

Aufgaben für Gruppenarbeit

Zwei Aufgaben

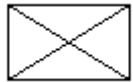
Zwei unabhängige Aufgaben werden gestellt, die in getrennten Gruppen bearbeitet werden sollen.

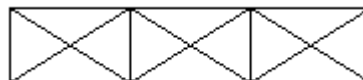
Die Ergebnisse sollen von den Gruppen am Ende ihrer Arbeit den anderen Gruppen vorgestellt werden.

1. Aufgabe

1. Zur Wiederholung überlege, weshalb wir Vererbung in unserem Projekt nutzen und schreibe die Gründe auf.
2. Untersuche das vorliegende Projekt "**02-Raumplaner-mit-Vererbung**" daraufhin, an welcher Stelle noch nicht alle Möglichkeiten der Vererbung genutzt wurden.
3. Mache einen Vorschlag zur Umsetzung weiterer Möglichkeiten und führe ihn durch.

2. Aufgabe

1. Schreibe zu unserem Projekt "**02-Raumplaner-mit-Vererbung**" eine Klasse Schrank, dessen grafische Darstellung rechts zu sehen ist. Der Schrank soll eine Breite von 60 und eine Tiefe von 37 haben. 
2. Ergänze dieses Projekt um eine Klasse Schrankwand, bestehend aus drei solchen Schränken der vorigen Aufgabe, deren grafische Darstellung unten zu sehen ist.



Zweiter Abschnitt

Kommentar und Lösung zum ersten Abschnitt:

Die Transformation der Figur wird eingesetzt, um unterschiedliche Positionen und Orientierungen zu ermöglichen.

Die hierhin ausgegliederte Methode bekommt die "Rohversion" der Figur (→ Shape) übergeben, verschiebt und dreht sie durch Affine Transformationen und gibt das so entstandene Shape zurück.

```
/**
 * transformiert das aktuelle Shape.
 */
protected Shape transformiere(Shape shape){
    AffineTransform t = new AffineTransform();
    t.translate(xPosition, yPosition);
    Rectangle2D umriss = shape.getBounds2D();
    t.rotate(Math.toRadians(orientierung), umriss.getX()
+umriss.getWidth()/2, umriss.getY()+umriss.getHeight()/2);
    return t.createTransformedShape(shape);
}
```

Die Methode `gibAktuelleFigur()` – hier gleich am Beispiel der Klasse `Schrank` gezeigt – muss am Ende noch entsprechend modifiziert werden:

```
protected Shape gibAktuelleFigur() {
    // einen GeneralPath definieren
    GeneralPath schrank = new GeneralPath();
    Rectangle2D rahmen = new Rectangle2D.Double(0, 0, breite, tiefe);
    schrank.append(rahmen, false);
    Line2D diagonale1 = new Line2D.Double(0, 0, breite, tiefe);
    schrank.append(diagonale1, false);
    Line2D diagonale2 = new Line2D.Double(0, tiefe, breite, 0);
    schrank.append(diagonale2, false);
    return transformiere(schrank);
}
```

Neue Gruppenaufgabe:

Unsere Unterrichtsbesprechung zu den Lösungen für die Schrankwand ergab, dass es nicht sinnvoll ist, einfach den Programmcode aus `Schrank` dreimal nach `Schrankwand` zu kopieren.

- Wiederholen Sie noch einmal die Überlegungen und schreiben Sie ihre Erkenntnisse auf.
- Überlegen Sie sich -ohne am Computer zu arbeiten- wie Sie dies umsetzen wollen.
- Versuchen Sie zu Lösungen zu kommen und probieren sie dabei möglichst verschiedenen Varianten aus.