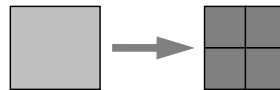


1. Geometrische Figuren (8 Punkte)

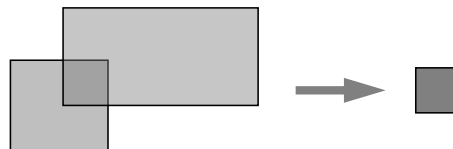
Implementieren Sie die Klasse `Rechteck` zur Darstellung zweidimensionaler, achsenparalleler Rechtecke. (d.h. sie können von zwei Punkten aufgespannt werden). Diese Punkte sollen mit Hilfe einer Klasse `Punkt` dargestellt werden. Die Koordinaten sind ganzzahlige Wert!

Für die Klasse `Rechteck` sind die folgenden Operationen zu realisieren:

- Erzeugen von Rechtecken, die an beliebigen Positionen stehen können
- Verschieben eines Rechtecks
- Drehen um 90 Grad, wobei die Ecke, um die gedreht werden kann angegeben werden muss.
- Prüfen auf Quadrat-Eigenschaft
- Bestimmen des Umkreises eines Rechtecks
- „Zoomen“ eines Rechtecks: Unter der Angabe von einem Faktor werden sowohl die Länge als auch die Breite vergrößert bzw. verkleinert.
- Teilen eines Rechtecks in vier gleiche große Rechtecke



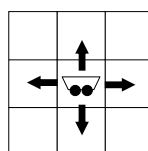
- Bestimmen der Schnittfläche zweier Rechtecke. Als Resultat soll diese Schnittfläche als neues Rechteck zurückgegeben werden



2. Simulationsumgebung (5 Punkte)

Erweitern Sie Ihren `Trolley`, um die folgenden Operationen zur Verfügung stellt zu können:

- Bewegen innerhalb des Verteilzentrums (Das Verteilzentrum ist als rechteckiges Netz von immer gleich großen Zellen aufgebaut)

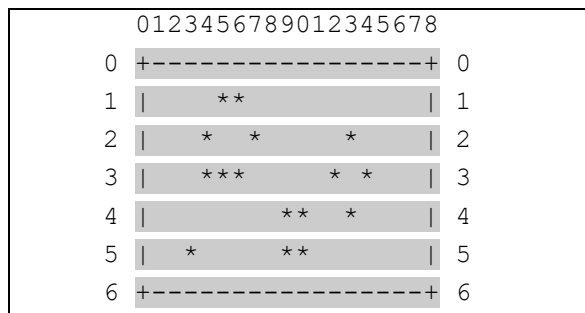


- Laden eines Produktes
 - Das erst geladene Produkt definiert das Ziel, das der Trolley ansteuert. Es dürfen nur Produkte dazu geladen werden, die das gleiche Ziel haben.
- Entladen aller Produkte: Am Bildschirm soll eine Liste der ausgeladenen Produkte ausgegeben werden.

Überlegen Sie sich für Ihr Design das Konzept eines „Transportauftrags“. Auch die Definition eines sinnvollen Aufbaus eines Verteilzentrums ist Teil dieser Aufgabe!

3. Qualitätskontrolle (7 Punkte)

Prof. Mayar hat vom Schweizer Ministerium für Käse und sonstige Milchprodukte (MfKsM) einen wichtigen Auftrag erhalten. Er soll ein System zur Qualitätskontrolle von Emmentaler Käse entwickeln. Es geht darum, die Anzahl der Löcher und den Umfang des größten Loches zu ermitteln.



Anzahl der Löcher: 4

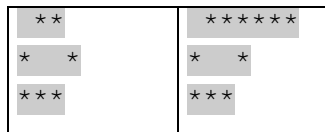
Umfang des größten Loches: 7

Beachten Sie die folgenden Vereinfachungen:

- Löcher grenzen nicht direkt aneinander
- Löcher liegen nicht direkt am Rand
- Die Ränder der Löcher sind nicht „ausgefranst“ d.h. es besteht immer ein geschlossener, eindeutiger Pfad:

Gültig

Ungültig (und kommt daher nicht vor!)



