

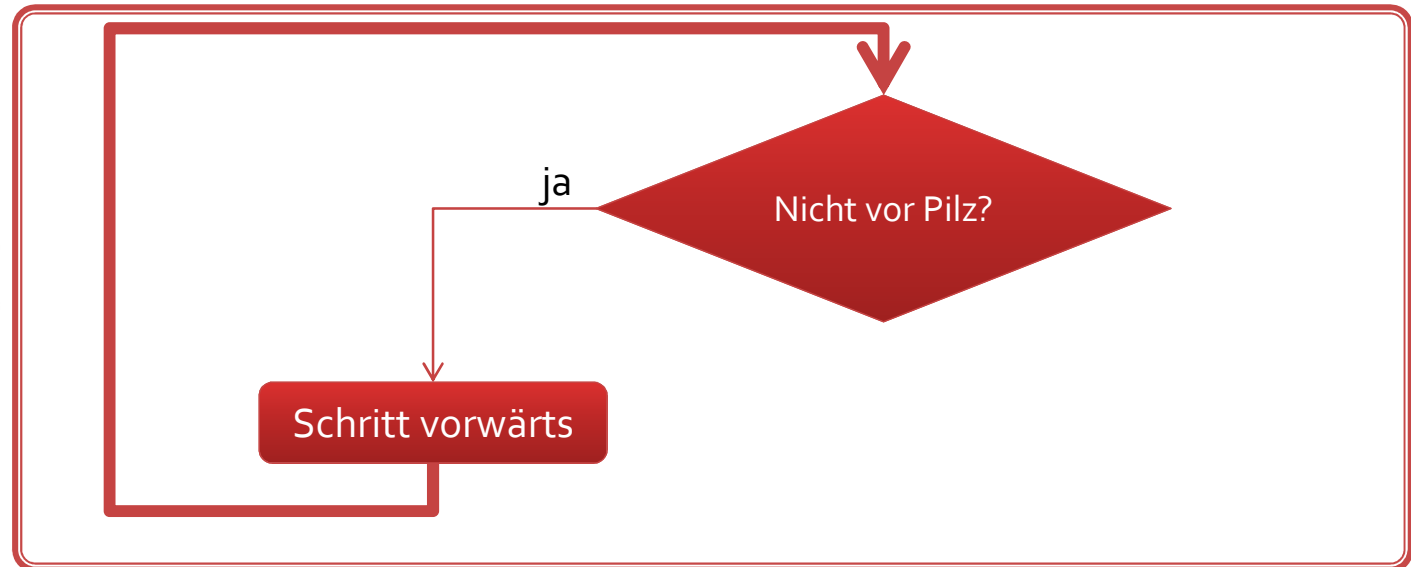
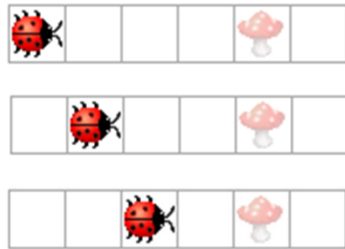
Wieder und wieder und wieder ...

