

1. Finde jeweils zu den KARA-Bedingungen die negierte Form!

Bed.:	<code>kara.treeFront()    kara.treeRight()</code>
neg.:	<code>!kara.treeFront() &amp;&amp; !kara.treeRight()</code>
Bed.:	<code>kara.treeFront() &amp;&amp; kara.treeRight()</code>
neg.:	<code>!kara.treeFront()    !kara.treeRight()</code>
Bed.:	<code>kara.treeFront()    !kara.onLeaf()</code>
neg.:	<code>!kara.treeFront() &amp;&amp; kara.onLeaf()</code>
Bed.:	<code>!kara.treeLeft() &amp;&amp; kara.treeFront() &amp;&amp; !kara.treeRight()</code>
neg.:	<code>kara.treeLeft()    !kara.treeFront()    kara.treeRight()</code>

2. Drücke die auf Deutsch formulierten Anweisungen in der Sprache Java für KARA aus:  
 „Wenn vorne kein Baum, aber ein Pilz steht, so gehe einen Schritt vor und drehe nach rechts, sonst gehe einen Schritt nach links, falls dort kein Baum steht.“

```

1   if (!kara.treeFront() && kara.mushroomFront())
2       {
3           kara.move();
4           kara.turnRight();
5       }
6   else {
7       if (!kara.treeLeft())
8           {
9               kara.turnLeft();
10              kara.move();
11          }
12      }
    
```

3. Im folgenden Programm sind mehrere Schreib- und Flüchtigkeitsfehler versteckt, die eine erfolgreiche Kompilierung verhindern. Verbessere!

```
1 import javakara . JavaKaraProgram ;
2 public class SchlampigGeschrieben extends JavaKaraProgram
3 {
4     void einenSchrittWeiter
5     {
6         if (!kara . treeFront ())
7             { kara . move (); }
8         else {
9             if (!kara treeLeft ())
10                { kara . turnLeft ();
11                  kara . move ();
12                }
13            else { kara . turnRight ();
14                  kara . move ();
15                }
16        }
17    }
18
19    public void myProgram()
20    {
21        while ( !(kara . treeFront () & kara . treeLeft ()
22                && kara . treeRight () )
23            {
24                einenSchrittWeiter ();
25            }
26    }
27 } // Ende von SchlampigGeschrieben
```

4. KARA steht auf dem ersten Blatt einer Spur von Kleeblättern (mit Unterbrechungen). Am Ende der Spur steht ein Baum. KARA soll bis zum Baum laufen. Wenn er auf ein Kleeblatt trifft, soll er rechts daneben ein weiteres Kleeblatt ablegen. Benutze zur Lösung eine Methode *legeRechts()*, die KARA rechts von seiner aktuellen Position ein Kleeblatt ablegen und zurückkehren lässt.

```
1 import javakara . JavaKaraProgram ;
2 public class RechtsDanebenLegen extends JavaKaraProgram
3 {
4
5     void legeRechts ()
6     {
7         kara . turnRight () ;
8         kara . move () ;
9         kara . putLeaf () ;
10        kara . turnLeft () ;
11        kara . turnLeft () ;
12        kara . move () ;
13        kara . turnRight () ;
14    }
15
16    public void myProgram ()
17    {
18        if ( kara . onLeaf () ) { legeRechts () ; }
19        while ( ! kara . treeFront () )
20        {
21            kara . move () ;
22            if ( kara . onLeaf () ) { legeRechts () ; }
23        }
24    }
25 }
```