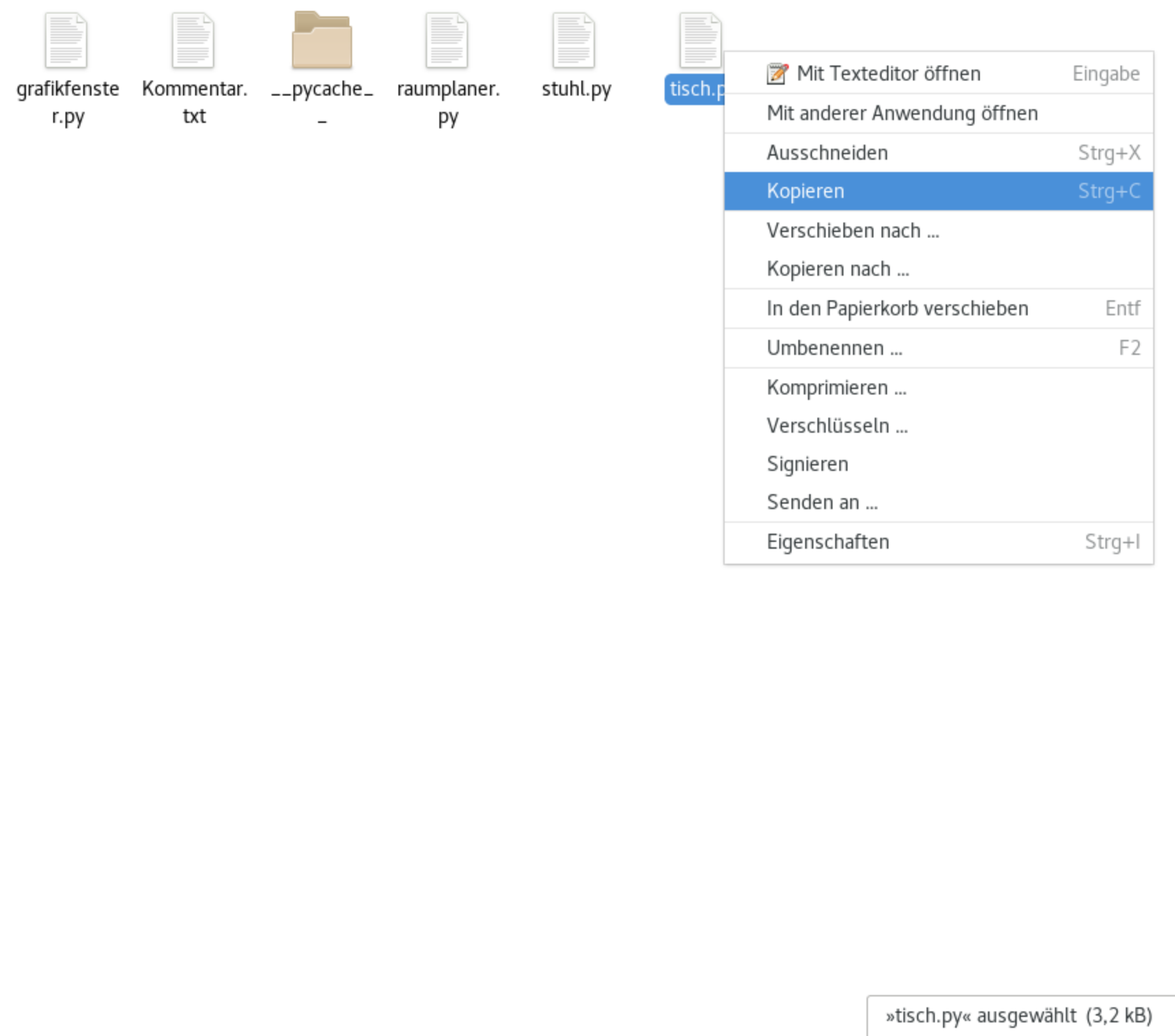
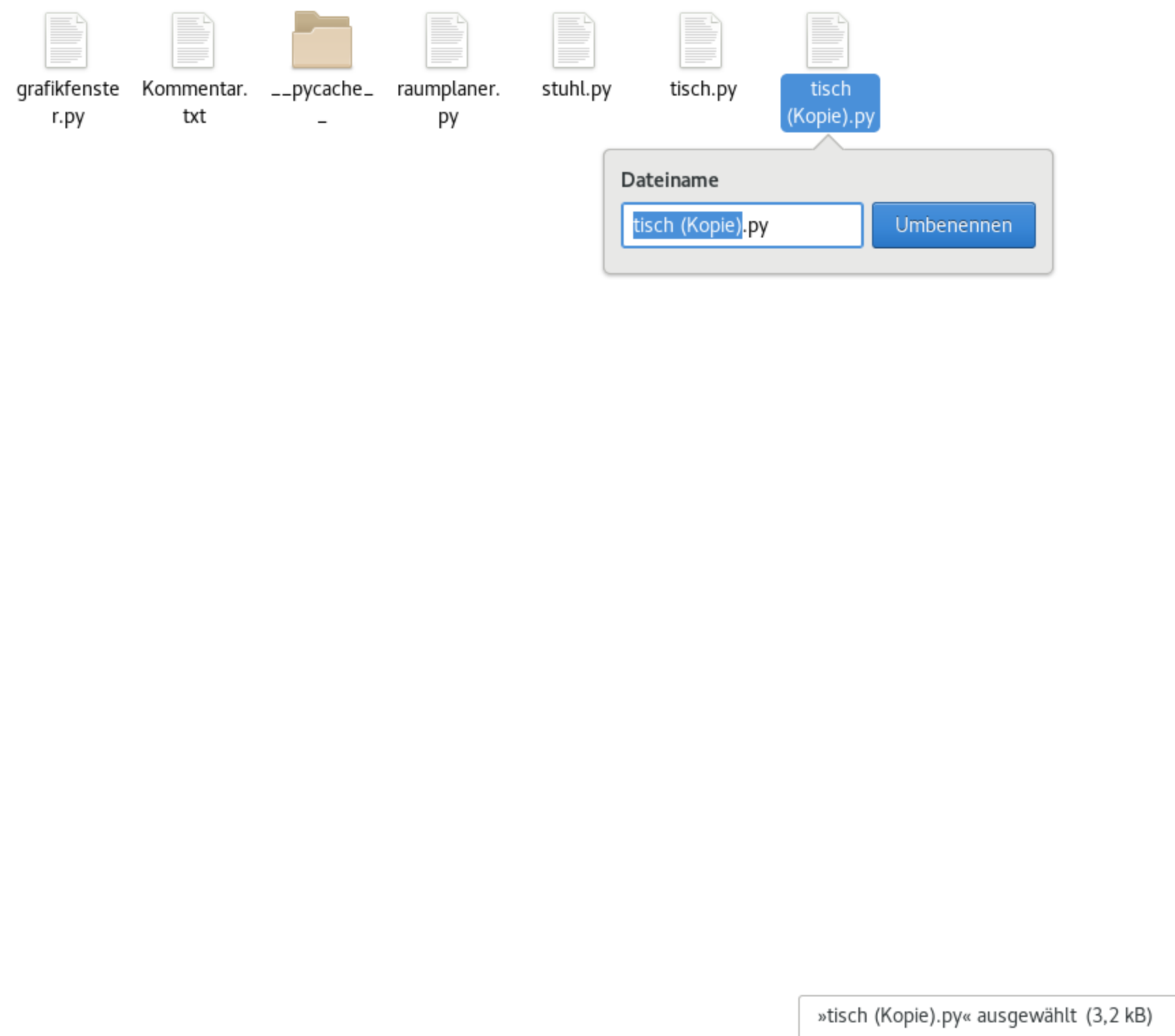

