

Sessel mit GraphicsPath

Lösung mit Rechtecken

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path =  
    Zeichenflaeche.GibZeichenflaeche().GibGC().CreatePath()  
    path.AddRectangle(0, 0, self.b, self.t)  
  
    path.AddRecta  
  
    path.AddRectang  
  
    path.AddRectang  
  
gc = ...
```

Der Zeichenpfad
wird von der
Klasse Zeichenflaeche
geholt.

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...
```

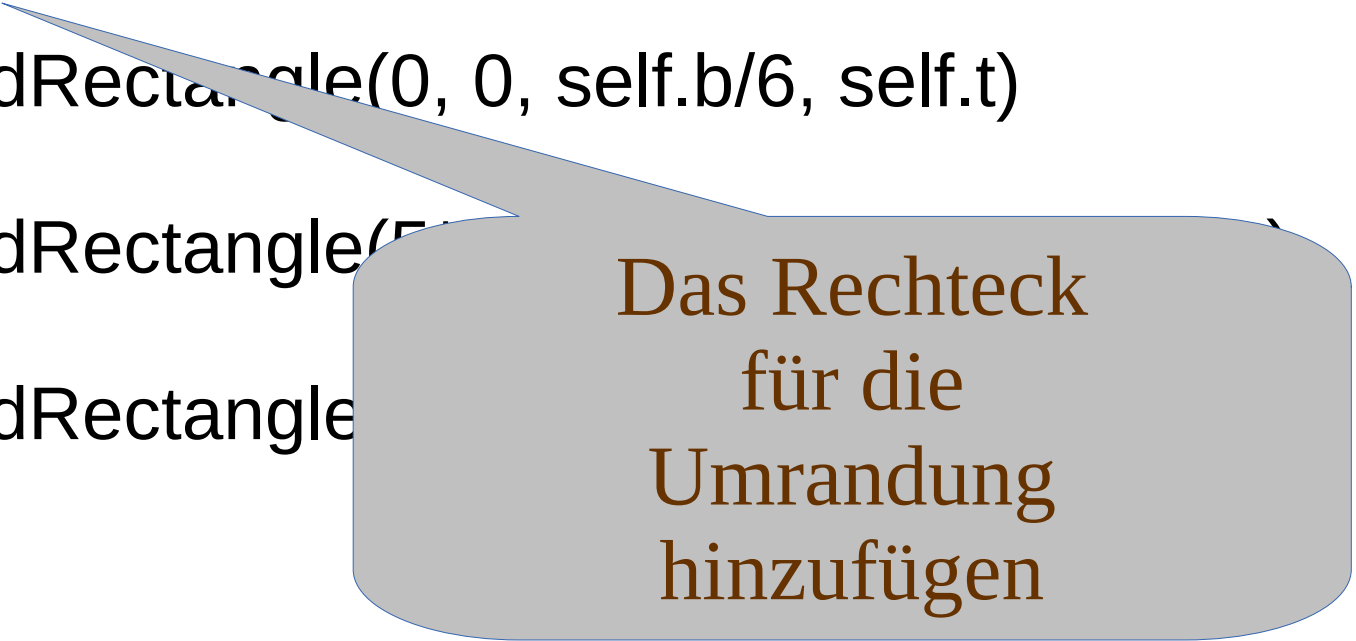
```
path.AddRectangle(0, 0, self.b, self.t)
```

```
path.AddRectangle(0, 0, self.b/6, self.t)
```

```
path.AddRectangle(5, ...)
```

```
path.AddRectangle
```

```
gc = ...
```



