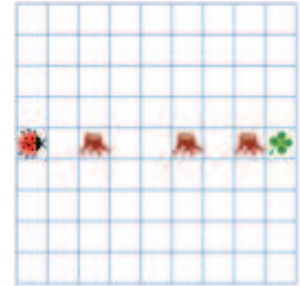


1. KARA soll ein Kleeblatt finden, das sich in der gleichen Zeile (oder Spalte) befindet wie er selbst. Zwischen ihm und dem Kleeblatt können Bäume stehen, wobei nie zwei Bäume direkt nebeneinander stehen.

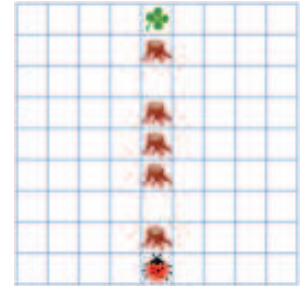
Schreibe das Programm mit der Methode *umBaumHerum*.



```
1 import javakara . JavaKaraProgram ;
2
3 public class Playit1 extends JavaKaraProgram
4 { // Anfang von Playit1
5
6     void geheZumBaum ()
7     {
8         while ( !kara . treeFront () && !kara . onLeaf () ) kara . move ();
9     }
10
11    void geheUmBaumHerum ()
12    {
13        kara . turnLeft ();
14        kara . move ();
15        kara . turnRight ();
16        kara . move ();
17        kara . move ();
18        kara . turnRight ();
19        kara . move ();
20        kara . turnLeft ();
21    }
22
23
24    public void myProgram ()
25    { // Anfang von myProgram
26        while ( !kara . onLeaf () )
27        {
28            geheZumBaum ();
29            if ( !kara . onLeaf () ) geheUmBaumHerum ();
30        }
31    }
32    } // Ende von myProgram
33
34 }
```

2. KARA soll ein Kleeblatt finden, das sich in der gleichen Zeile (oder Spalte) befindet wie er selbst. Zwischen ihm und dem Kleeblatt können Bäume stehen, wobei mehrere Bäume hintereinander stehen können.

Schreibe das Programm mit einer Methode *umBaeumeHerum*.

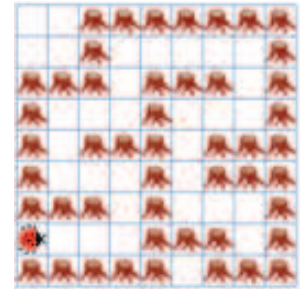


```

1 import javakara . JavaKaraProgram ;
2
3 public class Playit2 extends JavaKaraProgram
4 { // Anfang von Playit2
5
6     void geheZumBaum ()
7     {
8         while ( !kara . treeFront () && !kara . onLeaf () ) kara . move () ;
9     }
10
11    void geheUmBaeumeHerum ()
12    {
13        kara . turnLeft () ;
14        kara . move () ;
15        kara . turnRight () ;
16        kara . move () ;
17        while ( kara . treeRight () ) kara . move () ;
18        kara . turnRight () ;
19        kara . move () ;
20        kara . turnLeft () ;
21    }
22
23
24    public void myProgram ()
25    { // Anfang von myProgram
26        while ( !kara . onLeaf () )
27            {
28                geheZumBaum () ;
29                if ( !kara . onLeaf () ) geheUmBaeumeHerum () ;
30            }
31    } // Ende von myProgram
32
33 } // Ende von Playit2
34

```

3. KARA sucht das Ende eines einfachen Labyrinths bestehend aus Bäumen, wobei keine Löcher in den Baumreihen auftreten. Das Ende des Labyrinths ist eine „Sackgasse“.



```

1 import javakara . JavaKaraProgram ;
2
3 public class Playit3a extends JavaKaraProgram
4 { // Anfang von Playit3a
5
6     void einenSchrittWeiter ()
7     {
8         if (!kara . treeFront ())
9             { kara . move (); }
10        else {
11            if (!kara . treeLeft ())
12                { kara . turnLeft ();
13                  kara . move ();
14                }
15            else { kara . turnRight ();
16                  kara . move ();
17            }
18        }
19    }
20
21
22    public void myProgram ()
23    { // Anfang von myProgram
24        while ( !(kara . treeFront ()
25                  && kara . treeLeft () && kara . treeRight () ) )
26            {
27                einenSchrittWeiter ();
28            }
29    } // Ende von myProgram
30
31 } // Ende von Playit3a

