

Überlegungen zum Kompositummuster bei Schrankwand

Eine kurze Vorbemerkung: Leider bietet sich bei diesem Anwendungsfall nicht die Möglichkeit einer tiefen Baumstruktur.

FuegeHinzu und Entferne auf einem Stack

Bei der Methode FuegeHinzu ebenso wie bei der nächsten sollte man beachten, dass es nicht sinnvoll ist, eine auseinander gerissene Schrankwand zu erstellen. Um das zu vermeiden, sollte man prinzipiell zur Verwaltung einen Stack (→ ADT) zum Speichern verwenden, damit die neu hinzugefügten Schrankobjekte immer am Ende hinzugefügt werden und beim Entfernen -anders als bei einer Warteschlange- ebenfalls nur am Ende entfernt wird.

Wenn eine normale Liste zum Speichern verwendet wird, muss das Programm gewährleisten, dass die Zugriffe in der beschriebenen Weise erfolgen.

FuegeHinzu

Damit der nächste Schrank ohne Abstand angehängt werden kann, muss das Schrankwandobjekt beim zuletzt hinzugefügten Schrank den Anschlusspunkt abfragen können. Die x-Koordinate des neuen Schranks ergibt sich aus der Summe entweder der x-Koordinate des anderen Schranks oder der Schrankwand mit der Schrankbreite. Für diese Werte muss es daher Zugriffsmethoden geben oder die Schrankwand muss sie verwalten.

Sinnvollerweise verwaltet man neben der Schrankbreite auch die Gesamtbreite der Schrankwand, da sie sowieso für die Transformation benötigt wird. Nach jeder Aktion, welche die Anzahl der Schränke verändert sollte daher ein update der Werte erfolgen. Der Ablauf ist also:

1. Erzeuge einen neuen Schrank an einer sinnvollen Ausgangsposition,
2. verschiebe ihn ggf. an die richtige Position in der Schrankwand
3. aktualisiere die Breite der Schrankwand.

Entferne

Prinzipiell umgekehrt ist das Vorgehen beim Entfernen.

1. Entferne das letzte Schrankwandobjekt und
2. aktualisiere die Breite der Schrankwand.