JavaKara programmieren: Wiederholungen

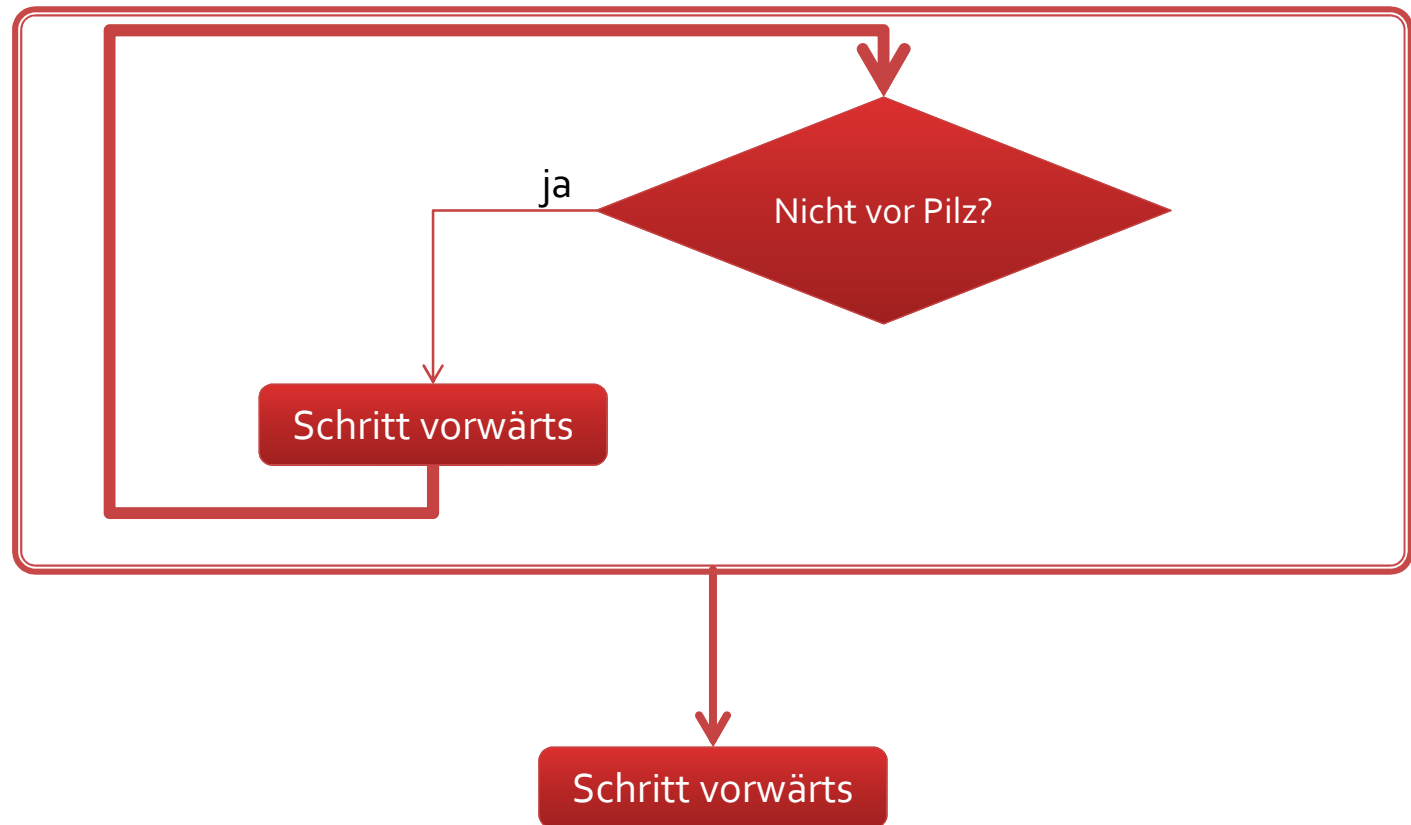
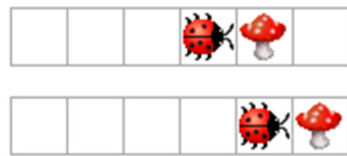
**Kara soll laufen, bis er vor Pilz ist,
und den Pilz dann ein Feld schieben**



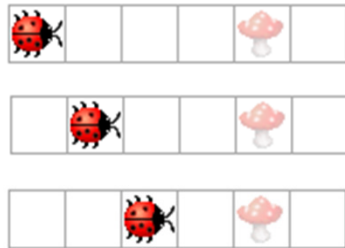
Kara soll laufen, bis er vor Pilz ist, und den Pilz dann ein Feld schieben



Kara soll laufen, bis er vor Pilz ist, und den Pilz dann ein Feld schieben



Pilzsuche in Java-Code

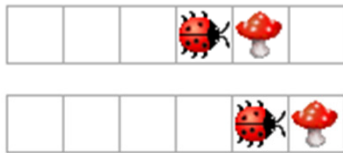


```
public void myProgram() {  
    while (! kara.mushroomFront()) {  
        kara.move();  
    }  
    kara.move();  
}
```

Die allgemeine Beschreibung einer Schleife mit Abbruchsbedingung:

```
while (Boole'scher Ausdruck) {  
    // solange der Ausdruck zu wahr ausgewertet wird,  
    // werden alle Anweisungen in diesem Block wiederholt  
}
```