```

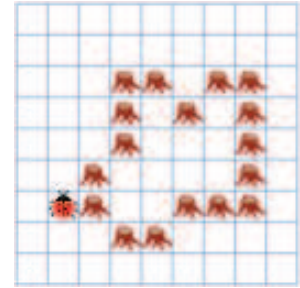
```

1 import javakara .JavaKaraProgram ;
2
3 public class Playit3b extends JavaKaraProgram
4 {
5     /* Kara laeuft durch ein einfaches Labyrinth aus Gaengen.
6     Es gibt keine Verzweigungen.
7     Das Ende ist eine Sackgasse.
8     */
9
10    void einenSchrittWeiter ()
11    { // Vorbedingung: Es liegt keine Sackgasse vor!
12
13        if (!kara .treeLeft ())
14
15            { kara .turnLeft (); }
16
17        else { // links steht Baum, also rechts oder vorne frei
18
19            if (!kara .treeRight ())
20
21                { kara .turnRight (); }
22            }
23
24        kara .move ();
25
26    } // Ende von einenSchrittWeiter
27
28
29    public void myProgram ()
30    {
31        while ( !(kara .treeFront ()
32                && kara .treeLeft () && kara .treeRight () ) )
33
34            { // Solange keine Sackgasse gefunden
35
36                einenSchrittWeiter ();
37
38            } // Ende von while
39
40    } // Ende von myProgram
41
42 } // Ende von Playit3b

```

4. KARA bewacht eine zusammenhängende Fläche, die durch Bäume begrenzt ist. Er soll endlos außen an den „Wänden“ aus Bäumen entlang laufen.

Variante: Er soll zuerst zu seiner Fläche und dann erst endlos darum herum laufen.



```

1 import javakara . JavaKaraProgram ;
2
3 public class Playit4 extends JavaKaraProgram
4 { // Anfang von Playit4
5
6 void zurFlaeche ()
7 {
8     while (!kara . treeFront ()) kara . move ();
9     kara . turnLeft (); // Einmal nach links drehen , dann steht
10                        // auf alle Faelle rechts ein Baum
11 }
12
13 void einenSchrittWeiter ()
14 { // rechtsherum um das Feld
15     if (!kara . treeRight ()) // Erst mal nach rechts schauen !
16     { kara . turnRight ();
17       kara . move (); // und dann nach rechts gehen
18     }
19     else { // rechts steht also ein Baum
20         if (!kara . treeFront ())
21         { kara . move (); } // vorne frei , also vor
22         else { //rechts und vorne stehen Baeume
23             if (!kara . treeLeft ())
24             { kara . turnLeft ();
25               kara . move ();
26             }
27             else { // rechts , vorne , und
28                   // links stehen Baeume : Sackgasse
29                 kara . turnLeft ();
30                 kara . turnLeft ();
31                 kara . move ();
32             }
33         }
34     }
35 }
36
37
38 public void myProgram ()

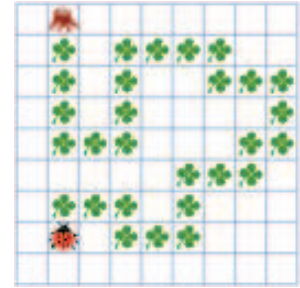
```

```
39 { // Anfang von myProgram
40   zurFlaeche ();
41   while ( true ) einenSchrittWeiter (); // Endlosschleife
42 } // Ende von myProgram
43
44 }
```

5. KARA spielt PACMAN: Er steht auf dem ersten Kleeblatt einer langen Spur von Kleeblättern, die vor einem Baum endet. Er soll alle Kleeblätter auffressen. Schwierige Variante: Die Kleeblattspur verläuft zwischen Bäumen und das Ende der Spur wird durch einen Pilz markiert.

