**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ**

кафедра фізико–технічних засобів захисту інформації

# ПРОГРАМУВАННЯ 4

**ЗВІТ ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №9**

**Виконала:**

Алієва Марія

Студентка 1 курсу ФТІ

групи ФЕ-81

Київ

2020

**ЗАВДАННЯ**

LOCATION(*РОЗМІЩЕННЯ*)

TPOINT (10 пікселів)

LINETO (пряма рухається і обертається)

POLYLINE (ламана рухається, обертається, змінює колір)

POLYGON (шестикутник рухається, обертається, змінює розмір)

**ВИКОНАННЯ РОБОТИ**

Код:

from tkinter import \*

import time

class Location:

def \_\_init\_\_(self, x=0.0, y=0.0):

self.x = x

self.y = y

def draw(self, canvas):

pass

def change\_all(self, canvas, offset):

pass

class TPoint(Location):

def \_\_init\_\_(self, x = 0.0, y = 0.0, width = 2.0):

Location.\_\_init\_\_(self, x, y)

self.width = width

class Line(Location):

def \_\_init\_\_(self, color = 'blue', point\_arr = None):

if point\_arr is None:

point\_arr = [TPoint(300, 10), TPoint(400.0, 100.0)]

self.first\_p = point\_arr[0]

self.color = color

self.second\_p = point\_arr[-1]

self.width = self.first\_p.width

self.line = None

def draw(self, canvas):

self.line = canvas.create\_line(self.first\_p.x, self.first\_p.y, self.second\_p.x, self.second\_p.y, width=self.width, fill=self.color)

def change\_all(self, canvas, offset):

canvas.coords(self.line, 300, 10, 350, 150)

x\_mov = offset.x - canvas.coords(self.line)[2]

y\_mov = offset.y - canvas.coords(self.line)[3]

canvas.itemconfig(self.line, fill="red")

canvas.move(self.line, x\_mov, y\_mov)

class PolyLine(Line):

def \_\_init\_\_(self, color='blue', point\_arr=None):

if point\_arr is None:

point\_arr = [TPoint(500, 10), TPoint(600.0, 100.0), TPoint(700.0, 100.0)]

Line.\_\_init\_\_(self, color, point\_arr)

self.point\_arr = point\_arr.copy()

self.color = color

self.line = []

def draw(self, canvas):

prev = self.point\_arr[0]

for i in range(1, len(self.point\_arr)):

self.line.append(canvas.create\_line(prev.x,

prev.y,

self.point\_arr[i].x,

self.point\_arr[i].y,

width=self.width,

fill=self.color))

prev = self.point\_arr[i]

def change\_all(self, canvas, offset):

canvas.coords(self.line[0], 500, 10, 550, 150)

canvas.coords(self.line[1], 550, 150, 650, 200)

for line in self.line:

canvas.itemconfig(line, fill="red")

canvas.move(line, offset.x, offset.y)

class PolyGon(Line):

def \_\_init\_\_(self, color='blue', point\_arr=None):

if point\_arr is None or len(point\_arr) < 5:

point\_arr = [TPoint(100.0, 100.0), TPoint(200.0, 100.0), TPoint(200.0, 200.0), TPoint(100.0, 200.0)]

Line.\_\_init\_\_(self, color, point\_arr)

self.point\_arr = point\_arr.copy()

self.line = []

def draw(self, canvas):

prev = self.point\_arr[0]

for i in range(1, 4):

self.line.append(canvas.create\_line(prev.x,

prev.y,

self.point\_arr[i].x,

self.point\_arr[i].y,

width=self.width,

fill=self.color))

prev = self.point\_arr[i]

self.line.append(canvas.create\_line(prev.x,

prev.y,

self.point\_arr[0].x,

self.point\_arr[0].y,

width=self.width,

fill=self.color))

def change\_all(self, canvas, offset):

canvas.coords(self.line[0], 100, 100, 150, 150)

canvas.coords(self.line[1], 150, 150, 100, 200)

canvas.coords(self.line[2], 100, 200, 50, 150)

canvas.coords(self.line[3], 50, 150, 100, 100)

for line in self.line:

canvas.itemconfig(line, fill="red")

canvas.move(line, offset.x, offset.y)

root = Tk()

c = Canvas(root, width=950, height=500, bg='white')

line = Line()

polyline = PolyLine()

polygon = PolyGon()

line.draw(c)

polyline.draw(c)

polygon.draw(c)

def line\_change():

line.change\_all(c, TPoint(600.0, 300.0))

def polyline\_change():

polyline.change\_all(c, TPoint(300.0, 200.0))

def polygon\_change():

polygon.change\_all(c, TPoint(600.0, 300.0))

b1 = Button(root, text="Change line", width=15, height=3, command=line\_change)

b2 = Button(root, text="Change polyline", width=15, height=3, command=polyline\_change)

b3 = Button(root, text="Change polygon", width=15, height=3, command=polygon\_change)

c.pack()

b1.pack()

b2.pack()

b3.pack()

root.mainloop()

Приклад виконання роботи:



