

Softwareentwicklung 1 – Praktikum

Kontrollstrukturen und numerische Datentypen

Abgabe bis 22.11.2016, 23:00 Uhr

Ziele

- Programmieren mit Kontrollstrukturen (if/else, while und for)
- Verwenden von numerischen Datentypen (Schleifen und Berechnungen)
- Lernen eines sauberen Programmierstils durch Nutzung von Methoden

Hinweis

Sie finden für jede der Aufgaben ein vorbereitetes Greenfoot Szenario in meinem GitHub Projekt **karatest**. Für jedes Aufgabenblatt können Sie die jeweils aktuelle Version der Aufgaben in einem Archiv herunterladen:

<https://github.com/uhafner/karatest/archive/assignment3.zip>

Bitte nutzen Sie immer diese vorgegebenen Szenarien zur Lösung der Aufgaben. Verwenden Sie insbesondere immer die dort bereits enthaltene Aufgabendateien für Ihre Lösung, nur so können wir Ihre Lösungen automatisiert auswerten. Ebenso dürfen Sie die Verzeichnisstruktur nicht umändern.

Erlaubte Elemente zur Umsetzung

Zur Lösung der Aufgaben dieses Blattes dürfen alle in der Vorlesung behandelten Java Sprachmittel sowie die 10 vorgestellten Kara Methoden und Sensoren (move(), turnRight(), turnLeft(), putLeaf(), removeLeaf(), isTreeFront(), isTreeLeft(), isTreeRight(), isMushroomFront(), isOnLeaf()) genutzt werden.

Achtung: Die Nutzung von Fields (Instanzvariablen) ist nicht erlaubt.

Beliebige Welten

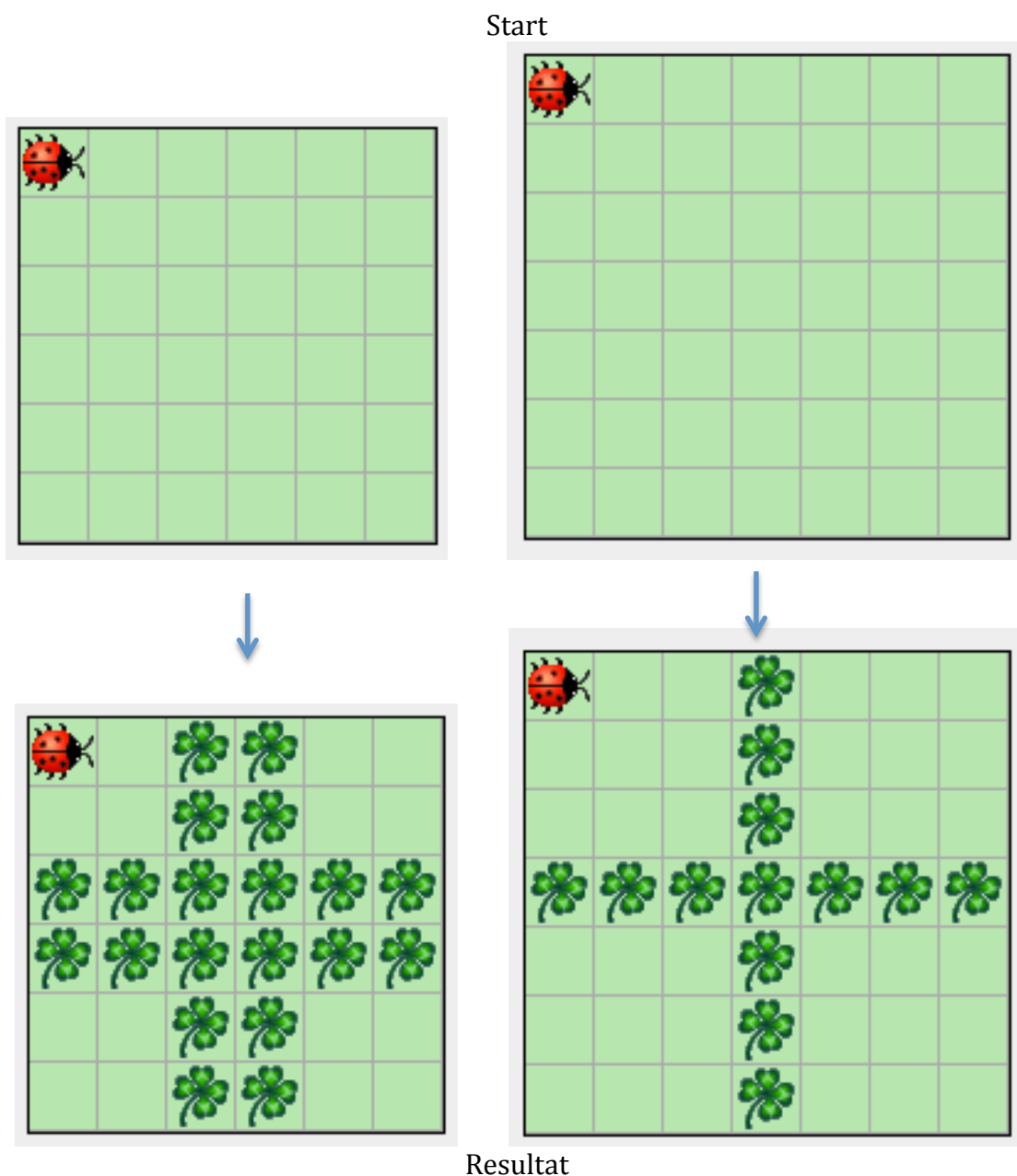
Beachten Sie, dass in allen Aufgaben dieses Blattes beliebige Welten (gemäß der Aufgabenstellung) zulässig sind. D.h. eine Aufgabe ist nur teilweise gelöst, wenn Ihr Programm nur für die im PDF abgedruckte Variante funktioniert. Nutzen Sie deshalb zum Testen eigene Welten (und die im Szenario und in den Tests vorgegebenen Welten).

