

Viele Schränke bilden eine Schrankwand

Das einfache Projekt Schrankwand könnte so erweitert werden, dass die Schrankwand eine „beliebige“ Zahl von Schränken enthalten kann. In diesem Fall brauchen wir statt der Objekte schrank1, schrank2 und schrank3 eine Variable [gesucht ist also ein Datentyp] mit der wir alle Schränke gemeinsam verwalten können, also z.B. eine Liste von Schränken oder ein Array von Schränken. Wir werden uns mehrere Fälle ansehen, zunächst das Beispiel Array.

Lösung mit dem Array

Unsere Klasse Schrankwand benötigt nun eine Objektvariable zum Speichern der Schränke (wir nennen sie schraenke) und eine integer – Variable anzahl, um die Zahl der Schränke zu speichern. Wir nehmen die Deklaration in den Kopf der Klasse auf:

```
private Schrank[] schraenke;
private int anzahl;
```

Die nachgestellte eckige Klammer deklariert schraenke als ein Array von Exemplaren der Klasse Schrank. Der Konstruktor sollte nun so verändert werden, dass Schrankwände mit verschiedenen Anzahlen erzeugt werden können. Dazu benötigt er einen Parameter für die Anzahl.

Außerdem muss das Array selbst initialisiert werden, anschließend sind noch die Schrankobjekte in gewünschter Anzahl und Position zu erzeugen. Die Änderungen im Konstruktor sind also:

```
public Schrankwand(int anzahl)
{
    xPosition = 40;
    yPosition = 80;
    farbe = "blau";
    orientierung = 0;
    istSichtbar = false;
    int schrankbreite = 60;
    this.anzahl = anzahl;
    breite = schrankbreite*anzahl;
    tiefe = 37;
    schraenke = new Schrank[anzahl];
    for (int i=0;i<anzahl;i++){
        schraenke[i] =
            new Schrank(i*schrankbreite, 0,
                farbe, orientierung,
                schrankbreite, tiefe);
    }
}
```

Parameter in Variable übernehmen

Breite der Schrankwand berechnen

Schrankobjekte erzeugen

Schrankposition berechnen

Beachten Sie bitte, dass im Schleifenkopf die richtigen Werte für die Variable i verwendet werden. i wird zunächst auf der Startwert 0 gesetzt, da die erste Position im Array den Index 0 hat, die Zählvorschrift i++ zählt jeweils um 1 weiter und die Laufbedingung, mit der man den Abbruch der Schleife festlegt, muss i < anzahl lauten, da der höchste zulässige Index den Wert anzahl-1 hat. Die zulässigen Indizes gehen also von 0..anzahl-1.

Die Änderungen in gibAktuelleFigur() sind auch sehr einfacher Art. Hier muss das Einfügen in den GeneralPath über eine Schleife gesteuert werden.

```
protected Shape gibAktuelleFigur()
{
    GeneralPath schrankwand = new GeneralPath();
    for (int i=0;i<anzahl;i++){
        schrankwand.append(schraenke[i].gibAktuelleFigur(), false);
    }
    return transformiere(schrankwand);
}
```

Zugriff auf das Array über den Index