Міністерство освіти і науки України

НТУУ «Київський політехнічний інститут ім. Ігоря Сікорського»

Фізико-технічний інститут

# Програмування 4

# Лабораторна робота №9

«Побудова об’єктної ієрархії геометричних фігур»

**Виконав:**

Студент II курсу ФТІ групи ФЕ-81

Юрченко Денис

2020

1. Завдання лабораторної роботи

**TShape**

**TPoint(30)**

**Квадрат(рух і розмір)**

**Прямокутник(розмір і колір)**

**Ромб(колір і рух)**

1. Код реалізації

|  |
| --- |
| Lab\_8.py |
| from tkinter import \*  import random  class Tshape:  def \_\_init\_\_(self, canvas):  self.coords = None  self.ID = None  self.canvas = canvas  def draw(self):  pass  def delete(self):  pass  class Tpoint(Tshape):  def \_\_init\_\_(self, canvas, coords, color = 'black'):  Tshape.\_\_init\_\_(self, canvas)  self.can\_coords = coords  self.my\_coords = tuple()  self.\_get\_point\_coords()  self.color = color  def \_get\_point\_coords(self):  x, y = self.can\_coords  self.my\_coords = (x, y, x+5, y+5)  def draw(self):  self.ID = self.canvas.create\_rectangle(self.my\_coords, fill = self.color, outline = self.color)  def delete(self):  if self.ID != None:  self.canvas.delete(self.ID)  class Kvadrat(Tpoint):  def \_\_init\_\_(self, canvas, coords, size, color = 'black'):  self.size = size  self.coords = coords  self.points = list()  self.canvas = canvas  self.color = color  self.\_get\_points()  def \_get\_points(self):  x, y = self.coords  for i in range(0, self.size):  for k in range(0, self.size):  new\_point = Tpoint(self.canvas,(x, y), self.color)  self.points.append(new\_point)  x += 5  y += 5  x = self.coords[0]  def draw(self):  for p in self.points:  p.draw()  def delete(self):  for p in self.points:  p.delete()  self.points = list()  def action(self):  self.delete()  self.coords = (random.randint(2, 500), random.randint(2, 500))  self.size = random.randint(1, 15)  self.\_get\_points()  self.draw()  class Romb(Kvadrat):  def \_\_init\_\_(self, canvas, coords, size, color = 'black'):  Kvadrat.\_\_init\_\_(self, canvas, coords, size, color = 'yellow')  def \_get\_points(self):  x, y = self.coords  curr\_line = 0  curr\_r = 0  for k in range(0, self.size):  for i in range(k ,self.size):  new\_point1 = Tpoint(self.canvas,(x, y), self.color)  new\_point2 = Tpoint(self.canvas,(self.coords[0] - curr\_line, y), self.color)  new\_point3 = Tpoint(self.canvas,(x, self.coords[1] - curr\_r), self.color)  new\_point4 = Tpoint(self.canvas,(self.coords[0] - curr\_line, self.coords[1] - curr\_r), self.color)  y += 5  curr\_r += 5  self.points.append(new\_point1)  self.points.append(new\_point2)  self.points.append(new\_point3)  self.points.append(new\_point4)  x += 5  y = self.coords[1]  curr\_line += 5  curr\_r = 0  def draw(self):  self.\_get\_points()  for p in self.points:  p.draw()  def action(self):  self.color = random.choice(['black', 'green', 'blue', 'red', 'yellow'])  self.delete()  self.\_get\_points()  self.draw()  class Prkyt(Kvadrat):  def \_\_init\_\_(self, canvas, coords, size, color = 'blue'):  Kvadrat.\_\_init\_\_(self, canvas, coords, size, color = 'blue')  self.\_get\_points()  def \_get\_points(self):  x, y = self.coords  for i in range(0, self.size):  for k in range(0, self.size + 5):  new\_point = Tpoint(self.canvas,(x, y), self.color)  self.points.append(new\_point)  x += 5  y += 5  x = self.coords[0]  def action(self):  self.color = random.choice(['black', 'green', 'blue', 'red', 'yellow'])  self.size = random.randint(1, 15)  self.delete()  self.\_get\_points()  self.draw()  class P30:  def \_\_init\_\_(self, canvas):  self.points = list()  self.canvas = canvas  def create\_points(self, canvas):  if len(self.points) != 0:  for p in self.points:  p.delete()  self.points = list()  else:  for i in range(0, 30):  x = random.randint(2, 500)  y = random.randint(2, 500)  p = Tpoint(self.canvas,(x, y))  self.points.append(p)  p.draw()  rt = Tk()  rt.minsize(500, 500)  rt.maxsize(600, 600)  can = Canvas(rt, width = 500, height = 500)  can.pack(expand = YES, fill = BOTH)  p30 = P30(can)  b1 = Button(text = 'Create/delete 30 points ', command = lambda canvas= can: p30.create\_points(canvas))  b1.pack(side = LEFT)  kv = Kvadrat(can, (50, 50), 5, color = 'blue')  b2 = Button(text = 'Квадрат', command = kv.action)  b2.pack(side = LEFT)  rm = Romb(can, (50, 50), 5)  b3 = Button(text = 'Ромб', command = rm.action)  b3.pack(side = LEFT)  pr = Prkyt(can, (50, 100), 8)  b4 = Button(text = 'Прямокут.', command = pr.action)  b4.pack(side = LEFT)  def clear():  kv.delete()  rm.delete()  pr.delete()  b5 = Button(text = 'Clear', command = clear )  b5.pack(side = LEFT)  rt.mainloop() |

1. Виконання програми

Изображение выглядит как текст, большой, внешний, птица

Автоматически созданное описаниеИзображение выглядит как текст, птица, большой, стая

Автоматически созданное описание