Міністерство освіти і науки, молоді та спорту України
Національний технічний університет України «Київський політехнічний інститут» Фізико-технічний інститут

Лабораторна робота з програмування No 9

Виконав:

Студент 2 курсу групи ФЕ-81

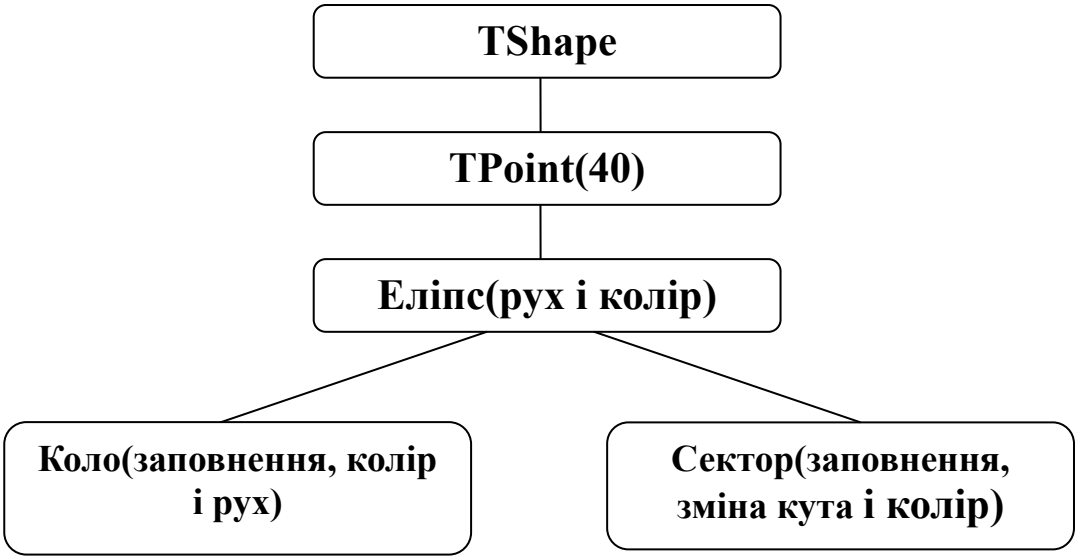
Кучер К. К.

Прогонов Д. О.

Київ 2020

Мета роботи:засвоїти базові поняття ООП на прикладі побудови ієрархії геометричних фігур засобами мови програмування Python.

Завдання



Код

```
from tkinter import *
import time


class TShape:
    def __init__(self, name):
        self.name = name

    def draw(self, canvas):
        print("Abstract class, cant draw")

    def move(self, canvas, dest):
        print("Abstract class, cant move")

    def delete(self, canvas):
        print("Abstract class, cant delete from canvas")


class TPoint (TShape):
    def __init__(self, x = 0.0, y= 0.0, width = 4.0):
        self.x = x
        self.y = y
        TShape.__init__(self, "Point")
        self.width = width


class Elips (TShape):
    def __init__(self, color = 'none', left_top = TPoint(10.0, 10.0), right_bottom =
TPoint(90.0, 50.0)):
        TShape.__init__(self, "Elips")
        self.left_top = left_top
        self.color = color
```

```
        self.right_bottom = right_bottom
        self.width = left_top.width
        self.pos_on_canvas = None

    def draw(self, canvas):
        print("Draw " + self.name)
        if self.color == 'none':
            self.pos_on_canvas = canvas.create_oval(self.left_top.x, self.left_top.y,
self.right_bottom.x, self.right_bottom.y, width = self.width)
        else:
            self.pos_on_canvas = canvas.create_oval(self.left_top.x, self.left_top.y,
self.right_bottom.x, self.right_bottom.y, width = self.width, fill = self.color)

    def move(self, canvas, dest):
        x_mov = dest.x - canvas.coords(self.pos_on_canvas)[2]
        y_mov = dest.y - canvas.coords(self.pos_on_canvas)[3]
        canvas.move(self.pos_on_canvas, x_mov, y_mov)

    def delete(self, canvas):
        canvas.delete(self.pos_on_canvas)



class Round (Elips):
    def __init__(self, radius, color = 'none', center = TPoint(100.0, 100.0)):
        left_top = TPoint(center.x + radius, center.y + radius)
        right_bottom = TPoint(center.x - radius, center.y - radius)
        Elips.__init__(self, color, left_top, right_bottom)
        TShape.__init__(self, "Round")



class Sector (Elips):
    def __init__(self, startangle, angle, color = 'none', left_top = TPoint(10.0, 10.0),
right_bottom = TPoint(100.0, 100.0)):
```

```
        self.angle = angle
        self.start_angle = startangle
        Elips.__init__(self, color, left_top, right_bottom)
        TShape.__init__(self, "Sector")

    def draw(self, canvas):
        print("Draw " + self.name)
        if self.color == 'none':
            self.pos_on_canvas = canvas.create_arc(self.left_top.x, self.left_top.y,
self.right_bottom.x, self.right_bottom.y, start = self.start_angle, extent = self.angle, width =
self.width)
        else:
            self.pos_on_canvas = canvas.create_arc(self.left_top.x, self.left_top.y,
self.right_bottom.x, self.right_bottom.y, start = self.start_angle, extent = self.angle, fill =
self.color, width = self.width)


#ROOT
root = Tk()

#ELEMS
sect = Sector(0, 90)
circle = Round(10, 'red')
elips = Elips('red')

#BOOLS
sect_spawned = False
elips_spawned = False
round_spawned = False

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

```
def spawn_elips():
    global elips
    global elips_spawned

    if elips_spawned:
        elips.delete(c)
        elips_spawned = False
        print(elips_spawned)
        return

    elips = Elips('red', TPoint(200.0, 200.0))
    elips.draw(c)
    elips_spawned = True


def spawn_round():
    global circle
    global round_spawned

    if round_spawned:
        circle.delete(c)
        round_spawned = False
        return
    circle = Round(10, 'red')
    circle.draw(c)
    round_spawned = True


def spawn_sector():
    global sect
    global sect_spawned

    if sect_spawned:
```

```
        sect.delete(c)
        sect_spawned = False
        return
    sect = Sector(0, 90)
    sect.draw(c)
    sect_spawned = True


def elips_move():
    elips.move(c, TPoint(600.0, 300.0))


def round_move():
    circle.move(c, TPoint(600.0, 300.0))


def sector_move():
    sect.move(c, TPoint(600.0, 300.0))

#BUTTON SETTINGS
b1 = Button(root, text="Spawn Elips", width=15, height=3, command=spawn_elips)
b2 = Button(root, text="Spawn Round", width=15, height=3, command=spawn_round)
b3 = Button(root, text="Spawn Sector", width=15, height=3, command=spawn_sector)
b4 = Button(root, text="Move Elips", width=15, height=3, command=elips_move)
b5 = Button(root, text="Move Round", width=15, height=3, command=round_move)
b6 = Button(root, text="Move Sector", width=15, height=3, command=sector_move)

#PACK SEGMENT
c.pack()
b1.pack(side = LEFT, padx=10)
b2.pack(side = LEFT, padx=10)
b3.pack(side = LEFT, padx=10)
b4.pack(side = LEFT, padx=10)
```

```
b5.pack(side = LEFT, padx=10)
b6.pack(side = LEFT, padx=10)

root.mainloop()
```