Das Ende des Programms:

```
tools.showMessage("Ich bin so satt, \n"+
                  "ich mag kein Blatt!");
```



```

1 import javakara . JavaKaraProgram ;
2
3 public class Playit5a extends JavaKaraProgram
4 { // Anfang von Playit5a
5
6     void turnAround ()
7     { kara . turnLeft () ;
8       kara . turnLeft () ;
9     }
10
11    void naechstesKleeblattSuchen ()
12    { // Sicher steht vorne kein Baum
13      kara . move () ; // Erst mal vorne nachschauen
14      if ( !kara . onLeaf () )
15          { // kein Blatt vorne , also zurueck und links nachschauen
16            this . turnAround () ;
17            kara . move () ;
18            kara . turnRight () ;
19            kara . move () ;
20            if ( !kara . onLeaf () )
21                { // links ist auch kein Blatt ;
22                  // dann muss es rechts liegen
23                  this . turnAround () ;
24                  kara . move () ;
25                  kara . move () ;
26                }
27          }
28    }
29
30
31    public void myProgram ()
32    { // Anfang von myProgram
33      while ( !kara . treeFront () )
34          {
35            kara . removeLeaf () ;

```

```

36     naechstesKleeblattSuchen ();
37     }
38     tools.showMessage("Ich bin so satt, \n" +
39                     "ich mag kein Blatt!");
40
41 } // Ende von myProgram
42
43 } // Ende von Playit5a

```

```

1 import javakara.JavaKaraProgram;
2
3 public class Playit5b extends JavaKaraProgram
4 { // Anfang von Playit5b
5
6     void turnAround()
7     { kara.turnLeft();
8       kara.turnLeft();
9     }
10
11    void naechstesKleeblattSuchen ()
12    {
13        if (!kara.treeFront())
14            { // Alles offen, erst mal vorne nachschauen
15              kara.move();
16              if (!kara.onLeaf())
17                  { // dann erst ma links nachschauen
18                    this.turnAround();
19                    kara.move();
20                    kara.turnRight();
21                    if (kara.treeFront())
22                        { // links Baum,
23                          // dann kann es nur rechts weitergehen
24                          this.turnAround();
25                          kara.move();
26                        }
27                    else { // links ist kein Baum
28                          kara.move();
29                          if (!kara.onLeaf())
30                              { // jetzt kann es nur
31                                // rechts weitergehen
32                                this.turnAround();
33                                kara.move();
34                                kara.move();
35                              }
36                        }
37                    }
38                }

```



```

39     else { // Vorne steht ein Baum;
40           // es kann nur nach links oder nach rechts gehen
41           if (kara.treeLeft())
42               { // rechts geht's weiter
43                 kara.turnRight();
44                 kara.move();
45               }
46           else { // links steht kein Baum
47                 kara.turnLeft();
48                 kara.move();
49                 if (!kara.onLeaf())
50                     { // rechts geht's weiter
51                       this.turnAround();
52                       kara.move();
53                       kara.move();
54                       return;
55                     }
56               }
57         }
58     }
59
60
61     public void myProgram()
62     { // Anfang von myProgram
63       while ( !kara.mushroomFront() )
64         {
65           kara.removeLeaf();
66           naechstesKleeblattSuchen();
67         }
68       tools.showMessage("Ich bin so satt ,\n" +
69                        "ich mag kein Blatt!");
70     } // Ende von myProgram
71
72 } // Ende von Playit5b

```

```

1  import javakara.JavaKaraProgram;
2
3  public class Playit5c extends JavaKaraProgram
4  { // Anfang von Playit5c
5
6     void turnAround()
7     { kara.turnLeft();
8       kara.turnLeft();
9     }
10
11    void moveBack()
12    { turnAround();

```

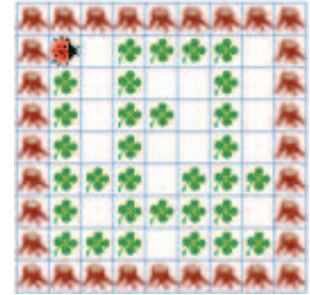
```

13     kara.move();
14     turnAround();
15 }
16
17 boolean vorneGehtsWeiter()
18 {
19     if (kara.treeFront()) {return false;}
20     else { // jetzt muss man nachschauen
21         kara.move();
22         if (kara.onLeaf())
23             { moveBack();
24               return true;
25             }
26         else { moveBack();
27               return false;
28             }
29     }
30 }
31
32 boolean linksGehtsWeiter()
33 {
34     if (kara.treeLeft()) {return false;}
35     else { // jetzt muss man nachschauen
36         kara.turnLeft();
37         kara.move();
38         if (kara.onLeaf())
39             { moveBack();
40               kara.turnRight();
41               return true;
42             }
43         else { moveBack();
44               kara.turnRight();
45               return false;
46             }
47     }
48 }
49
50
51 void naechstesKleeblattSuchen()
52 { if (!vorneGehtsWeiter())
53     { if (linksGehtsWeiter()) { kara.turnLeft(); }
54       else { kara.turnRight(); }
55     }
56     kara.move();
57 }
58
59

