

# Probleme bei MoebelGruppe

*Probleme  
bei der Klasse  
MoebelGruppe*

# Probleme bei MoebelGruppe

- Beachten Sie die Reihenfolge im Konstruktor der Klasse MoebelGruppe:

```
def __init__(self,  
             xPos=50,  
             yPos=150,  
             winkel=0,  
             farbe=""):  
    self.__komponenten=[]  
    Moebel.__init__(self, xPos, yPos, 0, 0, winkel, farbe, True)
```

- Anderenfalls gibt es eine Fehlermeldung beim Erzeugen einer Instanz von MoebelGruppe

# Probleme bei MoebelGruppe

- Der Grund liegt in der Reihenfolge der Initialisierung:
- Der Konstruktor der Oberklasse Moebel wird anderenfalls aufgerufen, bevor das Attribut angelegt wird.
- Er wird mit **sichtbar=True** aufgerufen.
- In der letzten Zeile des Konstruktors von Moebel wird bei Zustand *sichtbar* die Methode **Zeige()** aufgerufen.
- Das ist aber die **Zeige()**-Methode von Tischgruppe, in der auf **self.\_\_komponenten** zugegriffen wird.

# Probleme bei MoebelGruppe

- Das ist aber die **Zeige()**-Methode von Tischgruppe, in der auf **self.\_\_komponenten** zugegriffen wird:

```
def Zeige(self):  
    """Veraendernde Methode fuer die Sichtbarkeit mit Wert True"""  
    self.__s = True  
    for moebel in self.__komponenten:  
        moebel.Zeige()
```

# Probleme bei MoebelGruppe

- Fehler: Kein Attribut `self.gc` in Zeichenflaeche beim Starten der Testanwendung aus der OnInit() - Methode der RaumplanerApp
- Die Ursache ist der Aufruf des Zeichnens für die leere Gruppe.
- Der Fehler ist von mir korrigiert durch den Einbau der Initialisierung des Attributs im Konstruktor

```
self.gc = None
```

- und Prüfen in der Get-Methode

```
def GibGC(self):
```

```
    """Gibt den aktuellen Graphics-Context zurueck."""
```

```
    if self.gc==None: self.InitBuffer()
```

```
    return self.gc
```

# Probleme bei MoebelGruppe

- Der Fehler ist also kein spezieller Fehler bei den Möbelgruppen, sondern bei ihnen nur deutlich geworden.
- Der Fehler ist eine unsaubere Initialisierung eines Attributs und damit ein gutes Beispiel für die Nachteile, die entstehen können, weil Python es zulässt Attribute auch an anderer Stelle als im Konstruktor zu initialisieren.
- Das Fehlen der Initialisierung selbst ist bei Java zwar prinzipiell auch möglich, allerdings bemerkt der Compiler eine fehlende Deklaration im Kopf der Klasse.

# Probleme bei MoebelGruppe

- Beachten Sie die unterschiedliche Reihenfolge der Transformationen mit Gruppentransformation im Concat (Verknüpfung) bei älteren und neueren Versionen von wxPython:
- V1 (neu)

```
matrix=gc.CreateMatrix()  
if self.__gruppe!=None:  
    matrix.Concat(self.__gruppe.GibGesamtTransformation())  
  
matrix.Concat(self.GibTransformation())
```

- V2 (älter)

```
matrix=gc.CreateMatrix()  
matrix.Concat(self.GibTransformation())  
if self.__gruppe!=None:  
  
    matrix.Concat(self.__gruppe.GibGesamtTransformation())e
```