

# OO Grafik mit wxPython

Erläuterung von Teilen des Programmcodes

# OO Grafik mit wxPython

- Kopf und Konstruktor

```
class TestPanel(wx.Panel):  
    def __init__(self, parent):  
        wx.Panel.__init__(self, parent, -1)  
        self.Bind(wx.EVT_PAINT, self.OnPaint)  
        if USE_BUFFER:  
            self.Bind(wx.EVT_SIZE, self.OnSize)
```

Frame ist "parent"

# OO Grafik mit wxPython

- Kopf und Konstruktor

```
class TestPanel(wx.Panel):  
    def __init__(self, parent):  
        wx.Panel.__init__(self, parent, -1)  
        self.Bind(wx.EVT_PAINT, self.OnPaint)  
        if USE_BUFFER:  
            self.Bind(wx.EVT_SIZE, self.OnSize)
```

Ereignis an Methode binden

# OO Grafik mit wxPython

- Kopf und Konstruktor

```
class TestPanel(wx.Panel):  
    def __init__(self, parent):  
        wx.Panel.__init__(self, parent, -1)  
        self.Bind(wx.EVT_PAINT, self.OnPaint)  
        if USE_BUFFER:  
            self.Bind(wx.EVT_SIZE, self.OnSize)
```

"buffered drawing" an Methode binden

# OO Grafik mit wxPython

- Methode OnSize(...)

```
def OnSize(self, evt):  
    self.InitBuffer()  
    evt.Skip()
```

Methode dazu

# OO Grafik mit wxPython

- Methode OnPaint(...)

```
def OnPaint(self, evt):  
    if USE_BUFFER:  
        dc = wx.BufferedPaintDC(self, self._buffer)  
    else:  
        dc = wx.PaintDC(self)  
        gc = self.MakeGC(dc)  
        self.Draw(gc)
```

dc: "drawing context"

# OO Grafik mit wxPython

- Methode OnPaint(...)

```
def OnPaint(self, evt):  
    if USE_BUFFER:  
        dc = wx.BufferedPaintDC(self, self._buffer)  
    else:  
        dc = wx.PaintDC(self)  
        gc = self.MakeGC(dc)  
        self.Draw(gc)
```

gc: "graphics context"

# OO Grafik mit wxPython

- Methode OnPaint(...)

```
def OnPaint(self, evt):  
    if USE_BUFFER:  
        dc = wx.BufferedPaintDC(self, self._buffer)  
    else:  
        dc = wx.PaintDC(self)  
        gc = self.MakeGC(dc)  
        self.Draw(gc)
```

Zeichenmethode aufrufen



# OO Grafik mit wxPython

- Die Methoden
  - `InitBuffer(self)`  
und
  - `MakeGC(self, dc)`sind hier nicht erläutert.

# OO Grafik mit wxPython

- Die Methode Draw(...)

```
def Draw(self, gc):  
    font = wx.SystemSettings.GetFont(  
        wx.SYS_DEFAULT_GUI_FONT)  
    font.SetWeight(wx.BOLD)  
    gc.SetFont(font)  
    ...
```

Zeichensatz festlegen

# OO Grafik mit wxPython

Zeichenpfad definieren

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    path = gc.CreatePath()  
    path.AddCircle(0, 0, BASE2)  
    path.MoveToPoint(0, -BASE2)  
    path.AddLineToPoint(0, BASE2)  
    path.MoveToPoint(-BASE2, 0)  
    path.AddLineToPoint(BASE2, 0)  
    path.CloseSubpath()  
    path.AddRectangle(-BASE4, -BASE4/2,  
BASE2, BASE4)
```

# OO Grafik mit wxPython

Figur mit Kreis beginnen

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    path = gc.CreatePath()  
    path.AddCircle(0, 0, BASE2)  
    path.MoveToPoint(0, -BASE2)  
    path.AddLineToPoint(0, BASE2)  
    path.MoveToPoint(-BASE2, 0)  
    path.AddLineToPoint(BASE2, 0)  
    path.CloseSubpath()  
    path.AddRectangle(-BASE4, -BASE4/2,  
BASE2, BASE4)
```

# OO Grafik mit wxPython

BASE, BASE2, BASE4:  
globale Variable

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    path = gc.CreatePath()  
    path.AddCircle(0, 0, BASE2)  
    path.MoveToPoint(0, -BASE2)  
    path.AddLineToPoint(0, BASE2)  
    path.MoveToPoint(-BASE2, 0)  
    path.AddLineToPoint(BASE2, 0)  
    path.CloseSubpath()  
    path.AddRectangle(-BASE4, -BASE4/2,  
BASE2, BASE4)
```

# OO Grafik mit wxPython

Kreuz hinzufügen

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    path = gc.CreatePath()  
    path.AddCircle(0, 0, BASE2)  
    path.MoveToPoint(0, -BASE2)  
    path.AddLineToPoint(0, BASE2)  
    path.MoveToPoint(-BASE2, 0)  
    path.AddLineToPoint(BASE2, 0)  
    path.CloseSubpath()  
    path.AddRectangle(-BASE4, -BASE4/2,  
BASE2, BASE4)
```

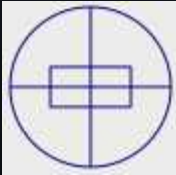
# OO Grafik mit wxPython

Rechteck hinzufügen

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    path = gc.CreatePath()  
    path.AddCircle(0, 0, BASE2)  
    path.MoveToPoint(0, -BASE2)  
    path.AddLineToPoint(0, BASE2)  
    path.MoveToPoint(-BASE2, 0)  
    path.AddLineToPoint(BASE2, 0)  
    path.CloseSubpath()  
    path.AddRectangle(-BASE4, -BASE4/2,  
BASE2, BASE4)
```

# OO Grafik mit wxPython

- Die Methode Draw definiert zunächst nur den Pfad für die Figur.
- Wie und wo sie gezeichnet wird, wird dann erst anschließend vorgegeben.
- Dafür gibt es in dieser Demo viele Varianten.
- Für die erste Figur  würde der folgende Code reichen:



# OO Grafik mit wxPython

Neudefinition des Ursprungs

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    gc.Translate(60, 75)  
    gc.SetPen(wx.Pen("navy", 1))  
    gc.StrokePath(path)
```

# OO Grafik mit wxPython

"Stift", also Zeichenfarbe setzen

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    gc.Translate(60, 75)  
    gc.SetPen(wx.Pen("navy", 1))  
    gc.StrokePath(path)
```

# OO Grafik mit wxPython

StrokePath zeichnet nur  
den Umriss der Figur

- Die Methode Draw(...)

```
def Draw(self, gc):  
    ...  
    gc.Translate(60, 75)  
    gc.SetPen(wx.Pen("navy", 1))  
    gc.StrokePath(path)
```