

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

#### Цель работы:

Получить практический навык в использовании базовых конструкций структурного программирования – операторов цикла. Работа составлена из двух заданий.

#### Постановка задачи:

Вычислить и вывести на экран в виде таблицы значения функции, заданной графически (из лаб. раб. 2 задания 1) на интервале от  $X_{нач}$  до  $X_{кон}$  с шагом  $dx$ . Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

#### Теоретическая часть:

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

```
<Начальное значение>
while <условие>:
    <инструкции>
    <приращение>
[else:
    <блок, выполняемый, если не использовался break>
]
```

#### Описание программы:

Программа написана на алгоритмическом языке Python 3.6, реализована в среде ОС Windows 10 и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

#### Описание алгоритма:

1. Вести значение  $X_{beg}$ ,  $X_{end}$ ,  $Dx$ .
2. Присвоить текущему значению  $X_t$  начальное значение  $X_t = X_{нач}$
3. Вычислить значение функции и вывести в виде строки таблицы
4. Вычислить новое значение аргумента  $X_t = X_t + Dx$
5. Если значение аргумента меньше  $X_{end}$ , то перейти к пункту 3
6. Завершить рисование таблицы и работу программы

#### Описание входных и выходных данных:

Входные данные поступают с клавиатуры, а выходные – выводятся на монитор для просмотра. Входные и выходные данные имеют тип real.

#### Листинг программы:

```
from math import sqrt
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("|      X      |      Y      +")
print("+-----+-----+")
while xt <= xe:
    if xt>-2 and xt<0:
        y = sqrt(1-(xt+1)**2)
    elif 4>xt>0:
        y = -(sqrt (4-(xt-2)**2))
    elif 6>=xt>=4:
        y = 2-xt/2
    elif xt >6:
        y = -1
    elif xt<-2:
        y = (-xt)/3
    else :
        y = 0
    print("|{0: 7.2f} |{1: 7.2f} |". format(xt, y))
    xt+= dx
print("+-----+-----+")

```

Блок-схема алгоритма этого решения приведена в приложении 1 (рис 1) к лабораторной работе.

#### Результат работы программы:

X	Y
-3.00	1.00
-2.00	0.00
-1.00	1.00
0.00	0.00
1.00	-1.73
2.00	-2.00
3.00	-1.73
4.00	0.00
5.00	-0.50
6.00	-1.00
7.00	-1.00

## **Задание 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  $X_{нач}$  до  $X_{кон}$  с шагом  $dx$ .

Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить заголовком и шапкой.

### **Задание 2 Разбор задания**

#### **Постановка задачи:**

Для десяти выстрелов, координаты которых задаются генератором случайных чисел, вывести текстовое сообщение о попадании в мишень (из лабораторной работы №2 из задания 2).

#### **Теоретическая часть:**

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

```
for <текущий элемент> in <последовательность>:  
    <инструкция внутри цикла>  
[else:  
    <блок, выполняемый, если использовался break>  
]
```

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

```
import random  
  
или  
  
from random import*
```

Для формирования случайного вещественного числа воспользуемся функцией

```
uniform(<Начало>, <Конец>)
```

#### **Описание программы:**

Программа написана на алгоритмическом языке Python 3.6, реализована в среде ОС Windows 10 и состоит из частей, отвечающих за ввод данных, вычисление и представление данных на экране монитора.

#### **Описание алгоритма:**

1. Вывести «шапку»
2. В цикле от 1 до 10
3. Сформировать координаты X,Y
4. Определить попадание точки в заданную область. Если есть попадание, то переменная flag получает значени1, 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 3 пункту
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("-----")
r=2
xp=4
yp=4
xv=-4
yv=-4
for n in range(10):
    x=uniform(-6, 6)
    y=uniform(-6, 6)
    if ((-y >= - (abs(x))) and ((x-xv)*(x-xv)+(y-yv)*(y-yv))<=(r*r)) or ((y<=abs(x)) and ((x-xp)*(x-xp)+(y-yp)*(y-yp))<=(r*r))):
        flag = 1
    else:
        flag = 0
    print("{0: 7.2f} {1:7.2f}".format(x, y), end = "")
    if flag > 0:
        print(" yes")
    else:
        print (" no")
```

Блок-схема алгоритма этого решения приведена в приложении 1 (рис 1) к лабораторной работе.

#### Результат работы программы:

X	Y	Res
-----		
-2.53	-1.57	no
-3.43	4.49	no
-0.19	3.18	no
-5.99	2.42	no
-5.73	2.94	no
5.58	4.05	yes
-4.62	2.86	no
-3.58	-3.92	yes
3.56	-3.10	no
-1.60	5.30	no

## **Задание 2**

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