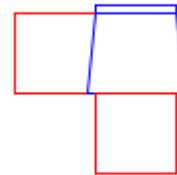


Stuhl, GetBox und Matrix

Im Kurs ist kritisiert worden, dass die Klasse Stuhl die Objekte so zeichnet, dass sie über die quadratischen Abmessungen von 40 x 40 Pixeln hinausragen. Das Bild rechts zeigt den Zustand unter Zuhilfenahme von zwei Tischobjekten mit den Abmessungen von 40 x 40 Pixeln.



Änderung von GibFigur

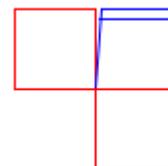
Nach einer Änderung sieht der Programmtext in der Methode GibFigur nun so aus:

```
def GibFigur(self):
    """definiert die zu zeichnende Figur"""
    path = self.GibZeichenPfad()

    # lokale Variable zur Schreibvereinfachung
    b, t = self.GibBreite(), self.GibTiefe()
    path.MoveToPoint(0, t)
    path.AddLineToPoint(b, t)
    path.AddLineToPoint(19*b/20-1.0, 0)
    path.AddLineToPoint(b/20+1.0, 0)
    path.AddLineToPoint(0, t)
    path.MoveToPoint(b/20.0, t/10+1.0)
    path.AddLineToPoint(19*b/20.0, t/10+1.0)

    return self.Transformiere(path)
```

Wie das Bild rechts zeigt, ist das gewünschte Ergebnis erreicht.



GetBox

Zur Abfrage des Umrisses der Figur kann man bei wxPython für ein `wxPath`-Objekt die Methode `GetBox` verwenden. In der Dokumentation steht dazu:

Gets the bounding box enclosing all points (possibly including control points).

Java stellt zu `shapes` eine Methode `getBounds2D()` zur Verfügung.

(Javadoc: *Returns a high precision and more accurate bounding box of the Shape than the `getBounds` method.*)

Mit

```
Rectangle2D dasRechteck = umriss.getBounds2D();
```

kann man ein Rechteckobjekt bekommen, von dem man mit Hilfe der Methoden `getHeight()` und `getWidth()` die Abmessungen abfragen kann. Die Methoden liefern bei einem entsprechend realisiertem Stuhlobjekt die Werte 40.0 für beide.

Erstaunlicherweise sind die Ergebnisse bei Python anders.

Die Abfrage der Werte im ShellFrame (PyShell) mit

```
print stuhl.GibFigur().GetBox()
```

liefert

```
(59.00390625, 19.0, 42.07421875, 42.0)
```

Eine Überprüfung für eines der Tischobjekte mit

```
print tisch2.GibFigur().GetBox()
```

liefert

```
(59.0, 59.0, 42.0, 42.0)
```

und zeigt damit die "andere Philosophie" bei Python. In beiden Fällen sind die Werte der Breite und der Höhe um jeweils 2 zu groß. Python gibt also die Werte für ein Rechteck zurück zu dem alle Punkte im Inneren liegen. Will man die Abmessungen der Figur haben, muss man also sowohl in der Breite als auch in der Höhe jeweils 2 abziehen.

Ovaler Tisch

Für ein Objekt

```
ovalerTisch=OvalerTisch(70, 30, 120, 60, 0, 'green', True)
```

erhält man mit

```
print ovalerTisch.GibFigur().GetBox()
```

entsprechend die Werte

```
(69.0, 29.0, 122.0, 62.0)
```



Auch hier also die jeweils um 2 zu großen Abmessungen.

Bei Berechnungen der Fläche muss man in allen Fällen diese Abweichung berücksichtigen.

Die Transformations-Matrix

Eine weitere Änderung am Projekt habe ich eingebaut, nachdem ich Schwierigkeiten mit dem aus dem Demoprojekt zum `GraphicsPath` entnommenen Programmcode bekommen habe, als ich das Kompositummuster für eine Tischgruppe realisieren wollte. Es geht dabei um die (affine) Transformation, die in beiden Lösungen mit einer Transformations-Matrix gelöst wird.

In der Beispiellösung aus dem Demoprojekt wird dazu allein auf die Methoden vom `GraphicsPath` zugegriffen. Das Generieren der Transformations-Matrix habe ich nun in eine eigene Methode ausgegliedert, so dass in der Methode direkt eine Instanz von `wxGraphicsMatrix` erzeugt wird, die nun keinen `GraphicsPath` mehr benötigt, sondern nur noch den `GraphicsContext gc`.

```
def GibTransformation(self):
    """erzeugt die zugehörige Transformation"""
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
    matrix=gc.CreateMatrix()
    matrix.Translate(self.__x+self.__b/2, self.__y+self.__t/2)
    matrix.Rotate(radians(self.__w))
    matrix.Translate(-self.__b/2, -self.__t/2)
    return matrix
```

Die Methode `Transformiere()` wird dadurch ganz kurz:

```
def Transformiere(self, path):
```

```
"""Transformiert den übergebenen Pfad"""  
path.Transform(self.GibTransformation())  
return path
```

Durch das Entkoppeln kann die eigene Transformationsmatrix ohne einen Bezug zum konkreten path erzeugt und an andere Objekte weiter gegeben werden.

Schnittstelle zu Moebel bleibt erhalten

Da durch diese Änderung die Schnittstelle zu Moebel erhalten bleibt, ist in den anderen Klassen des Projektes keine Änderung notwendig.