You will need some additional Java constructs for the exercises:

Variables added to Kara's state (fields or instance variables) to memorize things, like numbers (int) or true/false (boolean) (define outside the act() method!)

```
 int zaehler = 0;
```

(Declaration and initialization)

These can then be modified from within the act() (or any other) method:

```
zaehler = zaehler + 1;
movingRight = !movingRight; //toggle
```

(to increment the counter by one or to toggle true/false - !movingRight is "not the old value")

If you define variables within the act method, they will be lost and newly initialized with each call to act().

Logical Operators

And: A && B – only true if both A and B are true.

Or: A || B – true if A, B, or both are true.

```
if(treeRight() || treeLeft()) ….
```

Printing to command line: e.g.

```
System.out.println(zaehler);
```

to print out the value of the variable zaehler.

You can also build new sensors by defining a new method returning a boolean value. You can use `return;` in a void method to jump back out of the method. (we've already seen a new method, walkAroundTree()):

```
public boolean isInTunnel(){
        return (treeRight() && treeLeft());
    }
```

Use the Greenfoot run() loop which calls act() wherever possible. In some cases, you will need a second loop within the act method or a method you define. For example, to walk to the next tree ahead:

```
while (!treeFront()){
   move();
}
```

Use Greenfoot.stop(); to stop the Greenfoot-loop which calls act repeatedly:

```
public void act(){
   move();
   if (treeFront()) Greenfoot.stop();
}
```

Method Parameters and Loops: Define and use methods that take parameters – this example defines and uses a method that calls move() n times:

```
    public void act()
    {
       move(5);
    }
    public void move(int n){
        int i = 0;
        while (i<n){
            move();
            i++;
        }
    }
```

Loops patterns: this is also an example how to do something n-times (a variable, but determined number of times) another useful loop pattern is to repeat something until a condition is met:

```
        while (!treeFront()){
            move();
        }
```

These can be combined to count something, e.g. the number of fields, the distance to the next tree ahead:

```
    public int distanceToTree(){
        int i = 0;
        while (!treeFront()){
            move();
            i++;
        }
        return i;
    }
```

Flags and toggles: Boolean variables can be used as flags or toggles, that is, mark that something happened with a flag or toggles that are switched back and forth to store a state-like information. Two examples:
A Flag can be used to keep track if an event has already happened, e.g.;

```
boolean enteredTunnel = false;
act(){
  ...
  enteredTunnel = true;
  ....
}
```

To keep track of the current direction Kara moves to, you can use a toggle:

```
boolean movingRight = true;
act(){
  ...
  turn();
  movingRight = ! movingRight;
  ....
}
```

AUFGABE 1: KARA ZÄHLT KLEEBLÄTTER

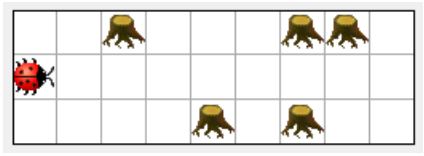
02-01-counting-leaves

Kara soll waagrecht von links nach rechts gehen bis zum Baum und dabei die Kleeblätter zählen.

Verwenden Sie zum Zählen ein Feld (Instanzvariable), die Sie **außerhalb** der act()-Methode definieren:
int zaehler = 0;

Geben Sie am Ende das Ergebnis mit System.out.println(zaehler); aus.

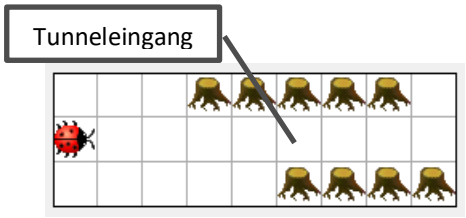
AUFGABE 2: BLATT BEIM BAUM



02-02-leaf-at-tree

Nun soll Kara wieder geradeaus gehen und überall dort ein Blatt legen, wo entweder links oder rechts oder auf beiden Seiten ein Baum steht.