Pilzsuche in Java-Code



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

Die allgemeine Beschreibung einer Schleife mit Abbruchsbedingung:

```
while (Boole'scher Ausdruck) {  
    // solange der Ausdruck zu wahr ausgewertet wird,  
    // werden alle Anweisungen in diesem Block wiederholt  
}
```

Kara soll laufen, bis er auf einem Kleeblatt zwischen Bäumen ist

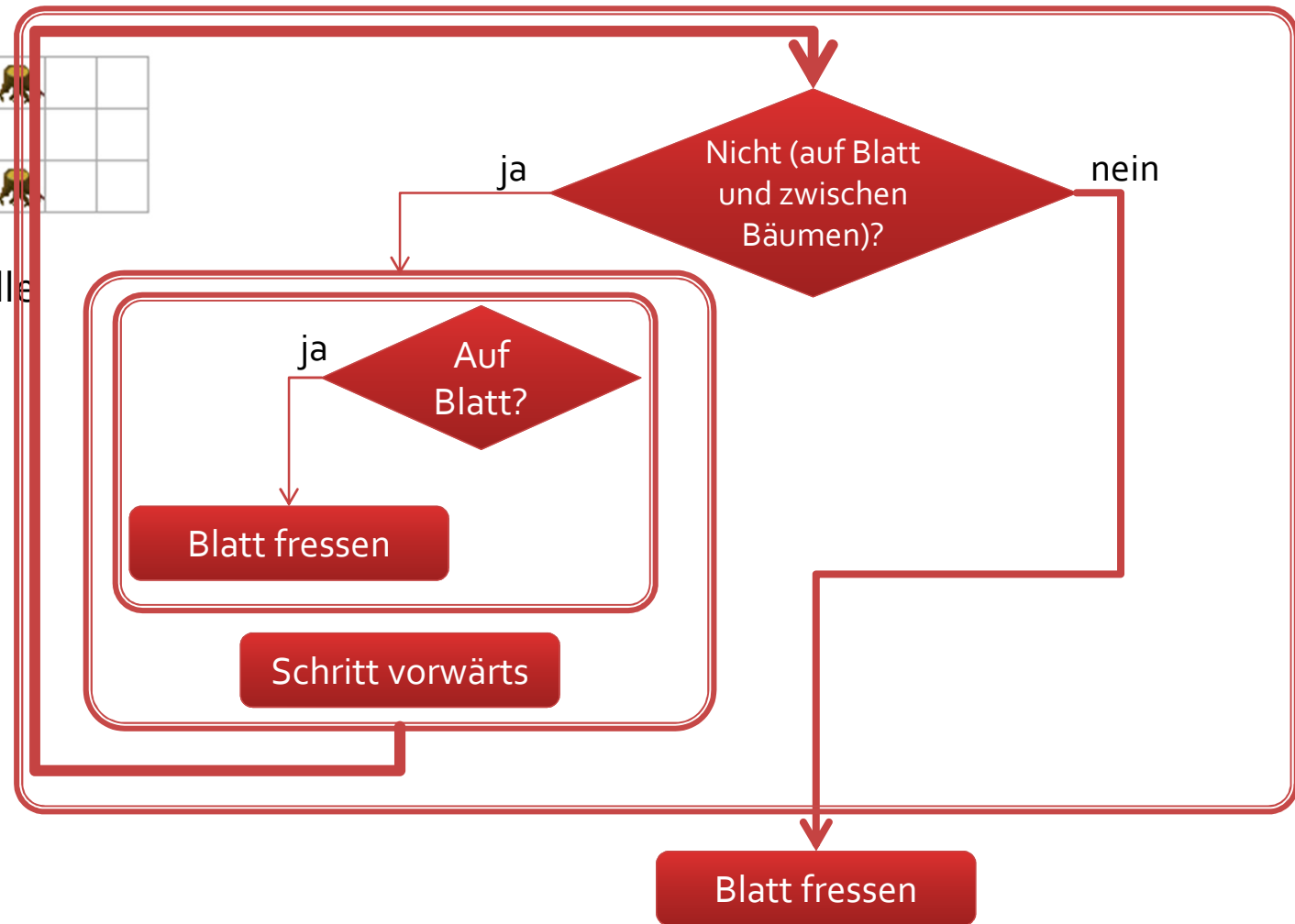


Und unterwegs alle
Blätter essen.

