

Konfigurationsdialog

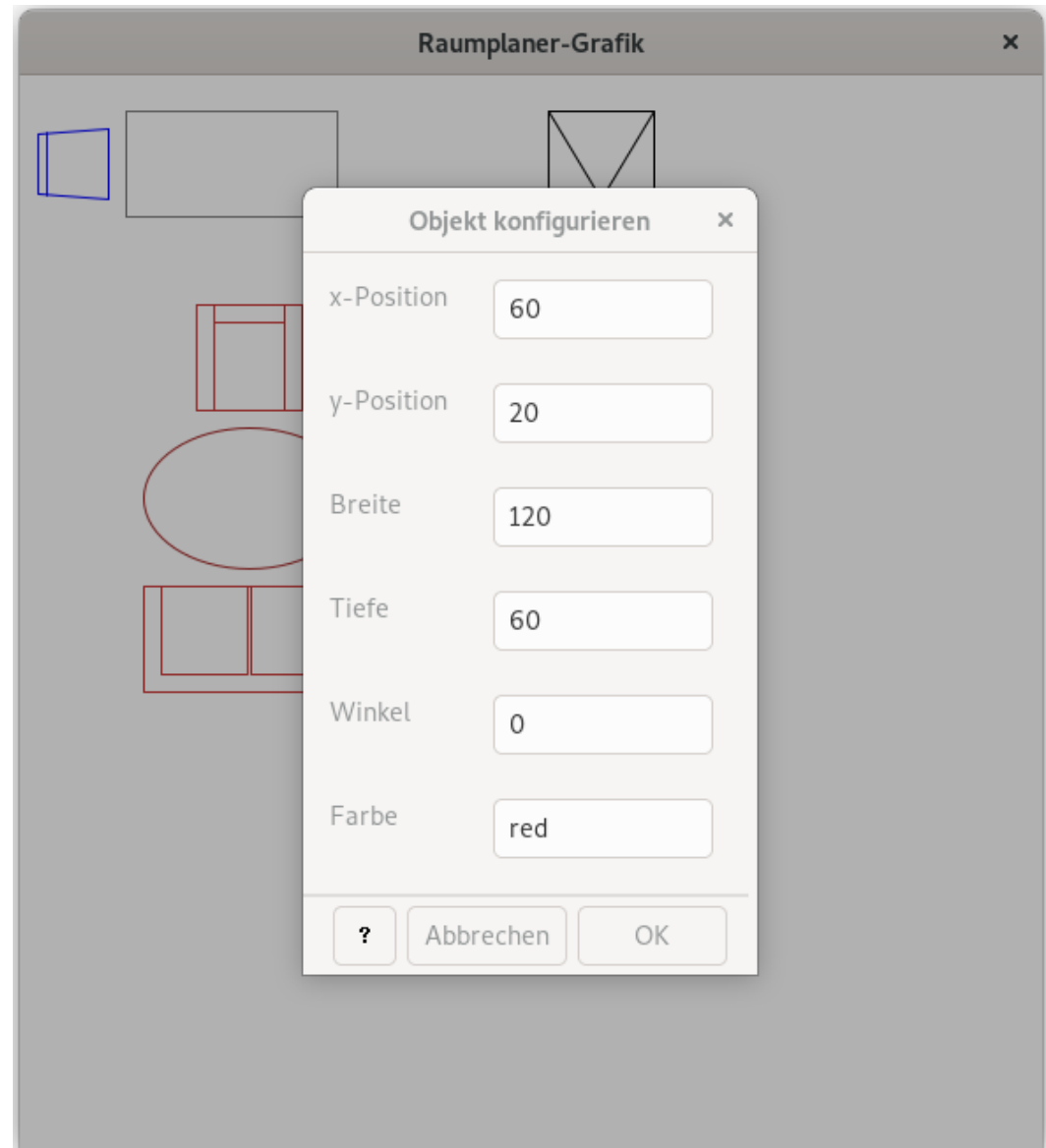
Einsatz eines
Konfigurationsdialogs
für die Moebel-Objekte

Konfigurationsdialog

- Rechter Mausklick auf eines der Moebel-Objekte auf der Zeichenflaeche startet einen Konfigurationsdialog
- Er bekommt die Daten des Objekts übergeben und zeigt sie in editierbaren TextCtrl-Feldern an
- Die Verarbeitung ihrer Inhalte erfolgt aber nicht sofort, sondern erst nach Schließen des Dialogfensters