grafikfenster.py
Kommentar.txt
__pycache__
raumplaner.py
stuhl.py
tisch.py


grafikfenster.py
Kommentar.txt
__pycache__
raumplaner.py
stuhl.py
tisch.p
Mit Texteditor öffnen Eingabe
Mit anderer Anwendung öffnen
Ausschneiden Strg+X
Kopieren Strg+C
Verschieben nach ...
Kopieren nach ...
In den Papierkorb verschieben Entf
Umbenennen ... F2
Komprimieren ...
Verschlüsseln ...
Signieren
Senden an ...
Eigenschaften Strg+I
»tisch.py« ausgewählt (3,2 kB)


grafikfenster.py
Kommentar.txt
__pycache__
raumplaner.py
stuhl.py
tisch.py
tisch (Kopie).py


grafikfenster.py
Kommentar.txt
__pycache__
raumplaner.py
stuhl.py
tisch.py
tisch (Kopie).py
Dateiname
tisch (Kopie).py
Umbenennen
»tisch (Kopie).py« ausgewählt (3,2 kB)


grafikfenster.py
Kommentar.txt
moebel.py
__pycache__
raumplaner.py
stuhl.py
tisch.py

grafikfenster.py

moebel.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-...

File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-
# tisch.py

from grafikfenster import *
from math import radians

### ------------------------------------------------
class Tisch():
    """Klasse Tisch
    ermoeglicht das Zeichnen und Bearbeiten eines
    Tisch-Symbols fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        self.__x=70
        self.__y=10
        self.__b=120
        self.__t=60
        self.__w=0
        self.__f='red'
        self.__s=sichtbar
        if sichtbar: self.Zeige()

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        path.AddRectangle(0, 0, self.__b, self.__t)

        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)
```

Ln: 1 Col: 0

»moebel.py« ausgewählt (3,8 kB)

grafikfenste
r.py

moebel.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-...

File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-
# moebel.py

from grafikfenster import *
from math import radians

### ------------------------------------------------
class Moebel():
    """Klasse Moebel
    Oberklasse der Moebel-Symbole fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        self.__x=70
        self.__y=10
        self.__b=120
        self.__t=60
        self.__w=0
        self.__f='red'
        self.__s=sichtbar
        if sichtbar: self.Zeige()

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        path.AddRectangle(0, 0, self.__b, self.__t)

        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()
```

Ln: 11 Col: 0

»moebel.py« ausgewählt (3,8 kB)

grafikfenster.py

*moebel.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-...

File Edit Format Run Options Window Help

```
    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 MoebelApp(wx.App):
    """Test-Anwendung speziell fuer Moebel"""
    def OnInit(self):
        self.__fenster = GrafikFenster(None, "Raumplaner-Grafik")
        self.SetTopWindow(self.__fenster)
        self.__fenster.Show(True)
        self.__fenster.panel.Refresh()
        self.__fenster.ZeigeShellFrame()
        #self.__fenster.ZeigeFillingFrame()
        self.TestAnwendung()
        return True

    def TestAnwendung(self):
        """Allein fuer die Testanwendung:"""
        global moebel
        moebel=Moebel(True)

### ------------------------------------------------
if __name__ == '__main__':
    app = MoebelApp(redirect=False)
    # Parameterwert True, wenn Ausgaben in der Standard E/A angezeigt werden sol
    app.MainLoop()
```

Ln: 107 Col: 0

»moebel.py« ausgewählt (3,8 kB)

grafikfenster.py

tisch.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-
# tisch.py

from grafikfenster import *
from math import radians

from moebel import Moebel

### ------------------------------------------------
class Tisch(Moebel):
    """Klasse Tisch
    ermoeglicht das Zeichnen und Bearbeiten eines
    Tisch-Symbols fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        self.__x=70
        self.__y=10
        self.__b=120
        self.__t=60
        self.__w=0
        self.__f='red'
        self.__s=sichtbar
        if sichtbar: self.Zeige()

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        path.AddRectangle(0, 0, self.__b, self.__t)

        gc.PushState()
        gc.Translate(self.__x+self.__b/2, self.__y+self.__t/2)
```

Ln: 14 Col: 0

»moebel.py« ausgewählt (3,8 kB)

grafikfenste
r.py

tisch.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File Edit Format Run Options Window Help

```
        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 den Winkel"""
##        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 BewegeHorizontal(self, weite):
##        """Veraendernde Methode fuer die x-Position"""
##        self.Verberge()
```

Ln: 55 Col: 25

Ln: 57 Col: 23

»moebel.py« ausgewählt (3,9 kB)

grafikfenste
r.py

tisch.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File

Python 3.6.10 Shell

File Edit Shell Debug Options Window Help

```
nfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 82, in OnPaint
    self.InitBuffer()
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-A
nfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 101, in InitBuffe
r
    self.Draw(self.gc)
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-A
nfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 141, in Draw
    f=self._HoleFarbe(objekt.GibFarbe())
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-A
nfangsprojekt/Moebel-Attribute-kapseln/moebel.py", line 57, in GibFarbe
    return self.__f
AttributeError: 'Tisch' object has no attribute '_Moebel__f'
Traceback (most recent call last):
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-A
nfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 82, in OnPaint
    self.InitBuffer()
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-A
nfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 101, in InitBuffe
r
    self.Draw(self.gc)
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-A
nfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 141, in Draw
    f=self._HoleFarbe(objekt.GibFarbe())
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-A
nfangsprojekt/Moebel-Attribute-kapseln/moebel.py", line 57, in GibFarbe
    return self.__f
AttributeError: 'Tisch' object has no attribute '_Moebel__f'
```

Ln: 5 Col: 0