Das Rechteck
für die
Umrandung
hinzufügen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...
```

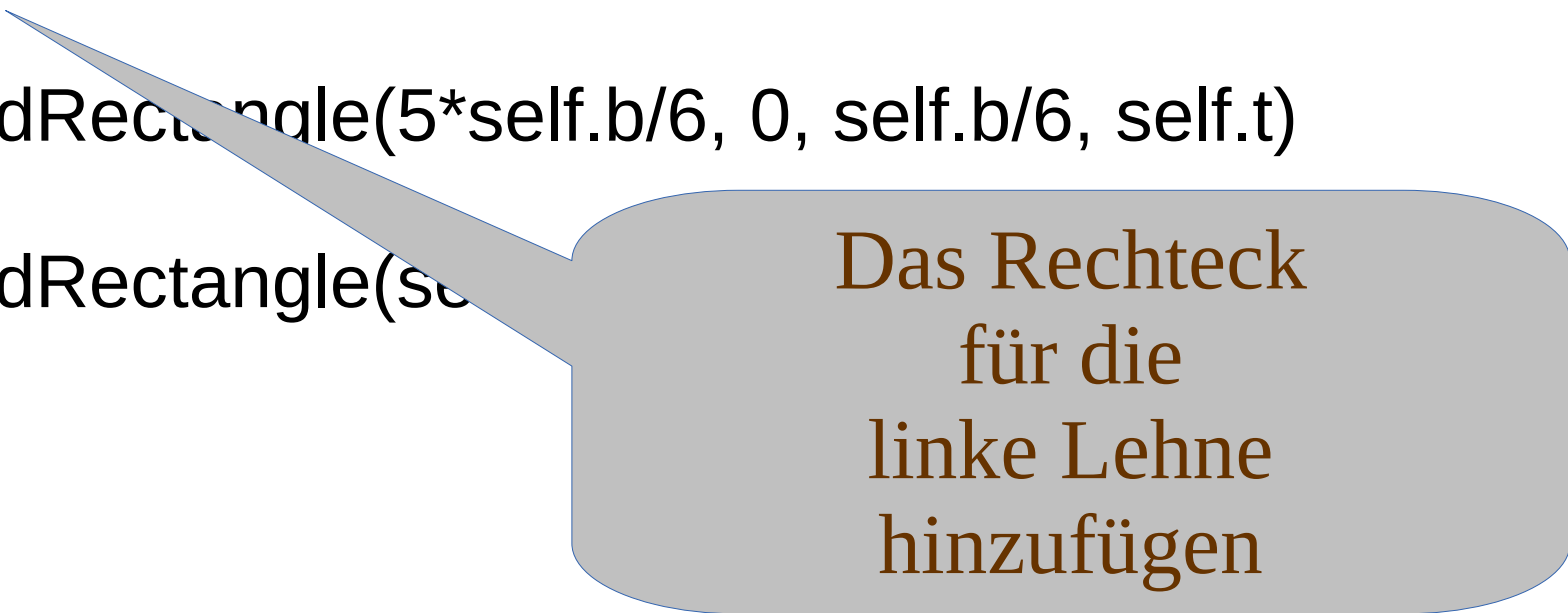
```
    path.AddRectangle(0, 0, self.b, self.t)
```

```
    path.AddRectangle(0, 0, self.b/6, self.t)
```

```
    path.AddRectangle(5*self.b/6, 0, self.b/6, self.t)
```

```
    path.AddRectangle(self.b, 0, self.b, self.t)
```

```
    gc = ...
```



Das Rechteck
für die
linke Lehne
hinzufügen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
  
    path.AddRectangle(0, 0, self.b, self.t)  
  
    path.AddRectangle(0, 0, self.b/6, self.t)  
  
    path.AddRectangle(5*self.b/6, 0, self.b/6, self.t)  
  
    path.AddRectangle(5*self.b/6, 0, self.b/6, self.t)  
  
    gc = ...
```

Das Rechteck
für die
rechte Lehne
hinzufügen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...
```

```
    path.AddRectangle(0, 0, self.l, self.t)
```

```
    path.AddRectangle(0, 0, s
```

```
    path.AddRectangle(0, 0, s
```

Das Rechteck
für die
Rück-Lehne
hinzufügen

```
path.AddRectangle(self.b/6, 0, 4*self.b/6, self.t/6)
```

```
    gc = ...
```

Sessel mit GraphicsPath

Lösung mit Linien

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.b, 0)  
    path.AddLineToPoint(self.b, self.t)  
    path.AddLineToPoint(0, self.t)  
    path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, 0)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
    path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t/6)  
    gc = ...
```

Ohne zu zeichnen
zum Startpunkt

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.b, 0)  
    path.AddLineToPoint(self.b, self.t)  
    path.AddLineToPoint(0, self.t)  
    path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, 0)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
    path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t/6)  
    gc = ...
```

Linie zur
rechten oberen
Ecke zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.b, 0)  
    path.AddLineToPoint(self.b, self.t)  
    path.AddLineToPoint(0, self.t)  
    path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, 0)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
    path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t/6)  
    gc = ...
```

Linie zur
rechten unteren
Ecke zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.b, 0)  
    path.AddLineToPoint(self.b, self.t)  
    path.AddLineToPoint(0, self.t)  
    path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, 0)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
    path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t/6)  
    gc = ...
```

Linie zur
linken unteren
Ecke zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.b, 0)  
    path.AddLineToPoint(self.b, self.t)  
    path.AddLineToPoint(0, self.t)  
path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, 0)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
    path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t/6)  
    gc = ...
```

