

Verzweigungs-Expertin

Fragen

1. Betrachten wir untenstehendes Programm

```
(...)  
while (!kara.treeFront())  
{  
    if (kara.onLeaf())  
    {  
        kara.removeLeaf();  
    }  
    else  
    {  
        kara.putLeaf();  
    }  
    kara.move();  
}  
(...)
```

Im Gegensatz zur Lösung 1 wurde die Anweisung `kara.move()`; hinter den `if`-Block gestellt. Ändert sich dadurch die Funktionalität des Programmes?

Lösung

Grundsätzlich bleibt die Funktionalität gleich: Das Programm invertiert die Felder bis zum nächsten Baum. Es gibt aber einen Unterschied: Das obige Programm berücksichtigt das letzte Feld vor dem Baum nicht.

Warum nicht? Das Programm prüft, ob Kara vor einem Baum steht. Falls nicht, invertiert das Programm das aktuelle Feld. Anschliessend macht Kara einen Schritt vorwärts. Steht er jetzt vor einem Baum, so ist die `while`-Schleife beendet.

2. Das Programm auf der ersten Seite arbeitet nicht in allen Situationen korrekt. Hier noch einmal das Programm:

```
while (!kara.treeFront())  
{  
    kara.move();  
    if (kara.onLeaf())  
    {  
        kara.removeLeaf();  
    }  
    else  
    {  
        kara.putLeaf();  
    }  
}
```

Puzzle: Expertin A

- a. Versuche den Spezialfall zu finden, bei welchem das Programm nicht richtig arbeitet.

Lösung

Wenn Kara zu Beginn vor dem Baum steht, ist die Bedingung für die while-Schleife nicht erfüllt. Das heisst aber auch, das Feld vor dem Baum wird in diesem Fall nicht invertiert.

- b. Entwirf auf Papier eine verbesserte Version des Programms, die auch den Spezialfall berücksichtigt.
- c. Teste am Computer, ob deine Lösung funktioniert. *Hinweis:* Teste dein Programm für Spezialfälle: Kara steht zu Beginn auf einem Kleeblatt, oder vor dem Baum liegt ein Kleeblatt, etc.

Lösung für c und d

Das Problem ist ein Klassiker und tritt in vielen Varianten immer wieder auf. Die Lösung ist, dass die Anweisungen für das Invertieren zwei Mal vorkommen müssen: Vor der Schleife für das erste Feld, und innerhalb der Schleife nach dem Vorwärtsschritt für alle weiteren Felder. Der besseren Lesbarkeit halber verwendet die Lösung eine Methode für das Invertieren.

```
void invertiere()
{
    if (kara.onLeaf())
    {
        kara.removeLeaf();
    }
    else
    {
        kara.putLeaf();
    }
}

public void myProgram()
{
    invertiere();
    while (!kara.treeFront())
    {
        kara.move();
        invertiere();
    }
}
```

Puzzle: Expertin A

Eine andere Möglichkeit: Innerhalb der Schleife wird zuerst invertiert, dann folgt der Vorwärtsschritt. Nach der Schleife muss das Feld vor dem Baum invertiert werden.

```
public void myProgram()
{
    while (!kara.treeFront())
    {
        invertiere();
        kara.move();
    }
    invertiere();
}
```

3. Kara steht in einer Welt ohne Bäume. Er soll nun einen Schritt nach vorne machen. Liegt dort ein Kleeblatt, so soll Kara es aufnehmen, sich nach links drehen und einen Schritt nach vorne machen. Ist auf dem Feld kein Kleeblatt, so soll Kara ein Kleeblatt hinlegen, sich nach rechts drehen und einen Schritt nach vorne machen.

Schreibe ein JavaKara Programm, das obige Aufgabe erfüllt und teste es anschliessend am Computer.

