

Softwareentwicklung 1 – Praktikum

Erste Schritte mit Kara

Abgabe bis 25.10.2014, 23:00 Uhr

Ziele

- Erste Schritte in Java mit der Entwicklungsumgebung Greenfoot
- Erstellen einfacher Programme durch Aneinanderreihen von Anweisungen
- Nutzen von Methoden zur Vereinfachung und Verkürzen von Programmen

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/assignment1.zip>

Bitte nutzen Sie immer diese vorgegebenen Szenarien zur Lösung der Aufgaben. Verwenden Sie insbesondere immer die dort bereits enthaltene Aufgabendatei (z.B. Assignment1.java) für Ihre Lösung, nur so können wir Ihre Lösungen automatisiert auswerten. Bitte packen Sie Ihre Lösungen in ein Archiv und laden Sie dieses **rechtzeitig** in Moodle hoch, es genügt dabei die jeweilige Aufgabendatei (z.B. Assignment1.java), nicht das gesamte Szenario. Es hat sich bewährt, dies nicht erst 1 Minute vor der Abgabe zu machen, laden Sie daher zur Sicherheit bereits schon Zwischenergebnisse hoch! Sie dürfen Zwischenergebnisse beliebig oft überschreiben, es zählt die zuletzt hochgeladene Version.

Erlaubte Elemente zur Umsetzung

Zur Lösung der Aufgaben dieses Blattes dürfen nur die 5 vorgestellten Kara Methoden `move()`, `turnRight()`, `turnLeft()`, `putLeaf()`, `removeLeaf()` oder eigenen Methoden aufgerufen werden. Java Kontrollstrukturen wie z.B. Schleifen sind noch nicht erlaubt.

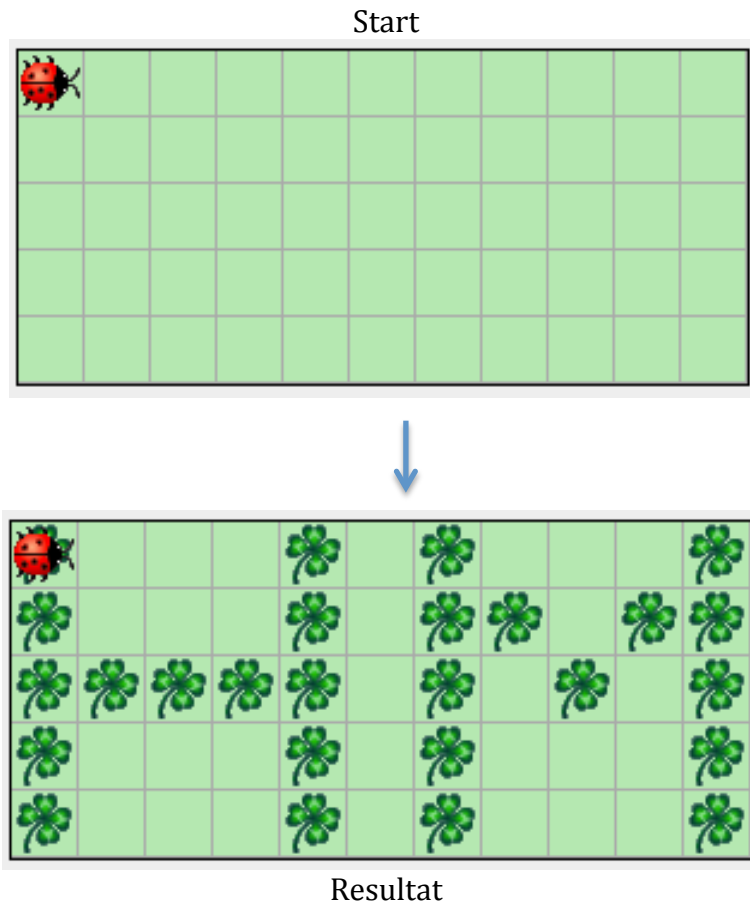
Nützlich Links

- <http://www.greenfoot.org/>
- <http://www.swisseduc.ch/informatik/karatojava/greenfootkara/index.html>
- <http://code.makery.ch/library/greenfoot-kara/>

Bitte unterschreiben Sie außerdem die Eigenständigkeitserklärung und geben Sie diese rechtzeitig im Praktikum oder vor der Vorlesung ab!

