**Лабораторная работа №3:   
«Организация циклов»**

**Задание 1**

**Постановка задачи**

Вычислить и вывести на экран в виде таблицы значения функции, заданной графически (см. лабораторная работа № 2, задание 1), на интервале от Xнач до Xкон с шагом dx. Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

**Теоретическое введение**

Для решения задачи использована программа, подготовленная в лабораторной работе №2, задание 1 и оператор цикла с предусловием:

<Начальное значение>

**while** <Условие>:

<Инструкции>

<Приращение>

[**else**:

<Блок, выполняемый, если не использовался break>

]

Для обмена с консолью (вывод сообщений и ввод начальных данных) использованы стандартные процедуры print() и input(). Результаты работы программы записываются в текстовый файл.

**Описание алгоритма**

1. Ввести значения переменных Xbeg, Xend, dx.
2. Присвоить текущему значению Xt начальное значение: Xt = Xнач.
3. Вычислить значение функции и вывести в виде строки таблицы.
4. Вычислить новое значение аргумента Xt = Xt + Dx.
5. Если значение аргумента меньше Xend, то перейти к пункту 3.
6. Завершить рисование таблицы и работу программы.

**Описание входных и выходных данных**

В предшествующей работе был принят вещественный тип данных (real). В этой работе тип данных сохранён. Для упрощения последующего контроля работы программы в выходной текстовый файл записываются и начальные данные.

**Листинг программы**

from math import \*

print('Введите Xbeg, Xend и Dx')

xb = float(input('Xbeg='))

xe = float(input('Xend='))

dx = float(input('Dx='))

print("Xbeg={0: 7.2f} Xend={1: 7.2f}".format(xb, xe))

print(" Dx={0: 7.2f}".format(dx))

xt = xb

print("+--------+--------+")

print("I X I Y I")

print("+--------+--------+")

while xt <= xe:

if xt < -5:

y = 1

elif xt >=-5 and xt<0:

y = -(3/5)\*xt-2

elif xt >= 0 and xt<2:

y = -sqrt(4-xt\*\*2)

elif xt >= 2 and xt<4:

y = xt-2

elif xt >= 4 and xt<8:

y = 2+sqrt(4-(xt-6)\*\*2)

else: y = 2

print("I{0: 7.2f} I{1: 7.2f} I".format(xt, y))

xt += dx

print("+--------+--------+")

**Результат работы программы**

Xbeg= -10.00 Xend= 10.00

Dx= 1.00

+--------+--------+

I X I Y I

+--------+--------+

I -10.00I 1.00 I

I -9.00 I 1.00 I

I -8.00 I 1.00 I

I -7.00 I 1.00 I

I -6.00 I 1.00 I

I -5.00 I 1.00 I

I -4.00 I 0.40 I

I -3.00 I -0.20 I

I -2.00 I -0.80 I

I -1.00 I -1.40 I

I 0.00 I -2.00 I

I 1.00 I -1.73 I

I 2.00 I 0.00 I

I 3.00 I 1.00 I

I 4.00 I 2.00 I

I 5.00 I 3.73 I

I 6.00 I 4.00 I

I 7.00 I 3.73 I

I 8.00 I 2.00 I

I 9.00 I 2.00 I

I 10.00 I 2.00 I

+--------+--------+

# Задания к лабораторной работе №3 «Организация циклов». Задание 1

Вычислить и вывести на экран в виде таблицы значения функции, заданной графически (см. задание 1 лабораторной работы № 2), на интервале от Xнач до Xкон с шагом dx. Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

**Задание 2**

**Постановка задачи**

Для десяти выстрелов, координаты которых задаются генератором случайных чисел, вывести текстовые сообщения о попадании в мишень (см. лабораторная работа № 2, задание 2).

**Теоретическое введение**

Для решения задачи использована программа, подготовленная в лабораторной работе №2, задание 2 и оператор цикла с параметром:

**for** <Текущий элемент> **in** <Последовательность>:

<Инструкции внутри цикла>

[**else:**

<Этот код выполняется, если в теле цикла>

<не использовался *break*>]

Вывод сообщения выполняется стандартной функцией print().

Для формирования координат точки используется модуль генератора случайных чисел, который подключается инструкцией:

import random  
или

from random import \*

Для формирования случайного вещественного числа воспользуемся функцией

uniform(<Начало>,<Конец>)

Эта функция генерирует псевдослучайное число в диапазоне от <Начало> до <Конец>.

При этом правая граница не входит в интервал генерируемых значений (интервал открыт справа). В нашей задаче значения X формируются в диапазоне (-1, 4), а для Y – (-1, 10).

**Описание алгоритма**

1. Вывести "шапку".
2. В цикле от 1 до 10.
3. Сформировать координаты точки X, Y.
4. Определить попадание точки в заданную область. Если есть попадание, то переменная flag получает значение 1, а иначе – 0.
5. Вывести координаты точки и маркер оставить на строке сообщения.
6. Вывести результат Yes или No в соответствии со значением переменной flag.
7. Изменить параметр цикла и проверить условие завершения.
8. Если условие False, то перейти к п. 3.
9. Завершить работу программы.

**Описание входных и выходных данных**

Типы переменных, использованные в предыдущей работе, не изменялись. Для организации цикла введена новая переменная целого типа (int).