Lösung

```
kara.move();
if (kara.onLeaf())
{
    kara.removeLeaf();
    kara.turnLeft();
}
else {
    kara.putLeaf();
    kara.turnRight();
}
```

Werden die Anweisungen endlos wiederholt, ist das Programm die „chaotische Ameise“ (siehe Aufgaben in Kara). Lässt man es in einer grossen Welt (100x100) lange genug laufen, entstehen aus dem Chaos plötzlich geordnete Bahnen...

```
while (true) // ein kleiner Trick, um etwas endlos zu machen
{
    kara.move();
    if (kara.onLeaf())
    {
        kara.removeLeaf();
        kara.turnLeft();
    }
    else {
        kara.putLeaf();
        kara.turnRight();
    }
}
```

Puzzle: Expertin A

4. Wozu könnte folgendes Programmschema gut sein? Hinweis: Stell dir vor, du musst dir die Ampelfarbe einer Lichtsignalanlage anschauen.

```
(...)  
if (Bedingung1) { Anweisungen1 }  
else {  
  if (Bedingung2) { Anweisungen2 }  
  else {  
    if (Bedingung3) { Anweisungen3 }  
    else { Anweisungen4 }  
  }  
}  
(...)
```

Lösung

Dieses Schema wird, wenn nur die Anweisungen für eine einzige Bedingung ausgeführt werden sollen. Dabei „gewinnt“ die Bedingung, die als erste erfüllt ist.

Eine andere, gleichwertige Schreibweise mit weniger Verschachtelung wäre:

```
(...)  
if (Bedingung1) { Anweisungen1 }  
else if (Bedingung2) { Anweisungen2 }  
else if (Bedingung3) { Anweisungen3 }  
else { Anweisungen4 }  
(...)
```

Aber Achtung: Folgende Anweisungen haben eine andere Bedeutung:

```
(...)  
if (Bedingung1) { Anweisungen1 };  
if (Bedingung2) { Anweisungen2 };  
if (Bedingung3) { Anweisungen3 };  
(...)
```

Hier werden die Anweisungen **aller** erfüllten Bedingungen ausgeführt!

5. Knifflige Zusatzaufgabe:

Kara soll folgendes tun, wenn er auf einem Kleeblatt steht:

- Zuerst soll er das Kleeblatt aufnehmen.
- Ist vor ihm ein Baum, dann soll Kara eine Drehung um 180° machen.
- Ist vorne kein Baum, dann soll Kara ein Feld nach vorne schreiten.

Steht Kara auf keinem Kleeblatt, soll er nichts tun.

Obige Aufgabe wurde bereits programmiert. Leider befindet sich ein Fehler im Programm.

```
(...)  
if (kara.onLeaf()) {  
  // Kara ist auf Kleeblatt: Also aufnehmen  
  kara.removeLeaf();  
  if (kara.treeFront()) {  
    // Kara steht vor Baum, drehe um
```

Puzzle: Expertin A

```
        kara.turnRight();
        kara.turnRight(); }
    }
    else {
        // Kara steht nicht vor einem Baum
        kara.move();
    }
// und sonst mache gar nichts
(...)
```

Finde den Fehler und verbessere das Programm.

Lösung

Die oben rot markierte Klammer stimmt nicht. So wie die Klammern gesetzt sind, müsste das Programm wie folgt eingerückt werden:

```
(...)
if (kara.onLeaf()) {
    // Kara ist auf Kleeblatt: Also aufnehmen
    kara.removeLeaf();
    if (kara.treeFront()) {
        // Kara steht vor Baum, drehe um
        kara.turnRight();
        kara.turnRight();
    }
}
else {
    // Kara steht nicht vor einem Baum
    kara.move();
}
// und sonst mache gar nichts
(...)
```

Das Programm mit korrigierten Klammern:

```
if (kara.onLeaf()) {
    // Kara ist auf Kleeblatt: Also aufnehmen
    kara.removeLeaf();
    if (kara.treeFront()) {
        // Kara steht vor Baum, drehe um
        kara.turnRight();
        kara.turnRight();
    }
}
else {
    // Kara steht nicht vor einem Baum
    kara.move();
}
// und sonst mache gar nichts
(...)
```