Chapter 14 , {exercises}

شایسته گیوه ای

Listing 14.1: comptrafficlight.py

```
import tkinter as tk
import tkinter.ttk as ttk

class CompLamp:
    """ Serves as one lamp within a traffic light object. """

    def init(self, parent, width, order, color="red", *args, **kwargs):
        """ Creates a new lamp to be used in a traffic light object.
         parent: The traffic light owning this lamp
        width: The width of the case of the circular lamp
        order: Distance of this lamp from the top of the traffic light
         color: The lamp's initial color (defaults to "red")
        *args: Additional arguments to pass to the ttk.Frame
       superclass constructor
      **kwargs: Additional keyword arguments to pass to the
           ttk.Frame superclass constructor """
```

```
self.frame = ttk.Frame(parent.frame, *args, **kwargs)
self.canvas = tk.Canvas(self.frame, width=width, height=width,
bg="gray",
                              highlightthickness=0)
self.canvas.pack()
self.color = color
offset = width//8
self.lamp = self.canvas.create_oval(offset, offset,
                  7*offset,
                  7*offset,
                  fill='black')
self.frame.grid(row=order, column=0)
self.state = "off"
   def turn_on(self):
         """ Illuminates the lamp """
self.state = "on"
self.canvas.itemconfigure(self.lamp, fill=self.color)
    def turn_off(self):
        """ Turns off the lamp """
self.state = "off"
self.canvas.itemconfigure(self.lamp, fill='black')
    def resize(self, width):
```

```
self.canvas.config(width=width, height=width)
offset = width//8
self.canvas.coords(self.lamp, offset, offset, 7*offset, 7*offset)
    class CompTrafficLight:
        """ Models a simple traffic light widget """
    def init(self, root, wd, initial_color="red", *args, **kwargs):
        """ Makes a new traffic light object.
root is the parent widget.
wd is the pixels width.
The light's initial color is initial_color.
Clients may pass additional arguments to the constructor of the
light's frame via *args and **kwargs. ""
if initial_color not in ("red", "yellow", "green"):
          raise ValueError(initial_color + " is not a valid color")
self.frame = ttk.Frame(root, width=wd, *args, **kwargs)
self.frame.grid(row=0, column=0)
self.color = initial_color
self.lamps = dict(zip(('red', 'yellow', 'green'),
(CompLamp(self, wd, 0, 'red'),
CompLamp(self, wd, 1, 'yellow'),
CompLamp(self, wd, 2, 'green'))))
self.lamps[self.color].turn_on()
```

```
    def change(self):
        """ Changes the traffic light's color to the next color in   the
sequence. """
if self.color == 'red':
       new_color = 'green'
elif self.color == 'green':
       new_color = 'yellow'
elif self.color == 'yellow':
      new_color = 'red'

self.lamps[self.color].turn_off()
self.color = new_color
self.lamps[self.color].turn_on()
def resize(self, width):
            """ Changes the traffic light's frame width according to the
parameter passed by the caller. """
for lamp in self.lamps.values():
lamp.resize(width)
```