**Листинг программы**

from math import \*

from random import \*

flag = 0

print(" X Y Res")

print("-------------------")

for n in range(10):

x = uniform(-1, 4)

y = uniform(-1, 10)

if (x < -1) or (x > 4):

flag = 0 #False

if (((x>=-1) and (x< 1) and (y>=2\*x+2)

and (y<= x\*\*3-4\*x\*\*2+x+6))

or

((x>=1)and(x<=4)and(y>=x\*\*3-4\*x\*\*2+x+6)

and (y<= 2\*x+2))):

flag = 1

else:

flag = 0

print("{0: 7.2f} {1: 7.2f}".format(x, y), end=" ")

if flag:

print("Yes")

else:

print("No")

**Результат тестирования программы**

X Y Res

-------------------

-0.68 7.15 No

3.53 3.44 No

-0.05 4.02 Yes

-0.24 0.01 No

2.48 5.58 Yes

0.41 4.77 Yes

0.09 0.89 No

0.68 0.98 No

-0.03 5.46 Yes

-0.73 0.73 Yes

# Задания к лабораторной работе №3 «Организация циклов». Задание 2

Для десяти выстрелов, координаты которых задаются генератором случайных чисел, вывести текстовые сообщения о попадании в мишень (см. лабораторная работа № 2, задание 2).

**Задание 3**

**Постановка задачи**

Вычислить и вывести на экран в виде таблицы значения функции интегрального синуса, заданной с помощью степенного ряда, на интервале от Xнач до Xкон с шагом dx с точностью ε – эпсилон.

Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.

**Теоретическое введение**

При решении подобных задач, в которых имеется общая формула вычисления элемента ряда, очень полезно получить рекуррентную формулу. Такая формула позволяет упростить процесс программирования и ускоряет работу программы, так как получение следующего результата основывается на предыдущем. Особое внимание следует обратить на сходимость ряда. Если ряд расходится или сходится слабо, то программа может формировать сообщения о переполнении значений переменных или выводить неверный результат. Следует обращать внимание на границы допустимых значений аргумента.

1. Будем искать рекуррентное соотношение в виде выражения:  Получим выражение для *k*:

При выполнении преобразования выражение факториала в знаменателе было преобразовано к виду:



1. Для решения задачи нам потребуется два цикла. Первый цикл While (с предусловием), будет обеспечивать изменение значения переменной X от Xнач до Xкон с шагом dx. Второй цикл – это цикл с постусловием. Он нужен для итерационных вычислений элементов ряда с

В языке Python цикл с постусловием отсутствует. Для организации такого цикла воспользуемся циклом с предусловием в следующей форме:

**while** True:

<тело цикла>

**if** not <**условие>**:

**break**;

Для обмена с консолью (ввод/вывод) использованы стандартные функции input() и print().

**Описание алгоритма**

1. Ввести значения переменных Xнач, Xкон, dx и параметр точности ε.

2. Вывести "шапку" таблицы.

3. Инициировать Xt начальным значением (Xнач).

4. В цикле по Xt.

5. Инициировать переменную для подсчёта суммы членов ряда и переменную, которая отвечает за номер члена ряда (n).

6. В цикле по an.

7. Вычислить k, элемент ряда an, сумму элементов ряда и номер элемента.

8. Если модуль элемента ряда меньше ε, то прервать цикл (break) по an, иначе перейти к п.6.

9. Вывести строки таблицы: значение Xt, вычисленное значение функции и количество просуммированных членов ряда

10. Вычислить новое значение переменной Xt =Xt + dx.

11. Если значение аргумента меньше Xкон, то перейти к пункту 4.

12. Завершить рисование таблицы, и работу программы.

**Описание входных и выходных данных**

Поскольку тип переменных и точность представления не ограничены условием задачи, то входные переменные (Xнач, Xкон, dx и параметр точности ε) и вычисляемые значения аргумента и функции представляются переменными вещественного типа (float). Количество членов ряда подсчитывается переменной целого типа (int).

**Листинг программы**

from math import \*

print('Введите Xbeg, Xend, Dx и Eps')

xb = float(input('Xbeg='))

xe = float(input('Xend='))

dx = float(input('Dx='))

eps = float(input('Eps='))

print("+--------+--------+-----+")

print("I X I Y I N I")

print("+--------+--------+-----+")

xt = xb

while xt <= xe:

an = xt

n = 0

y = an

while True:

k=-(xt\*\*2)\*(2\*n+1)/((2\*n+2)\*(2\*n+3)\*\*2)

an = an\*k

y = y + an

n = n + 1

if abs(an) < eps:

break

print("I{0: 7.2f} I{1: 7.3f} I{2: 4} I".format(xt,y,n))

xt = xt + dx

print("+--------+--------+-----+")

**Результат работы программы**

Xnach= -4.00 Xkon= 6.00

Dx= 2.00 Eps= 0.00003

+--------+--------+-----+

I X I Y I N I

+--------+--------+-----+

I -4.00 I -1.758 I 8 I

I -2.00 I -1.605 I 5 I

I 0.00 I 0.000 I 1 I

I 2.00 I 1.605 I 5 I

I 4.00 I 1.758 I 8 I

I 6.00 I 1.425 I 10 I

+--------+--------+-----+

# Задание к лабораторной работе №3 "Организация циклов". Задание 3

Вычислить и вывести на экран монитора в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от Xнач до Xкон с шагом dx и точностью ε. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.