Aufgabe 8 (ca. 8 Punkte)

Schreiben Sie ein Programm für Kara, das in einer beliebig großen leeren quadratischen Welt (Breite und Höhe jeweils identisch, mindestens jedoch 3) ein Kreuz zeichnet. Kara startet dabei in der linken oberen Ecke mit Blick nach rechts. Nach dem Ende des Programms soll Kara wieder in der Ausgangsposition stehen.

Achtung: Testen Sie Ihr Programm mindestens an den vorgegebenen Welten.

Beachten Sie, dass das Kreuz bei ungerader Breite und Höhe mit einer „Strichstärke“ von Eins, bei gerader Breite und Höhe mit einer „Strichstärke“ von Zwei gezeichnet werden soll.



Aufgabe 9 (ca. 8 Punkte)

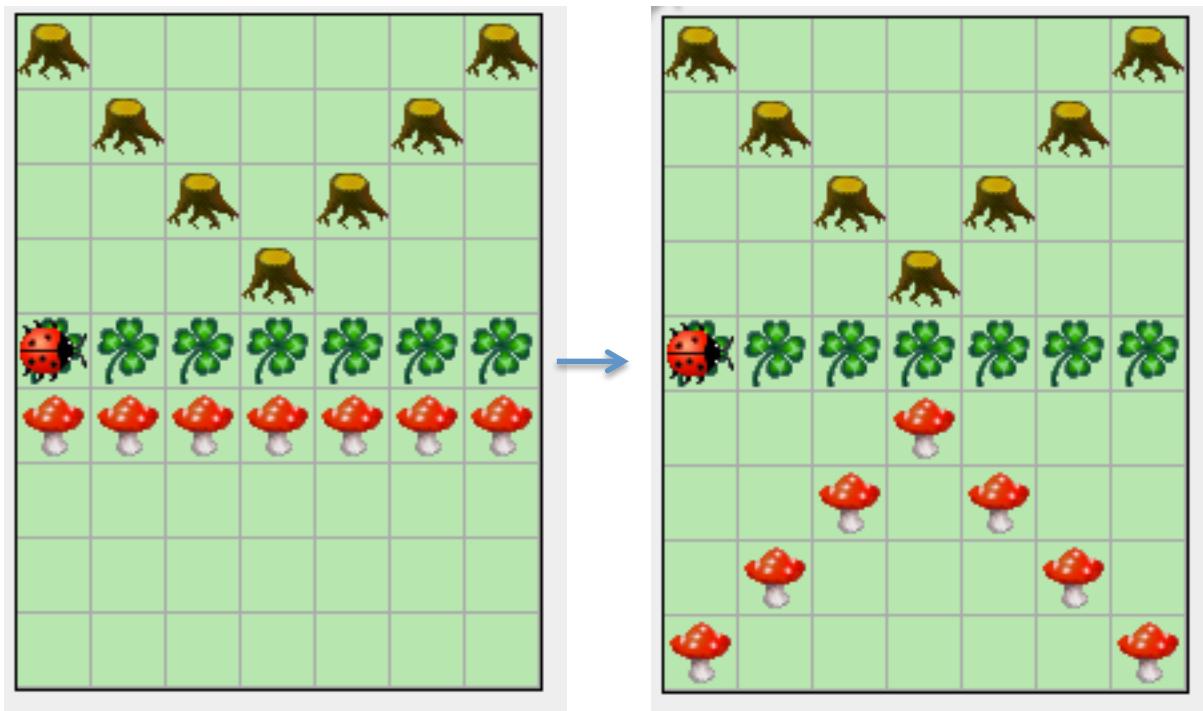
Schreiben Sie ein Programm für Kara, das ein Baummuster mit Pilzen (gespiegelt) nach malt. Die Ausgangswelt ist nach folgendem Muster aufgebaut:

- Die Welt hat eine ungerade Anzahl an Zeilen, in der Mitte findet sich eine Spur aus Blättern. Darüber befindet sich ein Muster aus Bäumen, darunter eine komplette Zeile mit Pilzen.
- In jeder Spalte befinden sich genau ein Blatt, ein Baum (darüber) und ein Pilz (darunter).
- Die Breite der Welt ist beliebig.

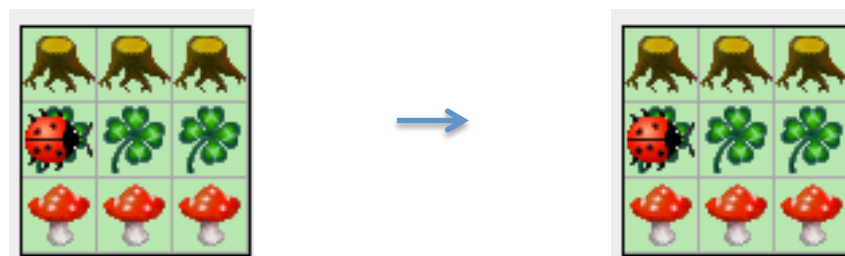
Kara startet dabei auf der linken Seite der Blätterspur mit Blick nach rechts. Nach dem Ende des Programms soll Kara wieder in der Ausgangsposition stehen, die Blätter sind unverändert. Die Pilze zeigen das an der Blätterlinie gespiegelte Muster dar.

Achtung: Testen Sie Ihr Programm mindestens an den vorgegebenen Welten.