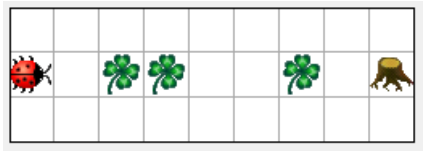
AUFGABE 3: ANGST VOR TUNNEL



Kara hat etwas Angst vor Tunneln. Sie soll auf jedem Feld überprüfen, ob es ein Tunneleingang ist (d.h. ob es auf beiden Seiten Bäume gibt). Ist dies der Fall, so lässt er vor Schreck gleich ein Kleeblatt fallen.

Laden Sie das Szenario ***02-03-afraid-of-tunnel,*** schreiben Sie das Programm und testen Sie es mit allen drei Wiesen.

Aufgabe 4: Blätter legen bis zum Baum



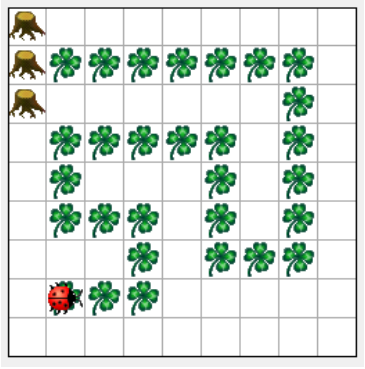
02-04-put-leaf-track

Kara soll vorwärts laufen und dabei überall ein Blatt legen, wo keines ist. Wenn er beim Baum angelangt ist, soll sie nichts mehr machen (auch wenn der Act- oder der Run-Knopf nochmals gedrückt wird).

Aufgabe 5 (schwierig): Kara spielt Pacman

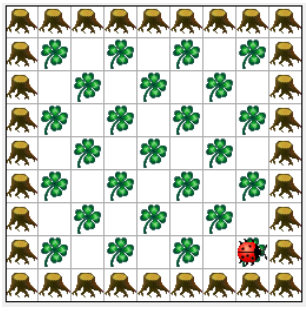
2-05-kara-plays-pacman/

Kara spielt Pacman: Er steht auf dem ersten Kleeblatt einer langen Spur von Kleeblättern, die vor einem Baum endet. Er soll alle Kleeblätter auffressen und vor den Bäumen stoppen.



Schreiben Sie zur besseren Übersicht für geeignete Programmteile eigene Methoden. (Kapitel 2/16)

Aufgabe6: Kara in a box - Blätter Zählen

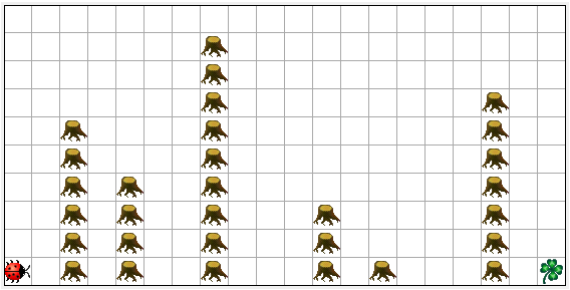


02-06-kara-in-a-box

Eine quadratische Fläche ist von Bäumen umrandet. Innerhalb der Fläche sollen die Blätter gezählt werden. Kara startet links oben in der Ecke mit Blick nach rechts.

