

Expertin für Boole'sche Ausdrücke

Fragen

1. Wie müsstest du das dritte Programm ändern, wenn Kara nur Kleeblätter legen dürfte, wenn sich links *und* rechts ein Baumstamm befindet? Probiere deine Lösung auf dem Computer aus!

Hier noch einmal das Programm:

```
import JavaKaraProgram;
public class LegeKleeblaetterAufWeg3 extends JavaKaraProgram
{ // Anfang von LegeKleeblaetterAufWeg3
  public void myProgram()
  { // Anfang von myProgramm
    while (!kara.treeFront())
    {
      kara.move();
      if ( !kara.onLeaf() &&
          ( kara.treeLeft() || kara.treeRight() ) )
      {
        kara.putLeaf();
      }
    }
  } // Ende von myProgramm
} // Ende von LegeKleeblaetterAufWeg3
```

Lösung

Die Bedingung für das if muss angepasst werden; die ODER-Verknüpfung von „Baum links“ und „Baum rechts“ durch eine UND-Verknüpfung ersetzt werden:

```
if ( !kara.onLeaf() &&
    ( kara.treeLeft() && kara.treeRight() ) )
```

Damit sind nun aber die Klammern überflüssig geworden, denn jetzt müssen alle drei Bedingungen erfüllt sein. Wir können also schreiben:

```
if ( !kara.onLeaf() && kara.treeLeft() && kara.treeRight() )
```

Puzzle: Expertin B

2. Das dritte Programm soll erweitert werden. Nun soll ein Kleeblatt nur gelegt werden, wenn sich entweder nur links ein Baum oder nur rechts ein Baum befindet. Wenn sich links und rechts ein Baum befindet, dann soll nichts gemacht werden.

Probiere deine Lösung wiederum am Computer aus!

Hinweis: Die Formulierung des Boole'schen Ausdrucks ist nicht ganz leicht. Versuche die Lösung aus Aufgabe 3 mit deiner Lösung zur vorhergehenden Frage zu kombinieren.

Lösung

Die Bedingung für das if muss angepasst werden. Die Frage lautet, wie kann „entweder-oder“ ausgedrückt werden mit den Boole'schen Operationen „nicht“, „und“, „oder“? Umgangssprachlich formuliert können wir das entweder-oder wie folgt umschreiben: „wenn (links ein Baum und rechts kein Baum) || (links kein Baum und rechts ein Baum)“.

```
if ( !kara.onLeaf() &&
    ( ( kara.treeLeft() && !kara.treeRight() ) ||
      ( !kara.treeLeft() && kara.treeRight() ) ) )
```

Die Klammern, welche die beiden Baum-UND-Verknüpfungen zusammenhalten, sind nur der besseren Lesbarkeit halber eingefügt. Sie können auch weggelassen werden:

```
if ( !kara.onLeaf() &&
    ( kara.treeLeft() && !kara.treeRight() ||
      !kara.treeLeft() && kara.treeRight() ) )
```

Kompakter geht es, wenn man weiss, dass es in Java eine ENTWEDER-ODER-Verknüpfung gibt, ein sogenanntes „exklusives oder“, abgekürzt XOR:

```
if ( !kara.onLeaf() && (kara.treeLeft() ^ kara.treeRight() ) )
```