Implementieren Sie einen entsprechenden Algorithmus.

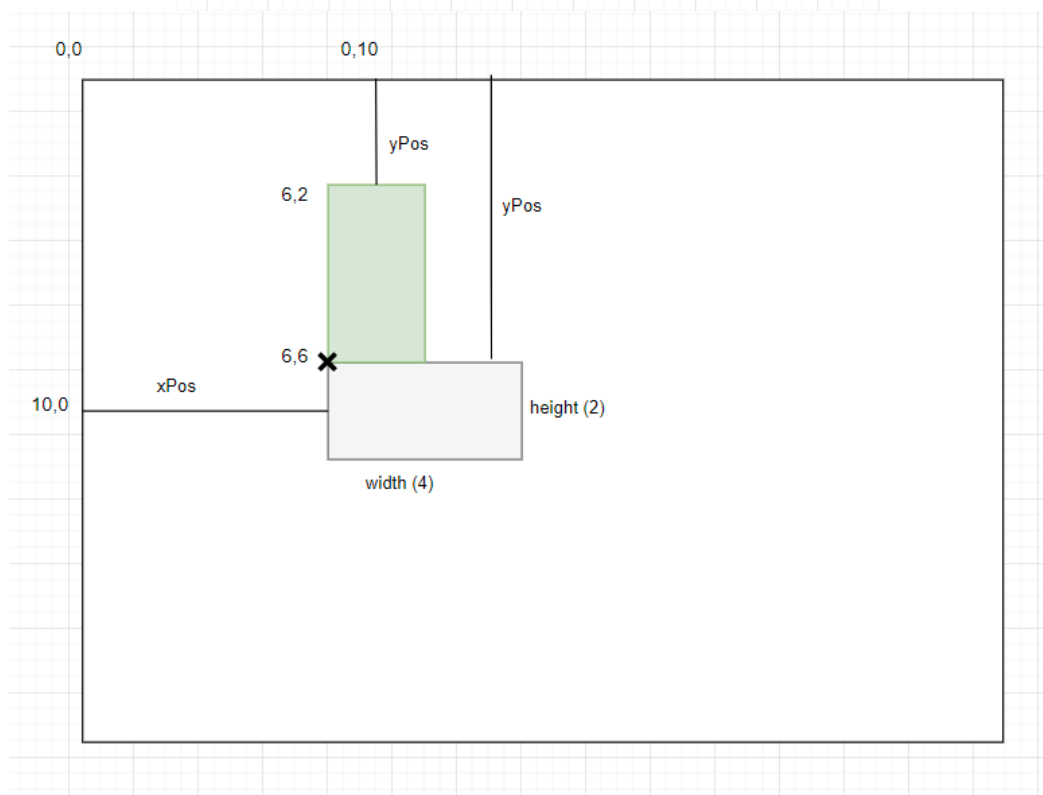
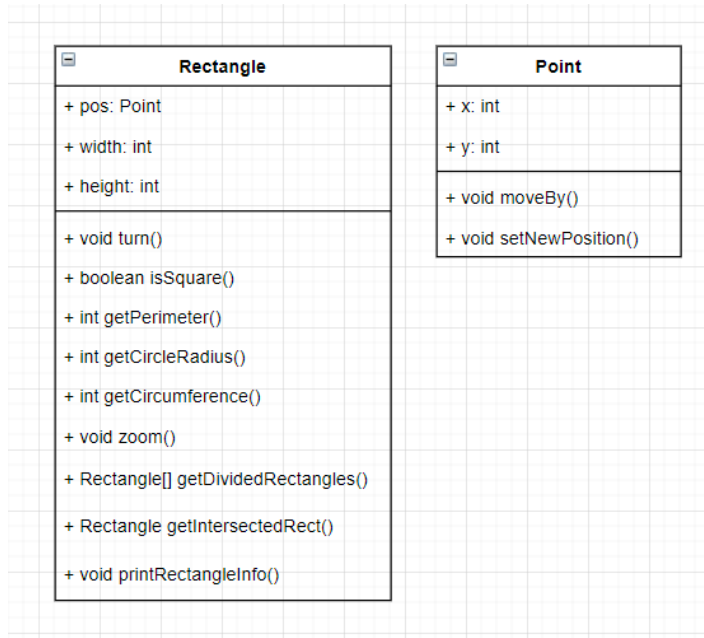
Hinweise

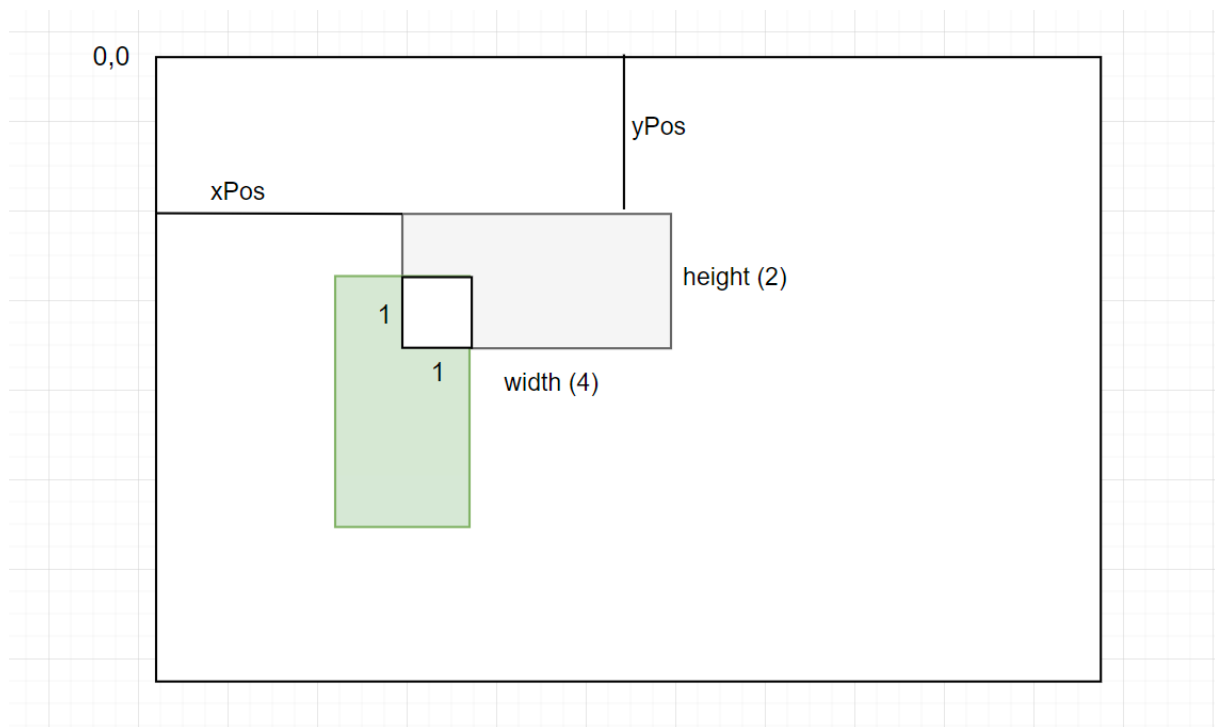
Vergessen Sie nicht auf die Lösungsidee und testen Sie Ihre Programme ausführlich und automatisiert!

1 Geometrische Figuren

Ziel ist es eine eigene Rectangle klasse zu implementieren. Diese Klasse bietet verschiedene geometrische Methoden um das Rechteck zu bearbeiten.

Umls



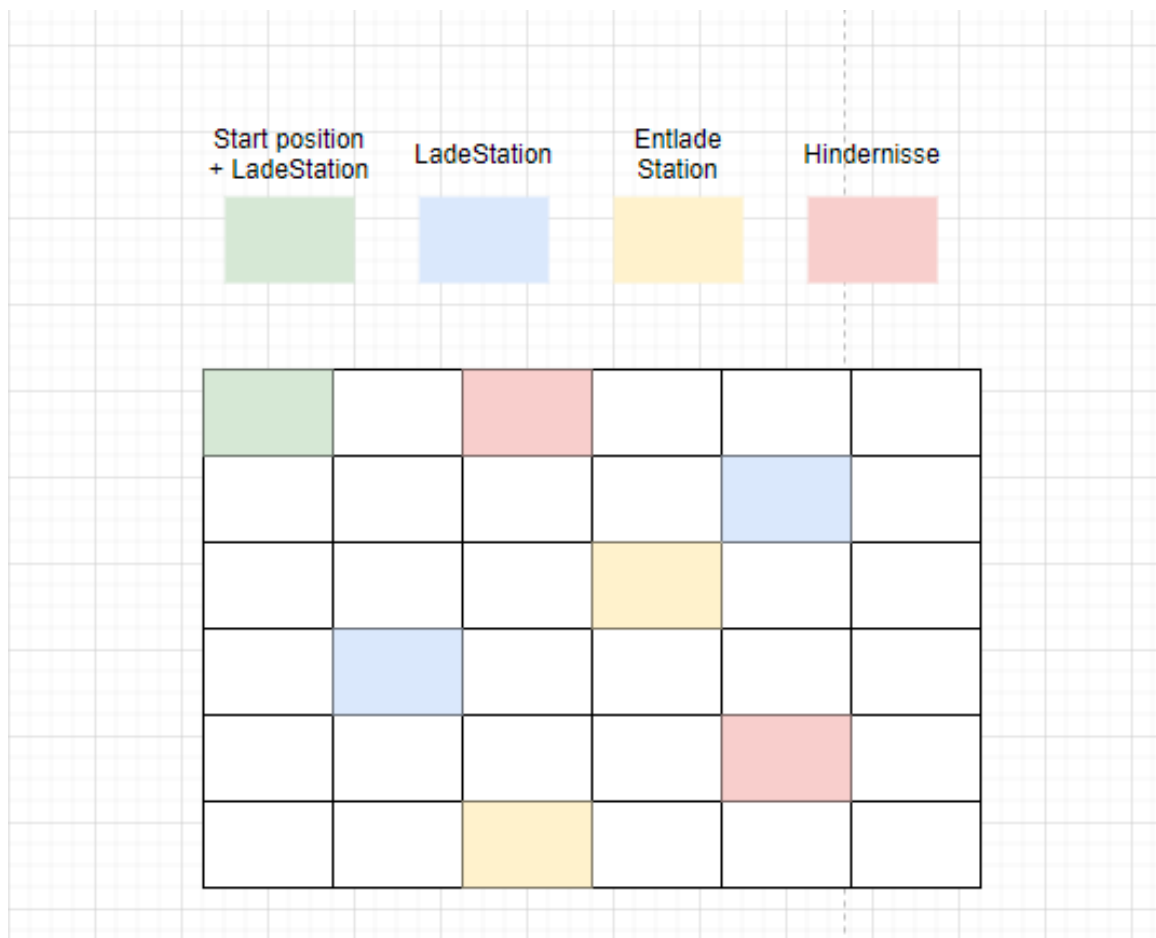
**Testfälle:**

siehe main.java file.

Java-Program code: Siehe beigelegtes files.

2 Simulationsumgebung

Ziel ist es eine Lieferung zu simulieren. Ein Trolley kann Produkte auf und abladen. Durch ein Grid kann sich der Trolley immer ein Feld in jede Himmelsrichtung bewegen. Dabei gibt es spezielle Lade/Entlade stationen.

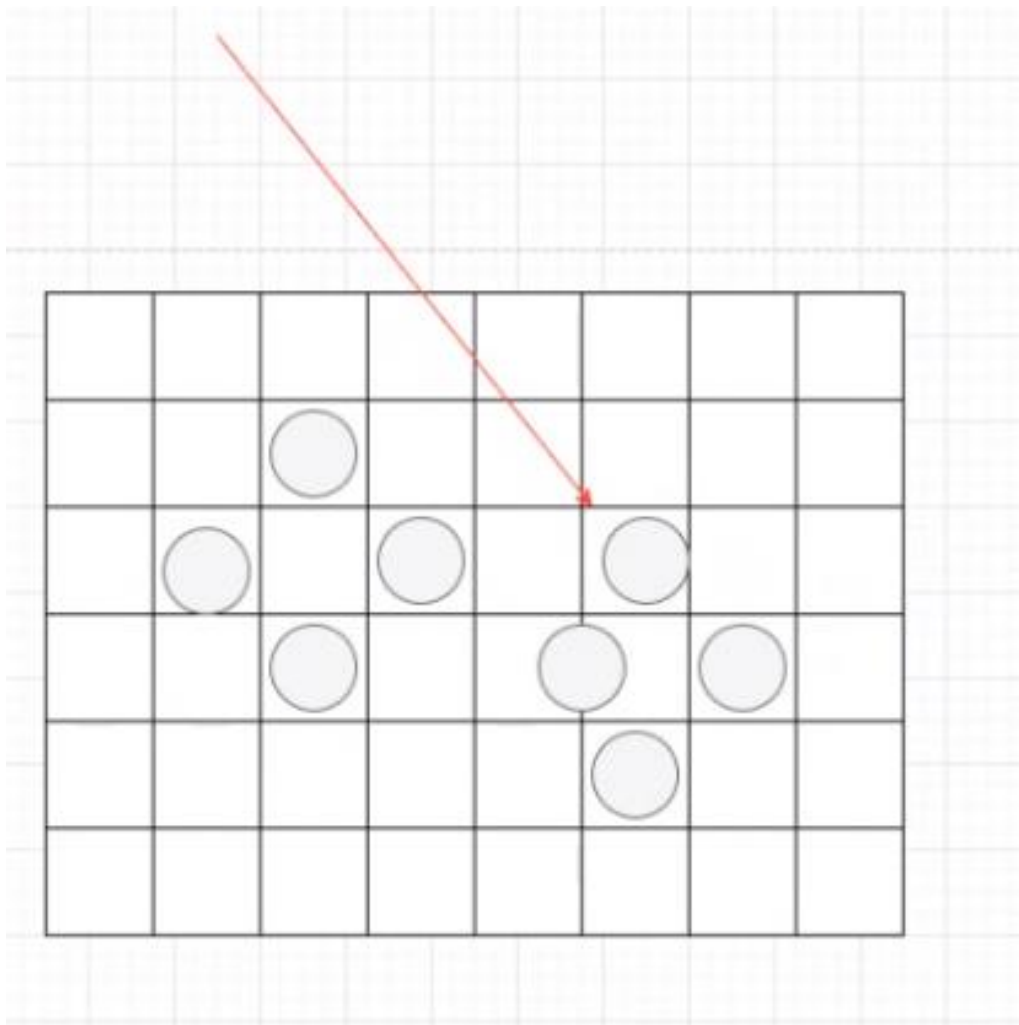
**Testfälle:**

siehe main.java file.

Java-Program code: Siehe beigelegtes files.

3 Qualitätskontrolle

Ziel ist es mithilfe einer char matrix löcher in form von stern kreise zu finden. Diese Kreise dürfen nicht direkt am Rand oder direkt beinander liegen. Wenn man ein stern gefunden hat muss man den sternern nachgehen und so das loch bestimmen.

**Testfälle:**

siehe main.java file.

Java-Program code: Siehe beigelegtes files.