

10 ago 02 11:23

geom3.py

Página 1/2

```
#!/usr/bin/python
# LEG10082002
# Cargar rutinas gráficas
from Tkinter import *
root = Tk() #Crear ventana principal llamada "root"
# Preparar una ventana de lienzo
scale = 5
canvas = Canvas(root, background = "white", height = 100*scale, width = 100*scale
)
canvas.pack()

# Nuestras clases

class punto:
    "Objeto gráfico - Punto"
    x = None
    y = None
    widget = None

    def __init__(self, x, y):
        self.widget = canvas.create_oval(-1, -1, 1, 1)
        self.set(x, y)
        self.draw()

    def set(self, x, y):
        self.x = x
        self.y = y

    def draw(self):
        canvas.coords(self.widget, self.x*scale-1,
                       self.y*scale-1,
                       self.x*scale+1,
                       self.y*5+1)

class linea:
    a = None
    b = None
    widget = None

    def __init__(self, comienzo, final):
        self.a = comienzo
        self.b = final
        self.widget = canvas.create_line(self.a.x, self.a.y,
                                         self.b.x, self.b.y)
        self.draw()

    def draw(self):
        self.a.draw()
        self.b.draw()
        canvas.coords(self.widget, self.a.x*scale, self.a.y*scale,
                       self.b.x*scale, self.b.y*scale)

# Demostración de la clase "Punto"
a = punto(10, 10)
b = punto(10, 90)
c = punto(90, 10)
d = punto(90, 90)
```

```
# Demostración de la clase "Línea"  
l = linea(b, c)  
# Entrar en bucle de Eventos  
root.title("Geometria: la clase Línea")  
root.mainloop()
```

10 ago 02 11:44

geom5.py

Página 1/2

```
#!/usr/bin/python
# LEG10082002
# Cargar rutinas gráficas
from Tkinter import *
root = Tk() #Crear ventana principal llamada "root"
# Preparar una ventana de lienzo
scale = 5
canvas = Canvas(root, background = "white", height = 100*scale, width = 100*scale
)
canvas.pack()

# Nuestras clases

class punto:
    "Objeto gráfico - Punto"
    x = None
    y = None
    widget = None

    def __init__(self, x, y):
        self.widget = canvas.create_oval(-1, -1, 1, 1)
        self.set(x, y)
        self.draw()

    def set(self, x, y):
        self.x = x
        self.y = y

    def draw(self):
        canvas.coords(self.widget, self.x*scale-1,
                       self.y*scale-1,
                       self.x*scale+1,
                       self.y*scale+1)

    def move(self, dx, dy):
        self.set(self.x+dx, self.y+dy)

class linea:
    a = None
    b = None
    widget = None

    def __init__(self, comienzo, final):
        self.a = comienzo
        self.b = final
        self.widget = canvas.create_line(self.a.x, self.a.y,
                                         self.b.x, self.b.y)

        self.draw()

    def draw(self):
        self.a.draw()
        self.b.draw()
        canvas.coords(self.widget, self.a.x*scale, self.a.y*scale,
                       self.b.x*scale, self.b.y*scale)

    def move(self, dx, dy):
        self.a.move(dx, dy)
        self.b.move(dx, dy)
```

```
self.draw()

class trapezoide:
    "Trapezoide: construido a partir de los cuatro vértizes"
    l1 = None
    l2 = None
    l3 = None
    l4 = None

    def __init__(self, a, b, c, d):
        self.l1 = linea(a, b)
        self.l2 = linea(b, c)
        self.l3 = linea(c, d)
        self.l4 = linea(d, a)
        self.draw()

    def draw(self):
        self.l1.draw()
        self.l2.draw()
        self.l3.draw()
        self.l4.draw()

    def move(self, dx, dy):
        self.l1.move(dx, dy)
        self.l3.move(dx, dy)
        self.draw()

# Demostración de la clase "Trapezoide"
a = punto(10, 10)
b = punto(20, 90)
c = punto(80, 30)
d = punto(70, 70)

t = trapezoide(a, b, d, c)

t.move(-10, -10)

# Entrar en bucle de Eventos

root.title("Geometria: Mover un Trapezoide")

root.mainloop()
```

10 ago 02 12:06

geom6.py

Página 1/2

```
#!/usr/bin/python
# LEG10082002
# Cargar rutinas gráficas
from Tkinter import *
root = Tk() #Crear ventana principal llamada "root"
# Preparar una ventana de lienzo
scale = 5
canvas = Canvas(root, background = "white", height = 100*scale, width = 100*scale
)
canvas.pack()

# Nuestras clases

class punto:
    "Objeto gráfico - Punto"
    x = 0
    y = 0
    widget = 0

    def __init__(self, x, y):
        self.widget = canvas.create_oval(-1, -1, 1, 1)
        self.set(x, y)
        self.draw()

    def set(self, x, y):
        self.x = x
        self.y = y

    def draw(self):
        canvas.coords(self.widget, self.x*scale-1,
                       self.y*scale-1,
                       self.x*scale+1,
                       self.y*5+1)

class linea:
    a = 0
    b = 0
    widget = 0

    def __init__(self, comienzo, final):
        self.a = comienzo
        self.b = final
        self.widget = canvas.create_line(self.a.x, self.a.y,
                                         self.b.x, self.b.y)
        self.draw()

    def draw(self):
        self.a.draw()
        self.b.draw()
        canvas.coords(self.widget, self.a.x*scale, self.a.y*scale,
                       self.b.x*scale, self.b.y*scale)

class trapezoide:
    "Trapezoide: construido a partir de los cuatro vértizes"
    l1 = None
    l2 = None
```

```
l3 = None
l4 = None

def __init__(self, a, b, c, d):
    self.l1 = linea(a, b)
    self.l2 = linea(b, c)
    self.l3 = linea(c, d)
    self.l4 = linea(d, a)
    self.draw()

def draw(self):
    self.l1.draw()
    self.l2.draw()
    self.l3.draw()
    self.l4.draw()

class rombo(trapezoide):
    "Rombo, un trapezoide, con el cuarto punto restringido"

    def __init__(self, a, b, c):
        d = punto(c.x - (b.x-a.x), c.y - (b.y-a.y))
        trapezoide(a, b, c, d)

# Demostración de la clase "Trapezoide"
a = punto(10, 10)
b = punto(20, 90)
c = punto(80, 30)
d = punto(70, 70)

t = trapezoide(a, b, d, c)

x = punto(40, 30)

r = rombo(d, b, x)

# Entrar en bucle de Eventos

root.title("Geometria: la clase Rombo – herencia de los Trapezoides")

root.mainloop()
```