```

```
60 public void myProgram()
61 { // Anfang von myProgram
62     while ( !kara.mushroomFront() )
63     {
64         kara.removeLeaf();
65         naechstesKleeblattSuchen();
66     }
67     tools.showMessage("Ich bin so satt ,\n" +
68                     "ich mag kein Blatt!");
69
70 } // Ende von myProgram
71
72 } // Ende von Playit5c
```

6. Eine quadratische Fläche ist von Bäumen umrandet. Innerhalb der Fläche ist ein Muster aus Kleeblättern gelegt, das von KARA invertiert werden soll. KARA startet links oben in der Ecke mit Blick nach rechts.



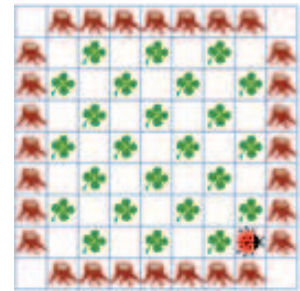
```

1 import javakara . JavaKaraProgram ;
2
3 public class Playit6 extends JavaKaraProgram
4 { // Anfang von Playit6
5
6     boolean nachRechts = true ;
7     boolean fertig = false ;
8
9     void zeileAbarbeiten ()
10    {
11        do
12        {
13            if (kara . onLeaf ()) { kara . removeLeaf (); }
14            else {kara . putLeaf (); }
15            kara . move ();
16        } while (!kara . treeFront ());
17        if (kara . onLeaf ()) { kara . removeLeaf (); }
18        else {kara . putLeaf (); }
19    }
20
21    public void myProgram ()
22    { // Anfang von myProgram
23        while ( !fertig )
24        {
25            zeileAbarbeiten ();
26            if ( (nachRechts && kara . treeRight ()) ||
27                (!nachRechts && kara . treeLeft ())) { fertig = true ;}
28            if ( !fertig )
29                { if (nachRechts)
30                    {
31                        kara . turnRight ();
32                        kara . move ();
33                        kara . turnRight ();
34                    }
35                    else {
36                        kara . turnLeft ();
37                        kara . move ();
38                        kara . turnLeft ();

```

```
39         }
40     }
41     nachRechts = !nachRechts;
42 }
43 } // Ende von myProgram
44 } // Ende von Playit6
```

7. Eine quadratische Fläche ist von Bäumen umrandet. Innerhalb der Fläche soll ein schachbrettartiges Muster aus Kleeblättern von KARA gelegt werden. KARA startet links oben in der Ecke mit Blick nach rechts.



```

1 import javakara . JavaKaraProgram ;
2
3 public class Playit7 extends JavaKaraProgram
4 { // Anfang von Playit7
5
6     boolean nachRechts = true ;
7     boolean fertig = false ;
8     boolean abgelegt = false ;
9
10    void legeBlattWennNoetigUndVor ()
11    {
12        if (!abgelegt) { kara . putLeaf () ; }
13        abgelegt = !abgelegt ;
14        kara . move () ;
15    }
16
17    void zeileAbarbeiten ()
18    {
19        while (!kara . treeFront ()) legeBlattWennNoetigUndVor () ;
20    }
21
22    public void myProgram ()
23    { // Anfang von myProgram
24        while ( !fertig )
25            {
26                zeileAbarbeiten () ;
27                if ( (nachRechts && kara . treeRight ()) ||
28                    (!nachRechts && kara . treeLeft ())) { fertig = true ; }
29                if ( !fertig )
30                    { if (nachRechts)
31                      {
32                        kara . turnRight () ;
33                        legeBlattWennNoetigUndVor () ;
34                        kara . turnRight () ;
35                      }
36                    else {
37                        kara . turnLeft () ;
38                        legeBlattWennNoetigUndVor () ;

```

```

39         kara.turnLeft ();
40     }
41 }
42     nachRechts = !nachRechts;
43 }
44 } // Ende von myProgram
45
46 } // Ende von Playit7
47

```

Hilfe: Bei den letzten beiden Aufgaben ist es hilfreich, mit booleschen Variablen zu arbeiten.

```

boolean fertig = false; // Deklaration und Initialisierung

fertig = !fertig;      // Aus true wird false und umgekehrt

if (...) fertig = true; // Direkte Zuweisung

if (fertig) {...}     // Boolesche Variable als Bedingung

```