Schrankwand-Kompositum

Die Klasse Schrankwand
als Kompositum
modellieren

Schrankwand-Kompositum

- Anlass ist der Vorschlag, die Klasse Schrankwand so zu modellieren, dass die Anzahl der Schränke zur Laufzeit verändert werden kann.
- Es gibt verschiedene Ansätze, das zu modellieren, hier wird nur einer davon betrachtet.

Schrankwand-Kompositum

Notwendige Methoden

- FuegeHinzu(self, schrank):
  Fügt zu einem Schrankwandobjekt ein (weiteres) Schrankobjekt hinzu
- Entferne(self, schrank):
  Entfernt eines der Schrankobjekte aus dem Schrankwandobjekt

Schrankwand-Kompositum

- FuegeHinzu(self, schrank):
  Fügt zu einem Schrankwandobjekt ein (weiteres) Schrankobjekt hinzu
- Die Methode arbeitet nach dem üblichen Prinzip
  - Schrankobjekt an Liste anhängen
  - Update der Darstellung
- Achtung: Sicherstellen, dass das Schrankobjekt selbst nicht sichtbar ist:
  `schrank.Verberge()`

Schrankwand-Kompositum

- Entferne(self, schrank):
  Entfernt eines der Schrankobjekte aus dem Schrankwandobjekt
- Die Methode arbeitet nach dem üblichen Prinzip
  - Schrankobjekt aus der Liste entfernen
  - Update der Darstellung
- Allerdings sollten fehlerhafte Anwendungen verhindert werden, bei der ein nicht enthaltenes Objekt entfernt werden soll.

Schrankwand-Kompositum

Problem

- Weshalb funktioniert Drehe() nicht wie gewünscht?
- Die Abmessungen der Schrankwand sind noch nicht angepasst worden, daher wird als Drehzentrum die linke obere Ecke verwendet.
- Daher muss diese Anpassung in das Projekt mit übernommen werden.

Schrankwand-Kompositum

- Da die Methode nicht von außen aufgerufen werden soll, sollte sie als nicht öffentlich gekennzeichnet werden.
  Also `_AktualisiereAbmessungen()`
- Sie kann in die Klasse Moebel eingefügt werden, wenn sie direkt auf die Attribute `self.__b` und `self.__t` zugreifen soll. Anderenfalls nutzt man die verändernden Methoden `AendereBreite(...)` und `AendereTiefe(...)`.

Schrankwand-Kompositum

- Leider gibt es eine kleine Veränderung in den beiden Python-Versionen. Für Python 2 gehen die Abmessungen des Rechtecks über die Figur hinaus:
- Die Methode holt sich die Werte über
  `daten = self.GibFigur().GetBox().Get()`
  und setzt dann die Werte korrigiert mit
  `self.__b = int(daten[2]-2)`
  `self.__t = int(daten[3]-2)`

Schrankwand-Kompositum

- In Python 3 ist das verbessert:
- Die Methode holt sich die Werte ebenso über

```
daten = self.GibFigur().GetBox().Get()
```

und setzt dann die Werte mit

```
self.__b = round(daten[2])
self.__t = round(daten[3])
```

- alternativ also mit

```
self.AendereBreite(round(daten[2]))
self.AendereTiefe(round(daten[3]))
```