Kara soll durch diesen «Wald» laufen, bis er beim Kleeblatt ist



Und unterwegs alle
Blätter essen.



Kara soll durch diesen «Wald» laufen, bis er beim Kleeblatt ist



Und unterwegs alle
Blätter essen.

```
public void myProgram() {  
  
    while (! (kara.treeLeft() &&  
             kara.treeRight() &&  
             kara.onLeaf())) {  
  
        if (kara.onLeaf()) {  
            kara.removeLeaf();  
        }  
  
        kara.move();  
    }  
  
    kara.removeLeaf();  
}
```

Kara soll fünf Kleeblätter legen: Schleife mit Abbruchbedingung



```
public void myProgram() {
```

```
    int i = 0;
```

```
    while (i < 5) {
```

```
        kara.putLeaf();
```

```
        kara.move();
```

```
        i++;
```

```
    }
```

```
}
```

Kara soll fünf Kleeblätter legen: Schleife mit Abbruchbedingung



```
public void myProgram() {
```

```
int i = 0; Zähler initialisieren
```

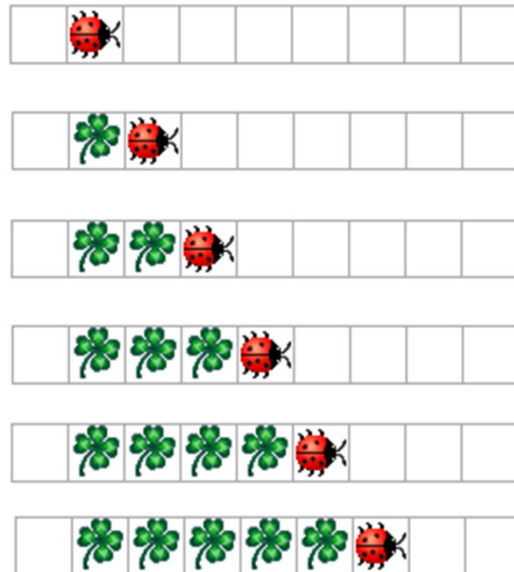
```
while (i < 5) { Abbruchbedingung
```

```
kara.putLeaf();  
kara.move();
```

```
i++; Zähler erhöhen
```

```
}  
}
```

Kara soll fünf Kleeblätter legen: Kurzschreibweise mit Zählschleife



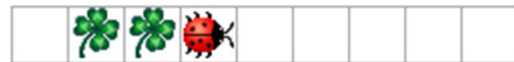
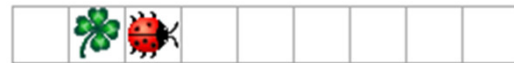
```
public void myProgram() {  
  for (int i = 0; i < 5; i++) {  
    kara.putLeaf();  
    kara.move();  
  }  
}
```

Zähler initialisieren: Beliebige Anweisungen

Abbruchbedingung: Boole'scher Ausdruck

Zähler erhöhen: Beliebige Anweisungen

Kara soll fünf Kleeblätter legen: Kurzschreibweise mit Zählschleife



Alles gleichwertige For-Schleifen:

```
for (int i = 0; i < 5; i++)
```

```
for (int i = 1; i <= 5; i++)
```

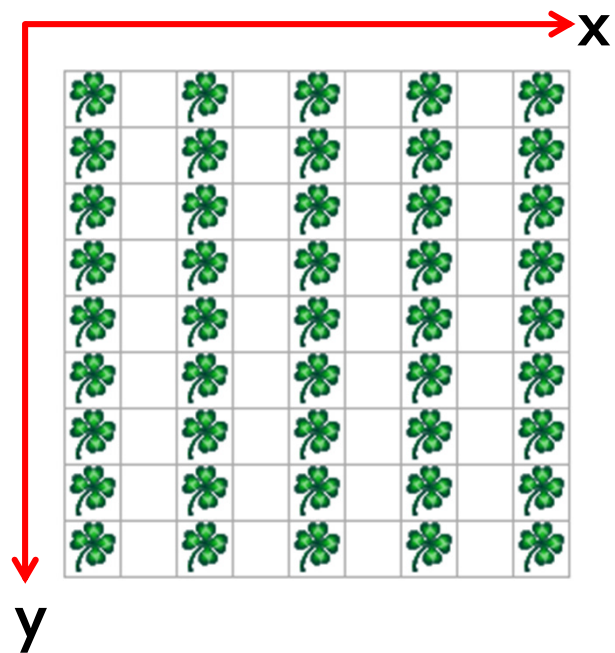
```
for (int i = 10; i < 20; i = i+2)
```

```
for (int i = 32; i > 1; i = i / 2)
```