Konfigurationsdialog

- Dialogfenster



Konfigurationsdialog

- Rechter Mausklick ...

```
def RechtsKlick(self, x, y):
    self.Waehle(-1)
    self.__clickPosition=None
    wahl=None
    for moebel in self.__alleMoebel:
        if self.Angeklickt(moebel, x, y):
            self.Waehle(self.__alleMoebel.index(moebel))
            self.__clickPosition=(x, y)
            wahl=self.GibAusgewaehltes()
    if wahl!=None:
        ...
```

Konfigurationsdialog

- Rechter Mausklick ...

```
...
if wahl!=None:
    dialog=KonfigurationsDialog(self.__zf,
                                -1,
                                "Objekt konfigurieren",
                                size=(350, 200))

    dialog.DatenSetzen(
        str(wahl.GibX()),
        str(wahl.GibY()),
        str(wahl.GibBreite()),
        str(wahl.GibTiefe()),
        str(wahl.GibOrientierung()),
        wahl.GibFarbe(True),
        wahl.GibFuellFarbe(True))
    dialog.CenterOnScreen()
    result=dialog.ShowModal()
...
```

Konfigurationsdialog

- Rechter Mausklick ...

```
...
if (dialog.OKAngeklickt(result)):
    try:
        wahl.BewegeHorizontal(
            int(dialog.textX.GetValue())
            -wahl.GibX())
    except ValueError as e:
        self.Meldung('Horizontal: Zahl eingeben!')
    try:
        wahl.BewegeVertikal(
            int(dialog.textY.GetValue())
            -wahl.GibY())
    except ValueError as e:
        self.Meldung('Vertikal: Zahl eingeben!')
...

dialog.Destroy()
```

Konfigurationsdialog

- Dialogfenster
 - Kopf

```
class KonfigurationsDialog(wx.Dialog):
    def __init__(
        self, parent,
        ID, title, size=wx.DefaultSize,
        pos=wx.DefaultPosition,
        style=wx.DEFAULT_DIALOG_STYLE):

        wx.Dialog.__init__(self,
                           parent, ID, title, pos, size, style)

        ...
```

Konfigurationsdialog

- Dialogfenster

- *Sizer* einsetzen, also kein durch Angabe der Positionen vorgegebenes statisches Layout

...

```
sizer = wx.BoxSizer(wx.VERTICAL)
```

```
gbs = wx.GridBagSizer(vgap=5, hgap=5)
```

```
labelX = wx.StaticText(self, -1, "x-Position")
```

```
self.textX = wx.TextCtrl(self, -1, "", size=(125, -1))
```

```
labelY = wx.StaticText(self, -1, "y-Position")
```

```
self.textY = wx.TextCtrl(self, -1, "", size=(125, -1))
```

....

```
gbs.Add(labelX, (0,0), flag=wx.ALL, border=10)
```

```
gbs.Add(self.textX, (0,1), flag=wx.ALL, border=10)
```

...

Konfigurationsdialog

- Dialogfenster
 - Buttonzeile

...

```
btnsizer = wx.StdDialogButtonSizer()
```

```
if wx.Platform != "__WXMSW__":
```

```
    btn = wx.ContextHelpButton(self)
```

```
    btnsizer.AddButton(btn)
```

```
okbtn = wx.Button(self, wx.ID_OK)
```

```
okbtn.SetHelpText("Beenden und uebernehmen")
```

```
okbtn.SetDefault()
```

```
btnsizer.AddButton(okbtn)
```

```
self.Bind(wx.EVT_BUTTON, self.OnOKButton, okbtn)
```

```
btn = wx.Button(self, wx.ID_CANCEL)
```

```
btn.SetHelpText("Abbruch ohne die Werte zu uebernehmen")
```

```
btnsizer.AddButton(btn)
```

```
btnsizer.Realize()
```

```
sizer.Add(btnsizer, 0, wx.ALL, 5)
```

...

Konfigurationsdialog

- Dialogfenster
 - Daten setzen

```
def DatenSetzen(self, x, y, breite, tiefe,  
                winkel, farbe, fuellfarbe):  
    self.textX.ChangeValue(x)  
    self.textY.ChangeValue(y)  
    self.textBreite.ChangeValue(breite)  
    self.textTiefe.ChangeValue(tiefe)  
    self.textWinkel.ChangeValue(winkel)  
    self.textFarbe.ChangeValue(farbe)  
    self.textFuellfarbe.ChangeValue(fuellfarbe)
```