

# Anfangsprojekt

## Attribute kapseln

### Beispiel Stuhlkasse

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/01 Anfangsprojekte/Raumplan... x

File Edit Format Run Options Window Help

```
# stuhl.py
# Zum Anfangsprojekt, mit Kapselung

from grafikfenster import *
from math import radians

### -----
class Stuhl():
    def __init__(self,
                  xPos=20,
                  yPos=20,
                  breite=40,
                  tiefe=40,
                  winkel=0,
                  farbe="black",
                  sichtbar=False):
        self.__x=xPos
        self.__y=yPos
        self.__b=breite
        self.__t=tiefe
        self.__w=winkel
        self.__f=farbe
        self.__s=sichtbar
        if sichtbar: self.Zeige()

    def GibFigur(self):
        """Definiert die Figur <path>"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()
        (x,y,b,t,w,f,s) = self.GibWerte()

        path.MoveToPoint(0, 0)
        path.AddLineToPoint(b, 0)
        path.AddLineToPoint(b*1.1, t)
        path.AddLineToPoint(-b*0.1, t)
        path.AddLineToPoint(0, 0)
        path.AddLineToPoint(0, -t*0.1)
        path.AddLineToPoint(b, -t*0.1)
        path.AddLineToPoint(b, 0)
```

Ln: 1 Col: 0

# Anfangsprojekt

## Attribute kapseln

vordefinierte  
Parameter

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/01 Anfangsprojekte/Raumplan... x

File Edit Format Run Options Window Help

```
# stuhl.py
# Zum Anfangsprojekt, mit Kapselung

from grafikfenster import *
from math import radians

### -----
class Stuhl():
    def __init__(self,
                  xPos=20,
                  yPos=20,
                  breite=40,
                  tiefe=40,
                  winkel=0,
                  farbe="black",
                  sichtbar=False):
        self.__x=xPos
        self.__y=yPos
        self.__b=breite
        self.__t=tiefe
        self.__w=winkel
        self.__f=farbe
        self.__s=sichtbar
        if sichtbar: self.Zeige()

    def GibFigur(self):
        """Definiert die Figur <path>"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()
        (x,y,b,t,w,f,s) = self.GibWerte()

        path.MoveToPoint(0, 0)
        path.AddLineToPoint(b, 0)
        path.AddLineToPoint(b*1.1, t)
        path.AddLineToPoint(-b*0.1, t)
        path.AddLineToPoint(0, 0)
        path.AddLineToPoint(0, -t*0.1)
        path.AddLineToPoint(b, -t*0.1)
        path.AddLineToPoint(b, 0)
```

Ln: 17 Col: 0

# Anfangsprojekt

## Attribute kapseln

Attribute privat

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/01 Anfangsprojekte/Raumplan... x

File Edit Format Run Options Window Help

```
# stuhl.py
# Zum Anfangsprojekt, mit Kapselung

from grafikfenster import *
from math import radians

### -----
class Stuhl():
    def __init__(self,
                  xPos=20,
                  yPos=20,
                  breite=40,
                  tiefe=40,
                  winkel=0,
                  farbe="black",
                  sichtbar=False):
        self.__x=xPos
        self.__y=yPos
        self.__b=breite
        self.__t=tiefe
        self.__w=winkel
        self.__f=farbe
        self.__s=sichtbar
        if sichtbar: self.Zeige()

    def GibFigur(self):
        """Definiert die Figur <path>"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()
        (x,y,b,t,w,f,s) = self.GibWerte()

        path.MoveToPoint(0, 0)
        path.AddLineToPoint(b, 0)
        path.AddLineToPoint(b*1.1, t)
        path.AddLineToPoint(-b*0.1, t)
        path.AddLineToPoint(0, 0)
        path.AddLineToPoint(0, -t*0.1)
        path.AddLineToPoint(b, -t*0.1)
        path.AddLineToPoint(b, 0)
```

Ln: 25 Col: 0

# Anfangsprojekt

## Attribute kapseln

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/01 Anfangsprojekte/Raumplan... x

File Edit Format Run Options Window Help

```
gc.PopState()
path.Transform(transformation)
return path
```

### Alle Get-Methoden

```
def GibX(self):
    """Get-Methode fuer die x-Position"""
    return self.__x
```

```
def GibY(self):
    """Get-Methode fuer die y-Position"""
    return self.__y
```

```
def GibBreite(self):
    """Get-Methode fuer die Breite"""
    return self.__b
```

```
def GibTiefe(self):
    """Get-Methode fuer die Tiefe"""
    return self.__t
```

```
def GibWinkel(self):
    """Get-Methode fuer die Orientierung"""
    return self.__w
```

```
def GibFarbe(self):
    """Get-Methode fuer die Farbe"""
    return self.__f
```

```
def GibSichtbar(self):
    """Get-Methode fuer die Sichtbarkeit"""
    return self.__s
```

```
def GibWerte(self):
    """Get-Methode fuer alle Attributwerte"""
    return (self.__x, self.__y, self.__b, self.__t, self.__w, self.__f, self
```

```
def BewegeHorizontal(self, weite=1):
    """Veraendernde Methode fuer die x-Position"""
    self.Verberge()
    self.__x += weite
```