...

Der Phantasie sind keine Grenzen gesetzt. Aber ein Programm soll seine Absicht immer so klar wie möglich kommunizieren. Üblicherweise wird daher die erste Schreibweise verwendet.

Kara soll Kleeblattstreifen legen

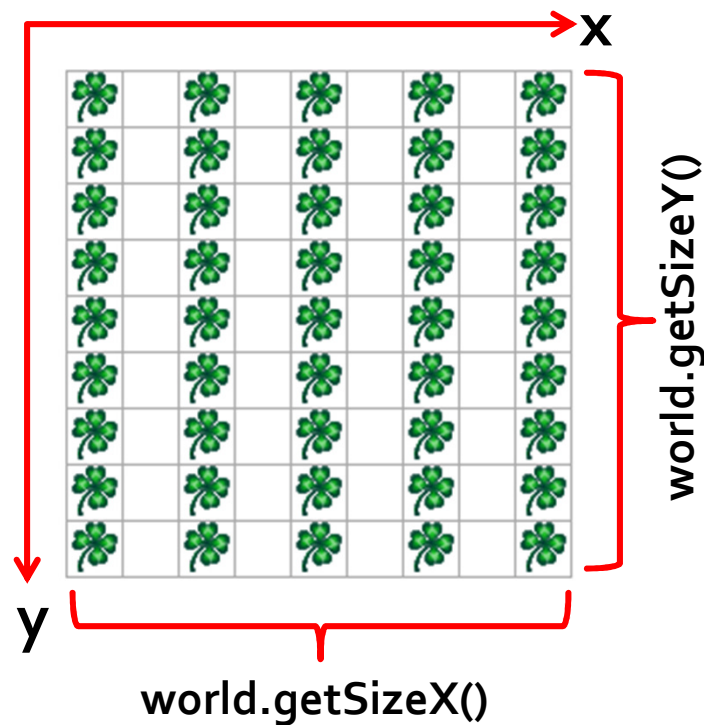


Objekt «world» bietet Methoden für direkte Manipulation der Welt (siehe auch Hilfe in JavaKara):

```
world.setLeaf(x, y, true);  
// platziert Kleeblatt an (x,y),  
// falls noch keines vorhanden
```

```
world.setLeaf(x, y, false);  
// löscht Kleeblatt an (x,y),  
// falls eines vorhanden
```

Kara soll Kleeblattstreifen legen

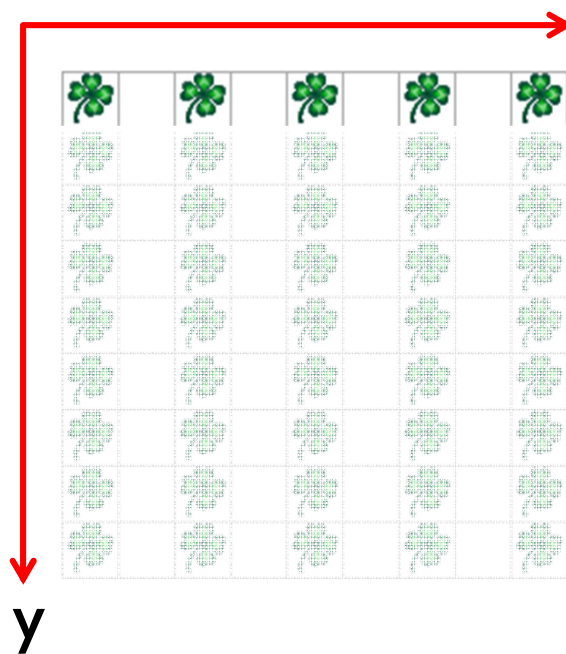


Objekt «world» bietet Methoden für direkte Manipulation der Welt (siehe auch Hilfe in JavaKara):

```
int breite = world.getSizeX();  
// Breite der Welt abfragen
```

```
int hoehe = world.getSizeY();  
// Höhe der Welt abfragen
```