Linie zur
linken oberen
Ecke zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.  
    path.AddLineToPoint(self.  
    path.AddLineToPoint(0  
    path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, 0)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
    path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t/6)  
    gc = ...
```

linke Lehne:
zur rechten oberen
Ecke ohne Zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):
    path = ...
    path.MoveToPoint(0, 0)
    path.AddLineToPoint(self.b/6, 0)
    path.AddLineToPoint(self.b/6, self.t)
    path.AddLineToPoint(5*self.b/6, self.t)
    path.AddLineToPoint(5*self.b/6, 0)
    path.MoveToPoint(0, 0)
    path.MoveToPoint(self.b/6, 0)
    path.AddLineToPoint(self.b/6, self.t)
    path.MoveToPoint(5*self.b/6, 0)
    path.AddLineToPoint(5*self.b/6, self.t)
    path.MoveToPoint(self.b/6, self.t/6)
    path.AddLineToPoint(5*self.b/6, self.t/6)
    gc = ...
```

linke Lehne:
zur rechten unteren
Ecke zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):
    path = ...
    path.MoveToPoint(0, 0)
    path.AddLineToPoint(self.
    path.AddLineToPoint(self.
    path.AddLineToPoint(0
    path.AddLineToPoint(0, 0)
    path.MoveToPoint(self.b/6, 0)
    path.AddLineToPoint(self.b/6, self.t)
    path.MoveToPoint(5*self.b/6, 0)
    path.AddLineToPoint(5*self.b/6, self.t)
    path.MoveToPoint(self.b/6, self.t/6)
    path.AddLineToPoint(5*self.b/6, self.t/6)
    gc = ...
```

rechte Lehne:
zur linken oberen
Ecke ohne Zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.AddLineToPoint(0, self.t)  
    path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t/6)  
    gc = ...
```

rechte Lehne:
zur rechten unteren
Ecke zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(self.b/6, self.t)  
    path.AddLineToPoint(0, self.t)  
    path.AddLineToPoint(0, 0)  
    path.MoveToPoint(self.b/6, self.t/6)  
    path.AddLineToPoint(5*self.b/6, self.t)  
    path.MoveToPoint(5*self.b/6, 0)  
    path.AddLineToPoint(5*self.b/6, self.t)  
path.MoveToPoint(self.b/6, self.t/6)  
    path.MoveToPoint(self.b/6, self.t/6)  
    gc = ...
```

Rück-Lehne:
zur linken unteren
Ecke ohne Zeichnen

Sessel mit GraphicsPath

```
def GibFigur(self):
    path = ...
    path.MoveToPoint(0, 0)
    path.AddLineToPoint(self.b/6, self.t)
    path.AddLineToPoint(self.b/6, 0)
    path.AddLineToPoint(0, self.t)
    path.AddLineToPoint(0, 0)
    path.MoveToPoint(self.b/6, self.t)
    path.AddLineToPoint(5*self.b/6, self.t)
    path.MoveToPoint(5*self.b/6, 0)
    path.AddLineToPoint(self.b/6, 0)
    path.AddLineToPoint(5*self.b/6, self.t)
    path.MoveToPoint(self.b/6, self.t/6)
    path.AddLineToPoint(5*self.b/6, self.t/6)
    gc = ...
```

Rück-Lehne:
zur rechten unteren
Ecke zeichnen

Sessel mit GraphicsPath

Positionieren und Drehen

*Der folgende Abschnitt ist nicht notwendig für das Verständnis von Modellierung und Programmierung.
Es wird erläutert, wie das entwickelte Element der Vektorgrafik gezielt dargestellt wird.*

Sessel mit GraphicsPath

```
def GibFigur(self):
    path = ...
    ...
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
    gc.PushState()
    gc.Translate(self.x+self.b/2, self.y+self.t/2)
    gc.Rotate(radians(self.w))
    gc.Translate(-self.b/2, -self.t/2)
    transformation = gc.GetTransform()
    gc.PopState()
    path.Transform(transformation)
    return path
```

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...
```

```
    ...
```

```
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
```

```
    gc.PushState()
```

```
    gc.Translate(self.x+self.b/2, self.y+self.t/2)
```

```
    gc.Rotate(radians(self.w))
```

```
    gc.Translate(-self.b/2, -self.t/2)
```

```
    transformation = gc.GetTransform()
```

```
    gc.PopState()
```

```
    path.Transform(transformation)
```

```
    return path
```