Ln: 77 Col: 0

# Anfangsprojekt

## Attribute kapseln

alle auf einmal ?

```
gc.PopState()
path.Transform(transformation)
return path

def GibX(self):
    """Get-Methode fuer die x-Position"""
    return self.__x

def GibY(self):
    """Get-Methode fuer die y-Position"""
    return self.__y

def GibBreite(self):
    """Get-Methode fuer die Breite"""
    return self.__b

def GibTiefe(self):
    """Get-Methode fuer die Tiefe"""
    return self.__t

def GibWinkel(self):
    """Get-Methode fuer die Orientierung"""
    return self.__w

def GibFarbe(self):
    """Get-Methode fuer die Farbe"""
    return self.__f

def GibSichtbar(self):
    """Get-Methode fuer die Sichtbarkeit"""
    return self.__s

def GibWerte(self):
    """Get-Methode fuer alle Attributwerte"""
    return (self.__x, self.__y, self.__b, self.__t, self.__w, self.__f, self.__s)

def BewegeHorizontal(self, weite=1):
    """Veraendernde Methode fuer die x-Position"""
    self.Verberge()
    self.__x += weite
```

Ln: 81 Col: 0

# Anfangsprojekt

## Attribute kapseln

### Verändernde Methode?

```
def GibWerte(self):
    """Get-Methode fuer alle Attributwerte"""
    return (self.__x, self.__y, self.__b, self.__t, self.__w, self.__f, self)

def BewegeHorizontal(self, weite=1):
    """Veraendernde Methode fuer die x-Position"""
    self.Verberge()
    self.__x += weite
    self.Zeige()

def BewegeVertikal(self, weite=1):
    """Veraendernde Methode fuer die y-Position"""
    self.Verberge()
    self.__y += weite
    self.Zeige()

def Drehe(self, winkel=10):
    """Veraendernde Methode fuer die Orientierung [Winkel]"""
    self.Verberge()
    self.__w += winkel
    self.Zeige()

def Verberge(self):
    """Veraendernde Methode fuer die Sichtbarkeit mit Wert False"""
    self.__s = False
    Zeichenflaeche.GibZeichenflaeche().Entferne(self)

def Zeige(self):
    """Veraendernde Methode fuer die Sichtbarkeit mit Wert True"""
    self.__s = True
    Zeichenflaeche.GibZeichenflaeche().Zeichne(self)

### -----
class StuhlApp(wx.App):
    """Test-Anwendung speziell fuer Stuhl"""
    def OnInit(self):
        self.__fenster = GrafikFenster(None, "Raumplaner-Grafik")
        self.SetTopWindow(self.__fenster)
        self.__fenster.Show(True)
```

# Anfangsprojekt

## Attribute kapseln

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/01 Anfangsprojekte/Raumplan... x

File Edit Format Run Options Window Help

AendereFarbe

```
def GibWerte(self):
    """Get-Methode fuer alle Attributwerte"""
    return (self.__x, self.__y, self.__b, self.__t, self.__w, self.__f, self

def AendereFarbe(self, neueFarbe="red"):
    """Veraendernde Methode fuer die Farbe"""
    self.Verberge()
    self.__f = neueFarbe
    self.Zeige()

def BewegeHorizontal(self, weite=1):
    """Veraendernde Methode fuer die x-Position"""
    self.Verberge()
    self.__x += weite
    self.Zeige()

def BewegeVertikal(self, weite=1):
    """Veraendernde Methode fuer die y-Position"""
    self.Verberge()
    self.__y += weite
    self.Zeige()

def Drehe(self, winkel=10):
    """Veraendernde Methode fuer die Orientierung [Winkel]"""
    self.Verberge()
    self.__w += winkel
    self.Zeige()

def Verberge(self):
    """Veraendernde Methode fuer die Sichtbarkeit mit Wert False"""
    self.__s = False
    Zeichenflaeche.GibZeichenflaeche().Entferne(self)

def Zeige(self):
    """Veraendernde Methode fuer die Sichtbarkeit mit Wert True"""
    self.__s = True
    Zeichenflaeche.GibZeichenflaeche().Zeichne(self)
```

### -----

Ln: 87 Col: 0