Start



Resultat



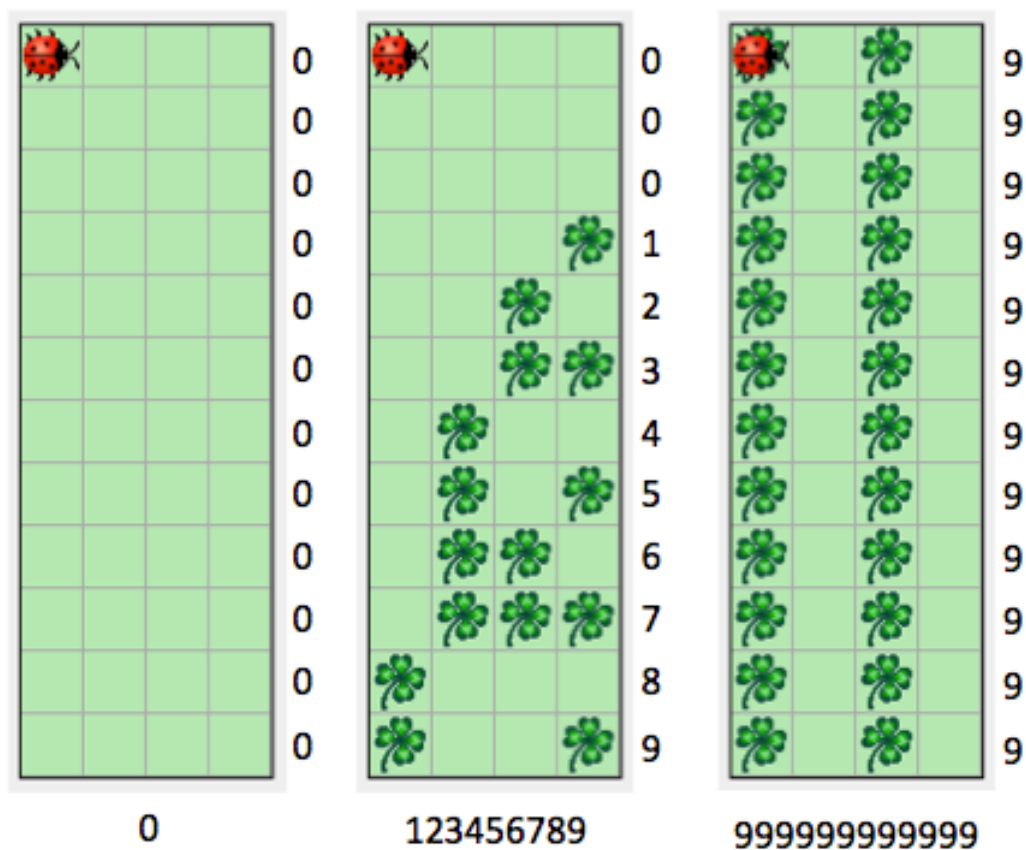
Aufgabe 10 (ca. 6 Punkte)

Schreiben Sie ein Programm, das Karas Welt zum Dekodieren von Zahlen verwendet. In diesem Programm wird die Welt von Kara nicht verändert – eine Welt dient jeweils zur Dekodierung einer Zahl. Ihre Aufgabe besteht darin, die in dieser Welt in Form von abgelegten Blättern dargestellte Zahl zu dekodieren. Laufen Sie daher von links nach rechts (und von oben nach unten) durch die gesamte Welt und geben Sie am Ende Ihres Programms die ermittelte Zahl aus. Dazu können Sie den folgenden Java Befehl nutzen:

```
showMessage("Gesuchte Zahl = " + number);
```

Die Variable **number** muss in diesem Fall die ermittelte Zahl beinhalten. Bitte keine weiteren Ausgaben machen, da diese Meldung automatisch ausgewertet wird, um die Korrektheit Ihrer Lösung zu überprüfen.

Eine Welt ist immer gleich aufgebaut: sie besteht aus 12 Zeilen mit jeweils 4 Spalten. Damit wird jeweils eine 12 stellige Zahl kodiert. Die erste Zeile steht für die erste Dezimalziffer, die zweite Zeile für die zweite, usw. Die vier Spalten einer Zeile repräsentieren die einzelnen Dezimalziffern: jede Ziffer von 0 bis 9 ist als [Binärzahl](#) kodiert. Die 1. Spalte von links speichert das höherwertigste Bit, die 1. Spalte von rechts das niederwertigste. D.h. eine Ziffer wird durch die Formel $b_1 * 2^3 + b_2 * 2^2 + b_3 * 2^1 + b_4 * 2^0$ dargestellt. b_1 ist 1, falls die 1. Spalte ein Blatt enthält, sonst 0, b_2 ist 1 falls die 2. Spalte ein Blatt enthält, usw. (Prinzipiell lassen sich damit 16 Werte darstellen, in unserem Fall können Sie aber davon ausgehen, dass nur die Zahlen 0-9 vorkommen).



Aufgabe 11 (ca. 6 Punkte)

Schreiben Sie das zu Aufgabe 10 passende Programm, das eine vorgegebene Zahl mit Blättern in Karas Welt kodiert (Umkehrfunktion zu Aufgabe 10). D.h. Sie bekommen als Eingabe ein maximal 12-stellige Zahl, die Sie als Blattmuster mit Kara zeichnen. Nutzen Sie zur Eingabe der Zahl folgende Anweisung:

```
long value = readLong("Zahl eingeben:");
```

Verwenden Sie die Beispiele aus Aufgabe 10 zum Testen.