисления и представления итоговых данных на экране монитора.

Описание алгоритма

1. Выводим шапку таблицы.
2. В цикле вычисляем(функцией `shot()` из прошлой работы) и выводим координаты и результат попадания.

Описание входных и выходных данных

Входные данные поступают с клавиатуры, а выходные – выводятся на монитор для просмотра. Входные имеют тип `float`. Выходные данные являются таблицей значений функции выведенной в терминал.

Листинг программы

```
from math import *
from random import *

def shot(r, x, y):
    if x >= 0 and y >= 0 and sqrt(x*x + y*y) <= r:
        return [x, y, 1]
    elif x <= 0 and y <= 0 and -x - r <= y:
        return [x, y, 1]
    else:
        return [x, y, 0]

r = 10
print ("      X          Y          RES")
print ("-----")
for i in range(10):
    x, y, res = shot(r, randrange(-r - 1, r + 1), randrange(-r - 1, r + 1))
    if res == 1: res = "YES"
    else: res = "NO"
    print("      {:3}          {:3}    {:3}".format(x, y, res))
```

Результат работы программы

X	Y	RES
4	9	YES
6	1	YES
7	4	YES
5	2	YES
0	6	YES
-6	-9	NO
-3	-3	YES
-1	1	NO
-6	-10	NO
9	7	NO

Задание 3

Описание программы

Программа написана на алгоритмическом языке Python 3.9.1, реализована в среде OS Windows 10 и состоит из частей, отвечающих за ввод данных, их преобразования к численному формату, вычисления и представления итоговых данных на экране монитора.

Описание алгоритма

1. Ввести значения аргументов `xs` `xe` `st` `es`, привести их к типу `float`.
2. Выводим шапку таблицы.
3. В цикле запускаем цикл для вычисления `Y` при заданном `X`.
4. Во вложенном цикле считаем следующий элемент последовательности и складываем с остальными, увеличиваем все счетчики.
5. Выводим получившиеся значения.
6. Увеличиваем счетчик цикла `xs` на шаг `st`.
7. Выводим последнюю строку таблицы.

Описание входных и выходных данных

Входные данные поступают с клавиатуры, а выходные – выводятся на монитор для просмотра. Входные имеют тип `float`. Выходные данные являются таблицей значений функции выведенной в терминал.

Листинг программы

```
from math import *

try:
    xs = float(input("Input x start : "))
    xe = float(input("Input x end   : "))
    st = float(input("Input dx      : "))
    es = float(input("Input eps     : "))
except:
    print("Value Error")
    exit()

print("+-----+-----+-----+-----+")
print("I      X      I      Y      I      N      I")
print("+-----+-----+-----+-----+")
while xs <= xe:
    if abs(xs) < 1:
        xs += st
        continue
    y = 0
    n = 1
    m = 0
    while True:
        de = 1 / (n * xs ** n)
        y += de
        n += 2
        m += 1
        if abs(de) < es:
            break
    print("I {:.10.3f}   I {:.10.3f}   I      {:7}      I".format(xs, y, m))
    xs += st
print("+-----+-----+-----+-----+")
```

Результат работы программы

```
Input x start : 1
Input x end   : 2
Input dx      : 0.2
Input eps     : 0.0001
+-----+-----+-----+-----+
I      X      I      Y      I      N      I
+-----+-----+-----+-----+
I      1.000   I      5.240   I      5001   I
I      1.200   I      1.199   I      17     I
I      1.400   I      0.896   I      10     I
I      1.600   I      0.733   I      8      I
I      1.800   I      0.626   I      7      I
I      2.000   I      0.549   I      6      I
+-----+-----+-----+-----+
```

Список используемой литературы

1. В.П. Рядченко, Методическое пособие по выполнению лабораторных работ
2. <https://pythonworld.ru/>