**Постановка задачи:** Написать скрипты, которые иллюстрируют работу кода в графическом виде:

− Задание 1 – Решить обратную задачу – построить график функции, используя программный код, написанный к этому заданию.

Листинг программы:

from math import \*  
import turtle as turtle  
  
  
# Функция кривых  
def fun\_graph(x):  
 if (x > 5) and (x < -9):  
 y = None  
 if (x >= -9) and (x < -5):  
 discr = 16 - 4 \* (x \*\* 2 + 14 \* x + 49)  
 if discr > 0:  
 y = (4 - sqrt(discr)) / 2  
 elif discr == 0:  
 y = 2  
 elif (x >= -5) and (x < -4):  
 y2 = 2  
 y = y2  
 elif (x >= -4) and (x < 0):  
 y3 = (-0.5) \* x  
 y = y3  
 elif (x >= 0) and (x < pi):  
 y4 = sin(x)  
 y = y4  
 elif (x >= pi) and (x <= 5):  
 y5 = x - pi  
 y = y5  
 else:  
 y = None  
 return y  
  
  
# Функция рисования оси  
def draw\_axes(txy, axy='X'):  
 turtle.penup()  
 if axy == 'X':  
 turtle.goto(txy[0], 0)  
 turtle.pendown()  
 turtle.goto(txy[1], 0)  
 else:  
 turtle.goto(0, txy[0])  
 turtle.pendown()  
 turtle.goto(0, txy[1])  
  
  
# Функция нанесения делений и надписей  
def draw\_mark(txy, axy='X'):  
 turtle.penup()  
  
 for i in range(txy[0], txy[1]):  
 if axy == 'X':  
 turtle.goto(i, 0)  
 turtle.pendown()  
 turtle.goto(i, 0.2)  
 turtle.penup()  
 turtle.goto(i, -0.5)  
 turtle.write(str(i))  
 else:  
 turtle.goto(0, i)  
 turtle.pendown()  
 turtle.goto(0.2, i)  
 turtle.penup()  
 turtle.goto(0.2, i)  
 turtle.write(str(i))  
  
  
# Функция рисования стрелки  
def draw\_arrow(txy, ax='X'):  
 # Параметры многоугольника  
 a = [0.1, 0, -0.1]  
 b = [-0.1, 0.3, -0.1]  
  
 turtle.up()  
 turtle.goto(0, 0) # в начало  
 turtle.begin\_poly() # начинаем запись вершин  
  
 for i in range(2): # для всех вершин  
 turtle.goto(a[i], b[i]) # многоугольника  
  
 turtle.end\_poly() # останавливаем запись  
 p = turtle.get\_poly() # ссылка на многоугольник  
  
 # регистрируем новую форму черепашке  
 turtle.register\_shape("myArrow", p)  
 turtle.resizemode("myArrow")  
 turtle.shapesize(1, 2, 1) # растягиваем (пример)  
  
 if ax == 'X': # для оси X  
 turtle.tiltangle(0) # угол для формы  
 turtle.goto(txy[1] + 0.2, 0) # к месту стрелки  
 pw = [int(txy[1]), -1.0] # надпись  
 else: # для оси Y  
 turtle.tiltangle(90) # угол для формы  
 turtle.goto(0, txy[1] + 0.2) # к месту стрелки  
 pw = [0.2, int(txy[1])] # надпись  
  
 turtle.stamp() # оставить штамп - стрелка  
  
 # надпишем ось  
 turtle.goto(pw) # к месту надписи  
 turtle.write(ax, font=("Arial", 14, "bold"))  
  
  
# Размеры по горизонтали и вертикали  
aX = [-9, 6]  
aY = [0, 3]  
  
# Размер главного окна  
Dx = 800  
Dy = Dx / ((aX[1] - aX[0]) / (aY[1] - aY[0]))  
turtle.setup(Dx, Dy)  
turtle.reset()  
  
# Число точек рисования для меньшей погрешности  
nMax = 1000  
  
# Создание системы координат  
turtle.setworldcoordinates(aX[0], aY[0], aX[1], aY[1])  
  
# Увеличение главного окна  
turtle.setup(Dx + 500, Dy + 500)  
  
# Ось - ox, метки, стрелка  
draw\_axes(aX, 'X')  
draw\_mark(aX, 'X')  
draw\_arrow(aX, 'X')  
  
# Ось - oy, метки, стрелка  
draw\_axes(aY, 'Y')  
draw\_mark(aY, 'Y')  
draw\_arrow(aY, 'Y')  
  
# Параметры  
turtle.color("blue") # цвет линии  
turtle.width(3) # толщина  
dx = (aX[1]-aX[0])/nMax # шаг  
  
# Начало координат  
x = aX[0]  
y = fun\_graph(x)  
if y is None:  
 turtle.up()  
 turtle.goto(x, 0)  
else:  
 turtle.goto(x, y)  
 turtle.down()  
  
# Рисование кривых  
while x <= aX[1]:  
  
 y = fun\_graph(x)  
 if y is None:  
 turtle.up()  
 turtle.goto(x, 0)  
 else:  
 turtle.goto(x, y)  
 turtle.down()  
 x = x + dx  
  
turtle.done()

Результаты выполнения:

