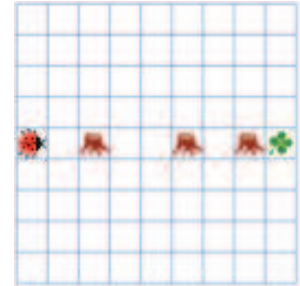


Informatik: Einführung in Java Play it again, Kara! (Aufgaben)

Gierhardt

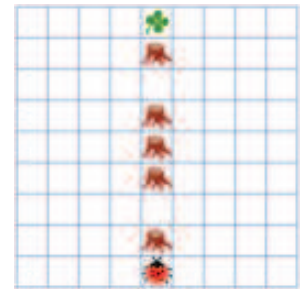
1. KARA soll ein Kleeblatt finden, das sich in der gleichen Zeile (oder Spalte) befindet wie er selbst. Zwischen ihm und dem Kleeblatt können Bäume stehen, wobei nie zwei Bäume direkt nebeneinander stehen.

Schreibe das Programm mit der Methode *umBaumHerum*.



2. KARA soll ein Kleeblatt finden, das sich in der gleichen Zeile (oder Spalte) befindet wie er selbst. Zwischen ihm und dem Kleeblatt können Bäume stehen, wobei mehrere Bäume hintereinander stehen können.

Schreibe das Programm mit einer Methode *umBaeumeHerum*.

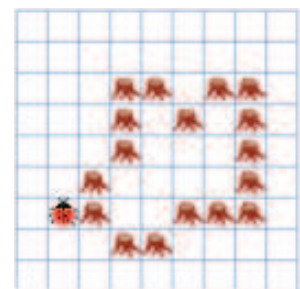


3. KARA sucht das Ende eines einfachen Labyrinths bestehend aus Bäumen, wobei keine Löcher in den Baumreihen auftreten. Das Ende des Labyrinths ist eine „Sackgasse“.



4. KARA bewacht eine zusammenhängende Fläche, die durch Bäume begrenzt ist. Er soll endlos außen an den „Wänden“ aus Bäumen entlang laufen.

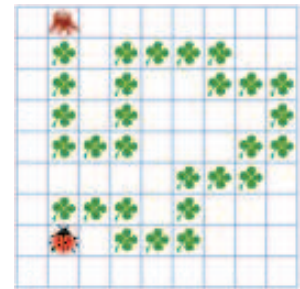
Variante: Er soll zuerst zu seiner Fläche und dann erst endlos darum herum laufen.



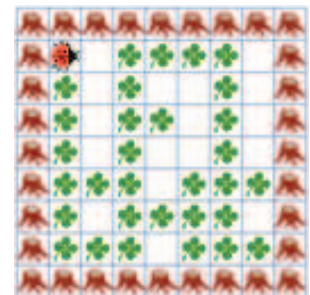
5. KARA spielt PACMAN: Er steht auf dem ersten Kleeblatt einer langen Spur von Kleeblättern, die vor einem Baum endet. Er soll alle Kleeblätter auffressen. Schwierige Variante: Die Kleeblattspur verläuft zwischen Bäumen und das Ende der Spur wird durch einen Pilz markiert.

Das Ende des Programms:

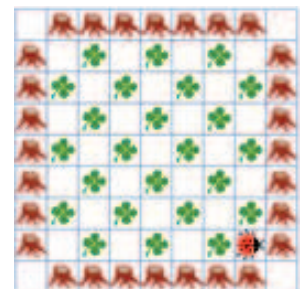
```
tools.showMessage("Ich bin so satt, \n"+
                  "ich mag kein Blatt!");
```



6. Eine quadratische Fläche ist von Bäumen umrandet. Innerhalb der Fläche ist ein Muster aus Kleeblättern gelegt, das von KARA invertiert werden soll. KARA startet links oben in der Ecke mit Blick nach rechts.



7. Eine quadratische Fläche ist von Bäumen umrandet. Innerhalb der Fläche soll ein schachbrettartiges Muster aus Kleeblättern von KARA gelegt werden. KARA startet links oben in der Ecke mit Blick nach rechts.



Hilfe: Bei den letzten beiden Aufgaben ist es hilfreich, mit booleschen Variablen zu arbeiten.

```
boolean fertig = false; // Deklaration und Initialisierung

fertig = !fertig;      // Aus true wird false und umgekehrt

if (...) fertig = true; // Direkte Zuweisung

if (fertig) {...}     // Boolesche Variable als Bedingung
```