Es wird in den
„graphics context“
gezeichnet

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...
```

```
    ...
```

```
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
```

```
    gc.PushState()
```

```
    gc.Translate(self.x+self.b/2, self.y+self.t/2)
```

```
    gc.Rotate(radians(self.w))
```

```
    gc.Translate(-self.b/2, -self.t/2)
```

```
    transformation = gc.GetTransform()
```

```
    gc.PopState()
```

```
    path.Transform(transformation)
```

```
    return path
```

Die Klasse Zeichenflaeche stellt ihn bereit mit GibGC()

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    ...  
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()  
    gc.PushState()  
    gc.Translate(self.x+self.b/2, self.y+self.t/2)  
    gc.Rotate(radians(self.w))  
    gc.Translate(-self.b/2, -self.t/2)  
    transformation = gc.GetTransform()  
    gc.PopState()  
    path.Transform(transformation)  
    return path
```

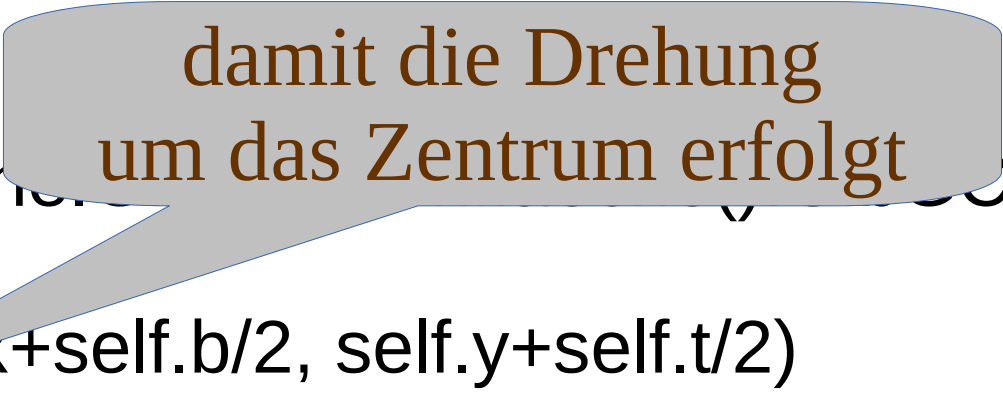
Zustand sichern

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    ...  
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()  
    gc.PushState()  
    gc.Translate(self.x+self.b/2, self.y+self.t/2)  
    gc.Rotate(radians(self.w))  
    gc.Translate(-self.b/2, -self.t/2)  
    transformation = gc.GetTransform()  
    gc.PopState()  
    path.Transform(transformation)  
    return path
```

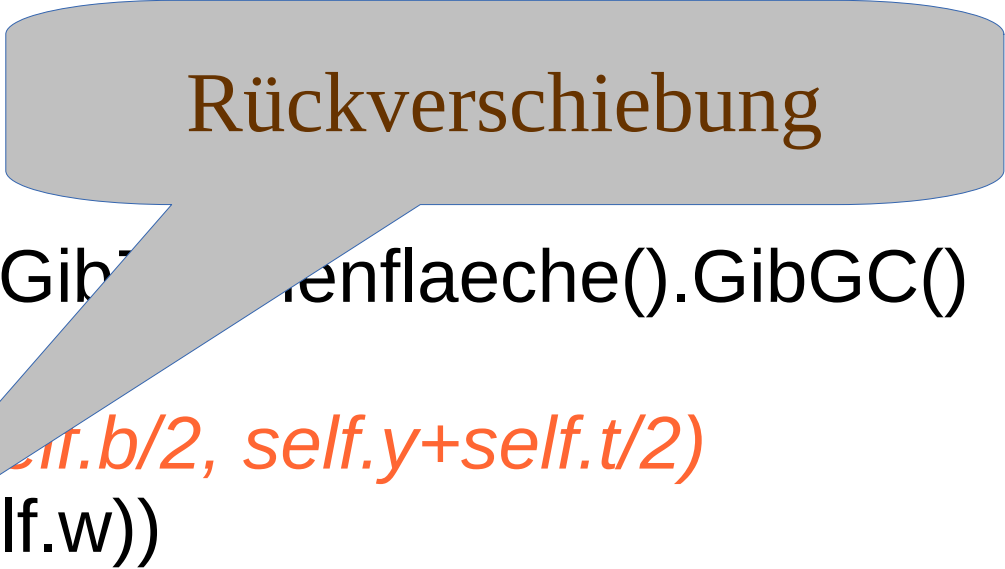
verschieben auf
das Zentrum der Figur

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    ...  
    gc = Zeichenflaeche()   
    gc.PushState()  
    gc.Translate(self.x+self.b/2, self.y+self.t/2)  
    gc.Rotate(radians(self.w))  
    gc.Translate(-self.b/2, -self.t/2)  
    transformation = gc.GetTransform()  
    gc.PopState()  
    path.Transform(transformation)  
    return path
```

