

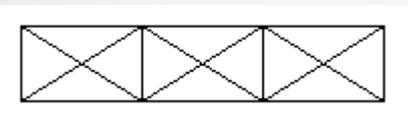
Wiederholung Schrankwand

Stand vor Termin 5 / 2024

Eine Klasse für
eine Schrankwand
erstellen

Wiederholung Schrankwand

- Anlass ist die Aufgabe, eine Klasse zu schreiben, die eine Schrankwand aus drei Schränken darstellt.



- Im ersten Schritt speichern wir schrank.py als schrankwand.py ab und benennen intern passend um.
- GibFigur(self) bleibt zunächst unverändert.

```
def GibFigur(self):  
    """Definiert den Pfad."""  
    gc = Zeichenflaeche.GibZeichenflaeche().GibGC()  
    path = gc.CreatePath()  
  
    # lokale Variable zur Schreibvereinfachung  
    b, t = self.GibBreite(), self.GibTiefe()  
    path.AddRectangle(0, 0, b, t)  
    path.MoveToPoint(0, 0)  
    path.AddLineToPoint(b, t)  
    path.MoveToPoint(b, 0)  
    path.AddLineToPoint(0, t)  
  
    return self.Transformiere(path)
```

Wiederholung Schrankwand

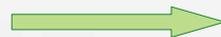
- Wir
 - kopieren den Abschnitt zur Definition eines Schrankes,
 - fügen ihn zweimal ein und
 - editieren die relativen Positionen.

```
# lokale Variable zur Schreibvereinfachung
b, t = self.GibBreite(), self.GibTiefe()
path.AddRectangle(0, 0, b, t)
path.MoveToPoint(0, 0)
path.AddLineToPoint(b, t)
path.MoveToPoint(b, 0)
path.AddLineToPoint(0, t)
# zweiter Schrank
path.AddRectangle(b, 0, b, t)
path.MoveToPoint(b, 0)
path.AddLineToPoint(2*b, t)
path.MoveToPoint(2*b, 0)
path.AddLineToPoint(b, t)
# dritter Schrank
path.AddRectangle(2*b, 0, b, t)
path.MoveToPoint(2*b, 0)
path.AddLineToPoint(3*b, t)
path.MoveToPoint(3*b, 0)
path.AddLineToPoint(2*b, t)
```

Wiederholung Schrankwand

- Die Darstellung gelingt wie gewünscht.
- Die Breite entspricht so aber nicht der Schrankwandbreite, daher wird geändert:

```
# erster Schrank
path.AddRectangle(0, 0, b/3, t)
path.MoveToPoint(0, 0)
path.AddLineToPoint(b/3, t)
path.MoveToPoint(b/3, 0)
path.AddLineToPoint(0, t)
# zweiter Schrank
path.AddRectangle(b/3, 0, b/3, t)
path.MoveToPoint(b/3, 0)
path.AddLineToPoint(2*b/3, t)
path.MoveToPoint(2*b/3, 0)
path.AddLineToPoint(b/3, t)|
# dritter Schrank
path.AddRectangle(2*b/3, 0, b/3, t)
path.MoveToPoint(2*b/3, 0)
path.AddLineToPoint(b, t)
path.MoveToPoint(b, 0)
path.AddLineToPoint(2*b/3, t)
```



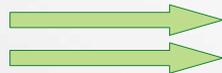
Wiederholung Schrankwand

- Die Darstellung gelingt auch so wie gewünscht.
- Das Vorgehen wird im Kurs hinterfragt und statt dessen 1. vorgeschlagen:

In der Methode `GibFigur()`
soll der Programmblock
für je einen der Schränke
wiederholt (Schleife!) aufgerufen werden

Wiederholung Schrankwand

- Die Darstellung gelingt wie gewünscht.



```
def GibFigur(self):  
    """definiert die zu zeichnende Figur (Pfad)"""  
    path = self.GibZeichenPfad()  
  
    b,t=self.GibBreite()/3,| self.GibTiefe() # Anpassung !  
    for i in range(3): # Zaehlschleife  
        path.AddRectangle(i*b, 0, b, t)  
        path.MoveToPoint(i*b, 0)  
        path.AddLineToPoint((i+1)*b, t)  
        path.MoveToPoint((i+1)*b, 0)  
        path.AddLineToPoint(i*b, t)  
  
    return self.Transformiere(path)
```

Wiederholung Schrankwand

- Die Darstellung gelingt wie gewünscht.
- Das Vorgehen wird im Kurs hinterfragt und statt dessen 2. vorgeschlagen:

Mit der Klasse Schrankwand sollen Schrankwände aus einer beliebigen Anzahl von Schränken erstellt werden können.

Wiederholung Schrankwand

- Wir erarbeiten, dass
 - dazu im Konstruktor ein Parameter `anzahl` notwendig ist und
 - dieser Wert in einem Attribut abgespeichert werden muss,



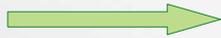
```
class Schrankwand(Moebel):
    def __init__(self,
                 anzahl=3,
                 xPos=0,
                 yPos=0,
                 modulbreite=60,
                 tiefe=37,
                 winkel=0,
                 farbe="black",
                 sichtbar=False):
        Moebel.__init__(self, xPos, yPos,
                       anzahl*modulbreite, tiefe, winkel,
                       farbe, sichtbar)
        self.__anzahl=anzahl
```

Wiederholung Schrankwand

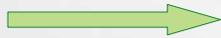
- ...
 - damit in der Methode `GibFigur()` darauf zugegriffen werden kann.



```
b,t=self.GibBreite(), self.GibTiefe()  
n = self.__anzahl
```



```
for i in range(n):  
    # irgendein Schrank  
    path.AddRectangle(i*b/n, 0, b/n, t)  
    path.MoveToPoint(i*b/n, 0)  
    path.AddLineToPoint((i+1)*b/n, t)  
    path.MoveToPoint((i+1)*b/n, 0)  
    path.AddLineToPoint(i*b/n, t)
```



Wiederholung Schrankwand

- Die Darstellung gelingt auch so wie gewünscht.
- Das Vorgehen wird im Kurs weiter hinterfragt und statt dessen 3. vorgeschlagen:

Die Modellierung soll geändert werden:
„Die Schrankwand hat Schrankobjekte“

- Mit dieser Anforderung gehen wir in den nächsten Termin.