```
##
##        self.Zeige()
##
##    def Bewege(self, xWeite, yWeite):
```

Ln: 47 Col: 8

»moebel.py« ausgewählt (3,8 kB)

grafikfenste
r.py

tisch.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File Edit Format Run Options Window Help

```
from grafikfenster import *
from math import radians

from moebel import Moebel

### ------------------------------------------------
class Tisch(Moebel):
    """Klasse Tisch
    ermoeglicht das Zeichnen und Bearbeiten eines
    Tisch-Symbols fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        Moebel.__init__(self,70,10,120,60,0,'red',sichtbar)

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        path.AddRectangle(0, 0, self.__b, self.__t)

        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

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

Ln: 17 Col: 29

»moebel.py« ausgewählt (3,8 kB)

*moebel.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-...

File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-
# moebel.py

from grafikfenster import *
from math import radians

### ------------------------------------------------
class Moebel():
    """Klasse Moebel
    Oberklasse der Moebel-Symbole fuer den Raumplaner"""

    def __init__(self,x,y,b,t,w,f,sichtbar):
        """Konstruktor"""
        self.__x=x
        self.__y=y
        self.__b=b
        self.__t=t
        self.__w=w
        self.__f=f
        self.__s=sichtbar
        if sichtbar: self.Zeige()

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        path.AddRectangle(0, 0, self.__b, self.__t)

        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()
```

Ln: 22 Col: 0

»moebel.py« ausgewählt (3,8 kB)

grafikfen
r.py

tisch.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

Python 3.6.10 Shell

File Edit Shell Debug Options Window Help

```
Traceback (most recent call last):
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 82, in OnPaint
    self.InitBuffer()
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 101, in InitBuffer
    self.Draw(self.gc)
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 134, in Draw
    if objekt.GibFigur()==NotImplemented: break
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/tisch.py", line 24, in GibFigur
    path.AddRectangle(0, 0, self.__b, self.__t)
AttributeError: 'Tisch' object has no attribute '_Tisch__b'
Traceback (most recent call last):
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 82, in OnPaint
    self.InitBuffer()
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 101, in InitBuffer
    self.Draw(self.gc)
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/grafikfenster.py", line 134, in Draw
    if objekt.GibFigur()==NotImplemented: break
  File "/home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anfangsprojekt/Moebel-Attribute-kapseln/tisch.py", line 24, in GibFigur
    path.AddRectangle(0, 0, self.__b, self.__t)
AttributeError: 'Tisch' object has no attribute '_Tisch__b'
>>>
```

Ln: 45 Col: 4

return self._x

Ln: 11 Col: 0

»moebel.py« ausgewählt (3,8 kB)

grafikfenster.py

*tisch.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-An... ×

File Edit Format Run Options Window Help

```
from moebel import Moebel

### ------------------------------------------------
class Tisch(Moebel):
    """Klasse Tisch
    ermoeglicht das Zeichnen und Bearbeiten eines
    Tisch-Symbols fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        Moebel.__init__(self,70,10,120,60,0,'red',sichtbar)

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        x,y = self.GibX(),self.GibY()
        b,t,w = self.GibBreite(),self.GibTiefe(),self.GibWinkel()

        path.AddRectangle(0, 0, b, t)
        gc.PushState()
        gc.Translate(x+b/2, y+t/2)
        gc.Rotate(radians(w))
        gc.Translate(-b/2, -t/2)
        transformation = gc.GetTransform()
        gc.PopState()
        path.Transform(transformation)
        return path

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

Ln: 32 Col: 0

»moebel.py« ausgewählt (3,9 kB)

grafikfenster.py

tisch.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-
# tisch.py

from grafikfenster import *
from math import radians

from moebel import Moebel

### ------------------------------------------------
class Tisch(Moebel):
    """Klasse Tisch
    ermoeglicht das Zeichnen
    Tisch-Symbols fuer den Ra

    def __init__(self, sichtb
        """einfacher Konstruk
        Moebel.__init__(self,

    def GibFigur(self):
        """definiert und tran
        gc = Zeichenflaeche.G
        path = gc.CreatePath(

