Daten lesbar speichern

Daten lesbar speichern

Daten lesbar speichern

- Der Einsatz des Moduls Pickle zum Speichern der Daten ist nicht die einzige sinnvolle Möglichkeit.
- Sinnvoll ist auch, die Daten in einer lesbaren Form abzuspeichern, also als Text.
 - Sie sind dann in einem einfachen Texteditor lesbar und ggf auch änderbar.
 - *Hinzu kommt: Das Speichern bei tkinter hat nicht mit Pickle funktioniert - warum auch immer.*

Daten lesbar speichern

- Dazu muss die Klasse Moebel die Daten (als Tupel) bereitstellen.

```
def GibDaten(self):  
    daten=(str(self.__class__).partition('.')[2].partition('')[0],  
           self.__x,  
           self.__y,  
           self.__b,  
           self.__t,  
           self.__w,  
           self.__f,  
           self.__ff,  
           self.__s)  
    return daten
```

Klassenname
als
String

Füllfarbe
eingebaut !

Daten lesbar speichern

- RaumplanerModell sammelt sie in einer Liste

...

```
def ErzeugeDaten(self):  
    modellDaten=[]  
    for moebel in self.__alleMoebel:  
        modellDaten.append(moebel.GibDaten())  
    return modellDaten
```

Daten lesbar speichern

- ... die dann von der Methode zum Speichern in einen String verwandelt werden.

```
def Speichere(self, dateiname, modellDaten=[]):  
    self.Waehle(-1)  
    try:  
        f=open(dateiname,'w')    ## nicht mehr binär  
    except IOError as e:  
        self.__gui.Ausgabe('Datei existiert nicht,...')  
        return None  
    f.write( str(self.ErzeugeDaten()) )  
    f.close()  
    return 'OK'
```

Daten lesbar speichern

- Beim Laden ist einiges zu tun:

```
def Lade(self, dateiname):  
    try:  
        f=open(dateiname,'r')  
    except IOError as e:  
        self.__gui.Ausgabe('Datei existiert nicht...')  
        return None  
    modellDatenString=f.read()  
    f.close()  
    self.LoescheAktuellesModell()  
    modellDaten=eval(modellDatenString)  
    self.ErzeugeModellAusDaten(modellDaten)  
    return 'OK'
```

Daten lesbar speichern

- Nicht zwei Modelle gleichzeitig!
- Achtung: Beim Löschen von Daten aus Listen muss erst eine echte Kopie erstellt werden, über die dann iteriert wird, nicht über die Liste selbst!

```
def LoescheAktuellesModell(self):  
    self.Waehle(-1)  
    kopie=[]+self.__alleMoebel  
    for moebel in kopie:  
        moebel.Verberge()  
        self.__alleMoebel.remove(moebel)  
        del(moebel)
```

Daten lesbar speichern

- RaumplanerModell hat die Daten aus einer Liste gespeichert, jetzt liegen sie als ein String vor und müssen erst wieder zu einer Liste werden ...

modellDaten=eval(modellDatenString)

Daten lesbar speichern

- ... damit sie nacheinander der Methode Neu übergeben werden können.

```
def ErzeugeModellAusDaten(self, modellDaten):  
    for daten in modellDaten:  
        self.Neu(daten[0], # Name der Klasse  
                daten[1], # x  
                daten[2], # y  
                daten[3], # b  
                daten[4], # t  
                daten[5], # w  
                daten[6], # f  
                daten[7], # ff  
                daten[8]) # sichtbar
```

Daten lesbar speichern

- Da Moebel in der verwendeten Version ggf keine Füllfarbe erwartet, muss diese in allen konkreten Möbelklassen und bei der Klasse Moebel selbst als Parameter im Konstruktor, als Attribut und mit einer Get-Methode auftreten.

```
self.__ff=fuellfarbe
```

```
def GibFuellFarbe(self, original=False):  
    if original: return self.__ff  
    if self.__ausgewaehlt: return "LIGHT GREY"  
    return self.__ff
```