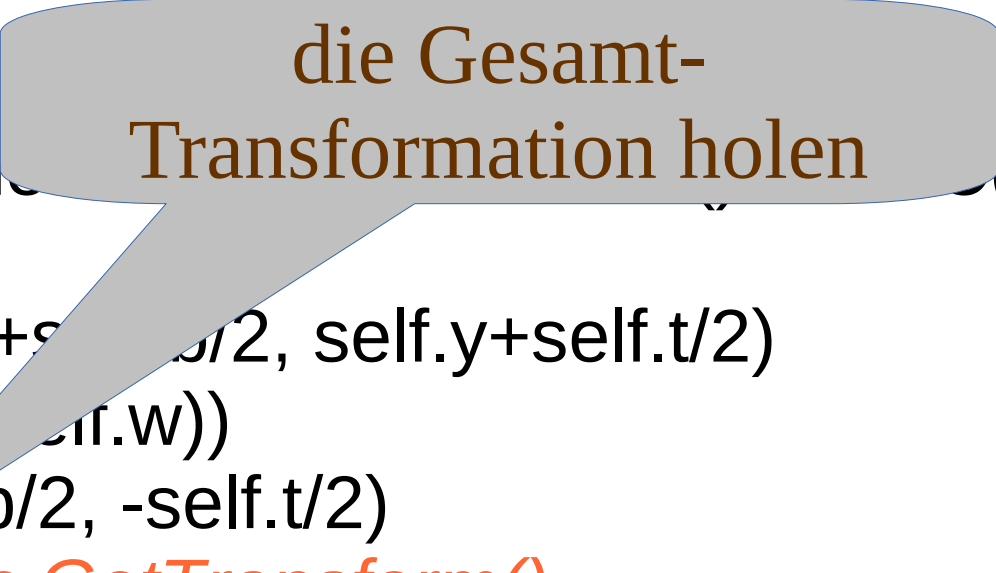
Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    ...  
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()  
    gc.PushState()  
    gc.Translate(self.x+self.b/2, self.y+self.t/2)  
    gc.Rotate(radians(self.w))  
    gc.Translate(-self.b/2, -self.t/2)  
    transformation = gc.GetTransform()  
    gc.PopState()  
    path.Transform(transformation)  
    return path
```



Rückverschiebung

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    ...  
    gc = Zeichenflaeche(...)   
    gc.PushState()  
    gc.Translate(self.x+self.b/2, self.y+self.t/2)  
    gc.Rotate(radians(self.w))  
    gc.Translate(-self.b/2, -self.t/2)  
    transformation = gc.GetTransform()  
    gc.PopState()  
    path.Transform(transformation)  
    return path
```

Sessel mit GraphicsPath

```
def GibFigur(self):  
    path = ...  
    ...  
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()  
    gc.PushState()  
    gc.Translate(self.x+self.b/2, self.y+self.t/2)  
    gc.Rotate(radians(self.w))  
    gc.Translate(-self.b/2, -self.t/2)  
    transformation = gc.GetTransform()  
    gc.PopState()  
    path.Transform(transformation)  
    return path
```

Zustand wieder herstellen

Sessel mit GraphicsPath

```
def GibFigur(self):
    path = ...
    ...
    gc = Zeichenflaeche()
    gc.PushState()
    gc.Translate(self.x+self.w/2, self.y+self.t/2)
    gc.Rotate(radians(self.winkel))
    gc.Translate(-self.w/2, -self.t/2)
    transformation = gc.GetTransform()
    gc.PopState()
    path.Transform(transformation)
    return path
```

die Gesamttransformation
auf den Zeichenfad
anwenden

Sessel mit GraphicsPath

```
def GibFigur(self):
    path = ...
    ...
    gc = Zeichnung.GC()
    gc.PushState()
    gc.Translate(self.x/2, self.y + self.h/2)
    gc.Rotate(radians(self.w))
    gc.Translate(-self.b/2, -self.t/2)
    transformation = gc.GetTransform()
    gc.PopState()
    path.Transform(transformation)
    return path
```

den transformierten
Zeichenfad an den Aufrufer
von GibFigur() zurückgeben