        x,y = self.GibX(),sel
        b,t,w = self.GibBreit

        path.AddRectangle(0,
        gc.PushState()
        gc.Translate(x+b/2, y
        gc.Rotate(radians(w))
        gc.Translate(-b/2, -t
        transformation = gc.Ge
        gc.PopState()
        path.Transform(transformation)
```

Raumplaner-Grafik

Ln: 1 Col: 0

Ln: 57 Col: 23

»moebel.py« ausgewählt (3,9 kB)

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-
# stuhl.py

from grafikfenster import *
from math import radians

from moebel import Moebel

### ------------------------------------------------
class Stuhl(Moebel):
    """Klasse Stuhl
    ermoeglicht das Zeichnen und Bearbeiten eines
    Stuhl-Symbols fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        Moebel.__init__(self,20,20,40,40,270,"blue",sichtbar)

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        path.MoveToPoint(0, 0)
        path.AddLineToPoint(self.__b, 0)
        path.AddLineToPoint(self.__b*1.1, self.__t)
        path.AddLineToPoint(-self.__b*0.1, self.__t)
        path.AddLineToPoint(0, 0)
        path.AddLineToPoint(0, -self.__t*0.1)
        path.AddLineToPoint(self.__b, -self.__t*0.1)
        path.AddLineToPoint(self.__b, 0)

        gc.PushState()
        gc.Translate(self.__x+self.__b/2, self.__y+self.__t/2)
```

Ln: 17 Col: 24

*stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-An... ×

File Edit Format Run Options Window Help

```
    """Klasse Stuhl
    ermoeglicht das Zeichnen und Bearbeiten eines
    Stuhl-Symbols fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        Moebel.__init__(self,20,20,40,40,270,"blue",sichtbar)

    def GibFigur(self):
        """definiert und transformiert die zu zeichnende Figur"""
        gc = Zeichenflaeche.GibZeichenflaeche().GibGC()
        path = gc.CreatePath()

        x,y = self.GibX(),self.GibY()
        b,t,w = self.GibBreite(),self.GibTiefe(),self.GibWinkel()

        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)

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

Ln: 40 Col: 0

grafikfer
r.py

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File Edit Format Run Options Window Help

```
from moebel import Moebel

### ------------------------------------------------
class Stuhl(Moebel):
    """Klasse Stuhl
    ermoeglicht das Zeichnen und Bearbeiten eines
    Stuhl-Symbols fuer den Raumplaner"""

    def __init__(self, sichtbar=False):
        """einfacher Konstruktor"""
        Moebel.__init__(self,20,20,40,40,270,

    def GibFigur(self):
        """definiert und transformiert die zu
        gc = Zeichenflaeche.GibZeichenflaeche
        path = gc.CreatePath()

        x,y = self.GibX(),self.GibY()
        b,t,w = self.GibBreite(),self.GibTief

        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)

        gc.PushState()
        gc.Translate(x+b/2, y+t/2)
        gc.Rotate(radians(w))
        gc.Translate(-b/2, -t/2)
```

Ln: 106 Col: 0

Raumplaner-Grafik

Ln: 57 Col: 23

»moebel.py« ausgewählt (3,9 kB)

grafikfenster.py

stuhl.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-aus-Anf...

File Edit Format Run Options Window Help

raumplaner.py - /home/nutzer/Dokumente/LI/2021-LI-OO/Projekte/02/Moebel-erarbeiten-a...

File Edit Format Run Options Window Help

```
# raumplaner.py
# Anfangsprojekt mit Kapselung der Attribute
# und allen sondierenden und veraendernden Methoden

from grafikfenster import *

from stuhl import *
from tisch import *

### ------------------------------
class MyApp(wx.App):
    """Testanwendung fuer Raumplaner
       ShellFrame: Interaktion mit T
       FillingFrame: Anzeige der Umg
    def OnInit(self):
        self.__fenster = GrafikFenst
        self.SetTopWindow(self.__fen
        self.__fenster.Show(True)
        self.__fenster.panel.Refresh
        self.__fenster.ZeigeShellFra
        #self.__fenster.ZeigeFilling
        self.TestAnwendung()
        return True

    def TestAnwendung(self):
        """Allein fuer die Testanwendung:"""
        global tisch, stuhl
        tisch=Tisch(True)
        stuhl=Stuhl(True)
```

Ln: 1 Col: 0

Ln: 13 Col: 40

Raumplaner-Grafik

Ln: 57 Col: 23

»moebel.py« ausgewählt (3,9 kB)