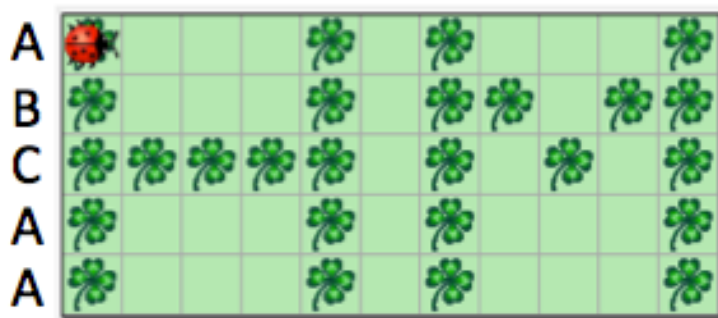
Aufgabe 1 (ca. 4 Punkte)

Schreiben Sie ein Programm für Kara, das in der vorgegebenen leeren Welt von 11x5 Kästchen die Initialen „HM“ unserer Hochschule in Form von Kleeblättern schreibt. Kara startet dabei in der linken oberen Ecke mit Blick nach rechts. Nach dem Ende des Programms soll Kara wieder in der Ausgangsposition stehen.



Aufgabe 2 (ca. 6 Punkte)

Verbessern Sie Ihr Programm aus Aufgabe 1, so dass Sie die Anzahl der **move()** Anweisungen minimieren. Nutzen Sie dabei die Tatsache, dass das Resultat aus Zeilen besteht, von denen einige identisch sind (siehe Abbildung). Schreiben Sie für jede unterschiedliche Zeile (A, B, C) eine Methode und rufen Sie diese Methoden in der richtigen Reihenfolge auf, bis wieder die Abkürzung HM zu sehen ist. Nutzen Sie auch die Möglichkeit, dass Kara beim Überschreiten des Rands wieder auf der gegenüberliegenden Seite erscheint! Lässt sich das Programm noch weiter kürzen? Versuchen Sie weniger als 40 Anweisungen zu verwenden.

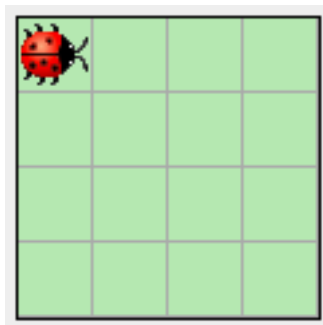


Resultat

Lässt sich Ihre Lösung durch Aufteilung in Spalten statt Zeilen optimieren? Sehen Sie weitere Einsparmöglichkeiten?

Aufgabe 3 (ca. 8 Punkte)

Schreiben Sie ein Programm, das ein Schachbrettmuster in einer leeren 4x4 Welt erzeugt. Schaffen Sie es, mit weniger als 15 Anweisungen auszukommen?



Start



Resultat

Aufgabe 4 (optional, ca. 4 Bonuspunkte)

Karas Sensoren und Fallunterscheidungen werden erst in der Vorlesung am 14.10. behandelt. Wenn Sie diese dennoch schon ausprobieren wollen, können Sie folgende Zusatzaufgabe lösen. Ändern Sie Ihr Programm aus Aufgabe 2 so ab, dass es auch funktioniert, wenn Karas Welt zu Beginn nicht leer ist, sondern bereits Blätter enthält. Zur Lösung können Sie den Sensor `isOnLeaf()` in einer Fallunterscheidung (`if`) nutzen.

Eine Ausgangswelt ist im Szenario 4 schon enthalten. Überlegen Sie sich weitere Ausgangswelten, für die Ihre Lösung außerdem funktionieren muss.

Aufgabe 5 (optional, erspart Arbeit und hilft für spätere Semester)

Die von mir vorbereiteten Szenarios sind in [GitHub](#) in der Versionsverwaltung [git](#) abgelegt. Ändere ich Szenarien, bzw. lade ich neue Szenarien für weitere Übungsaufgaben hoch, erscheinen diese automatisch in meinem Projekt [KaraTest](#). Sie können sich das mehrmalige Herunterladen der Aufgaben sparen, wenn Sie meinem Projekt folgen. Voraussetzung ist die Installation von git auf Ihrem Rechner. Gehen Sie folgendermaßen vor:

1. Installieren Sie git
2. Konfigurieren Sie git
3. Öffnen Sie ein Terminal
4. Clonen Sie mein Repository, indem Sie im Terminal folgenden Befehl ausführen:
`git clone https://github.com/uhafner/karate.git`
5. Öffnen Sie das Projekt mit:
`cd karatest`
6. Springen Sie in die Aufgaben des 1. Blattes mit:
`git checkout assignment1`
7. Lösen Sie die Aufgaben wie bisher.
8. Speichern Sie Ihre geänderten Dateien mit:
`git commit -a`
9. Nach Freigabe des Ausgabenblattes 2: aktualisieren Sie Ihre lokale Kopie mit:
`git pull`
10. Springen Sie in die Aufgaben des 2. Blattes mit:
`git checkout assignment2`