

Erster Lösungsvorschlag

```
class Schrank(Moebel):
    def __init__(self,
                 xPos=0,
                 yPos=0,
                 breite=100,
                 tiefe=60,
                 winkel=0,
                 farbe="black",
                 sichtbar=False):
        Moebel.__init__(self, xPos, yPos, ..., sichtbar)

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

```
class Schrankwand(Moebel):
    def __init__(self,
                 xPos=0,
                 yPos=0,
                 breite=180,
                 tiefe=37,
                 winkel=0,
                 farbe="black",
                 sichtbar=False):
        Moebel.__init__(self, xPos, yPos, ..., sichtbar)

    def GibFigur(self):
        path = self.GibZeichenPfad()
        # lokale Hilfsvariable
        b,t=self.GibBreite(), self.GibTiefe()
        # erster Schrank
        path.AddRectangle(0, 0, b/3.0, t)
        path.MoveToPoint(0, 0)
        path.AddLineToPoint(b/3.0, t)
        path.MoveToPoint(b/3.0, 0)
        path.AddLineToPoint(0, t)
        # zweiter Schrank
        path.AddRectangle(b/3.0, 0, b/3.0, t)
        path.MoveToPoint(b/3.0, 0)
        path.AddLineToPoint(2*b/3.0, t)
        path.MoveToPoint(2*b/3.0, 0)
        path.AddLineToPoint(b/3.0, t)
        # dritter Schrank
        path.AddRectangle(2*b/3.0, 0, b/3.0, t)
        path.MoveToPoint(2*b/3.0, 0)
        path.AddLineToPoint(b, t)
        path.MoveToPoint(b, 0)
        path.AddLineToPoint(2*b/3.0, t)
        return self.Transformiere(path)
```