ZUSATZAUFGABE 7 (FÜR SCHNELLE): DIE LÄNGSTE BAUMREIHE

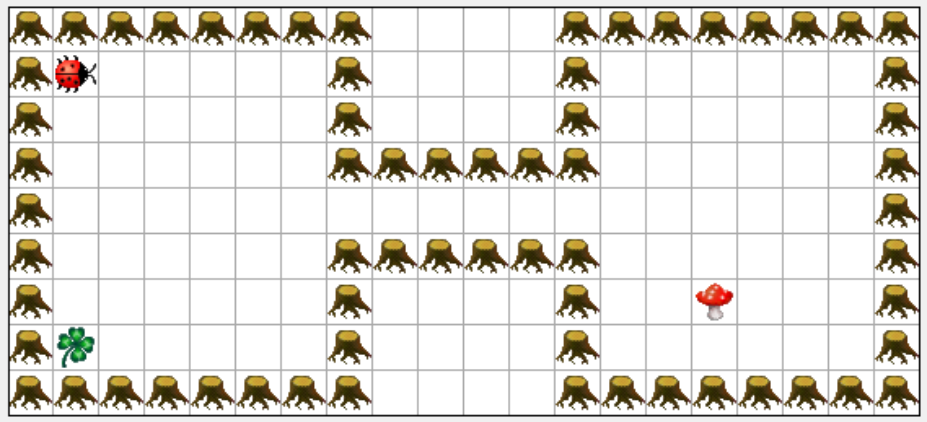
02-07-the-longest-tree-line



Auf der Wiese gibt es verschiedene Baumreihen. Kara soll nun die Länge (in Anzahl Bäumen) der längsten Baumreihe ermitteln und auf die Konsole ausgeben. Zwischen den Baumreihen ist immer mindestens ein Feld Platz. Auf dem letzten Feld liegt ein Kleeblatt.

ZUSATZAUFGABE 8 (FÜR SEHR SCHNELLE): PILZ DURCH TUNNEL SCHIEBEN

02-08-push-mushroom-through-tunnel



Die Welt von Kara hat zwei Boxen, welche durch einen Tunnel verbunden sind. In der Box links befindet sich Kara und ein Kleeblatt. In der Box rechts befindet sich ein Pilz. Kara soll nun auf die andere Seite gelangen, den Pilz finden und ihn auf die andere Seite schieben. Auf der anderen Seite angekommen soll der Pilz schliesslich auf das Kleeblatt geschoben werden.

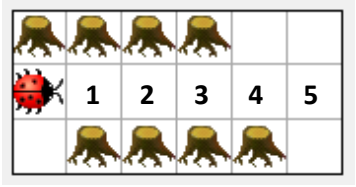
Sie können davon ausgehen, dass Kara immer oben links in der Ecke startet und das Kleeblatt sich unten links befindet. Der Pilz kann jedoch an einer beliebigen Stelle rechts vom Tunnel stehen.

Hinweis: Diese Aufgabe kann auch in Zusammenarbeit zu zweit gelöst werden. Dabei können die Teilprobleme untereinander aufgeteilt werden:

- Tunneleingang finden
- Pilz finden
- Pilz vor den Tunneleingang stossen
- Pilz auf das Kleeblatt stossen

AUFGABE 9:

Gegeben ist die folgende Situation: Kara steht vor einem Tunnel.

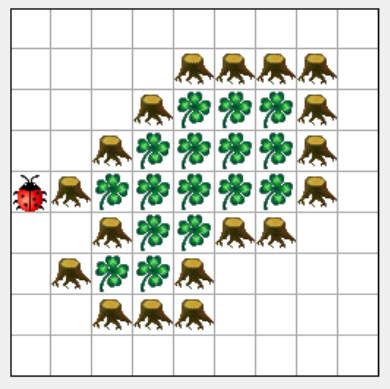


Beschreiben Sie, was die folgenden Schleifen bewirken und wie viele Schritte Kara macht. Verwenden Sie dazu die Technik des „Manual Walkthroughs" – d.h. gehen Sie das Programme manuell Schritt für Schritt durch!

Code	Beschreibung	Anzahl Schritte
`while (treeLeft()) { move(); }`		
`while (treeRight()) { move(); }`		
`while (treeLeft() || treeRight()) { move(); }`		
`if (treeLeft()) { move(); } while (treeLeft() && treeRight()) { move(); }`		
`while (!treeFront) { if (treeLeft()) { move(); } }`		

ZUSATZAUFGABE 10 (SCHWIERIG): KARA ALS WÄCHTER

Kara will einen Wald bewachen. Sie soll endlos aussen am Waldrand entlang laufen. Als Hilfe können Sie Sich ein Flussdiagramm zeichnen.



02-10-kara-as-guard