Kara soll Kleeblattstreifen legen

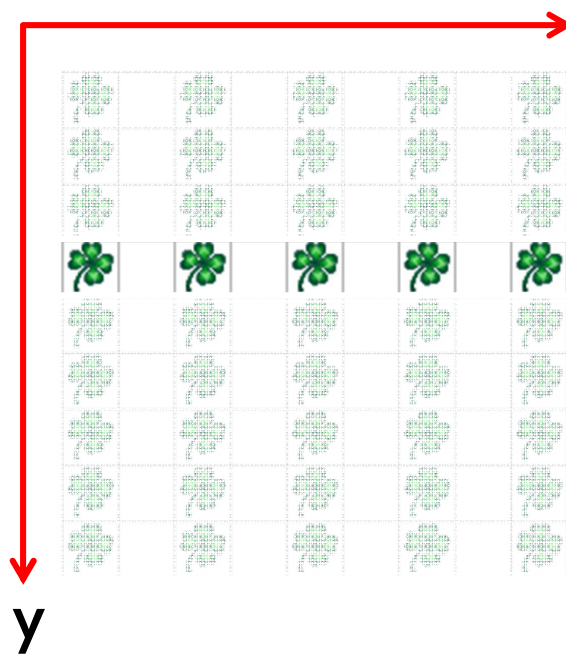


Betrachten wir zuerst nur die einzelne Zeile **0**:

```
for (int x = 0; x < world.getSizeX(); x = x + 2) {  
    world.setLeaf(x, 0, true);  
}
```

x nimmt die Werte 0, 2, ..., world.getSizeX()-1 an.
Im konkreten Beispiel links: 0, 2, 4, 6, 8.

Kara soll Kleeblattstreifen legen

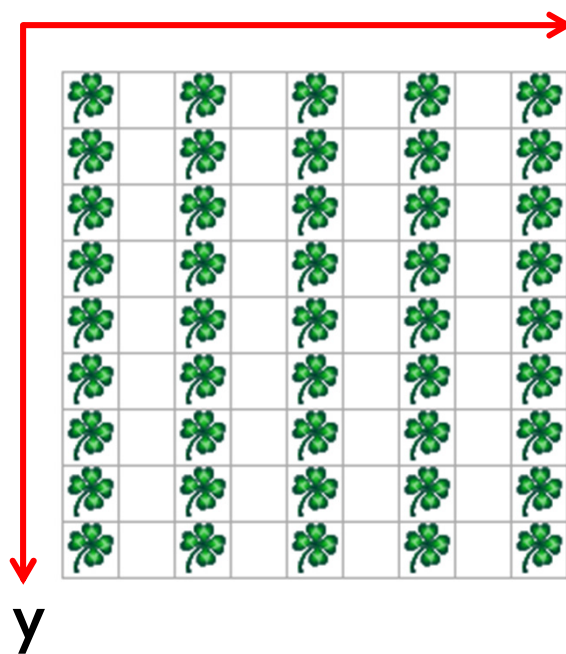


→ **x** Betrachten wir zuerst nur die einzelne Zeile **y=3**:

```
int y = 3;  
for (int x = 0; x < world.getSizeX(); x = x + 2) {  
    world.setLeaf(x, y, true);  
}
```

x nimmt die Werte 0, 2, ..., world.getSizeX()-1 an.
Im konkreten Beispiel links: 0, 2, 4, 6, 8.

Kara soll Kleeblattstreifen legen



Betrachten wir nun alle Zeilen y :

```
for (int y = 0; y < world.getSizeY(); y++) {  
    for (int x = 0; x < world.getSizeX(); x = x + 2) {  
        world.setLeaf(x, y, true);  
    }  
}
```

y nimmt die Werte $0, 1, 2, \dots, \text{world.getSizeY()}-1$ an.
Im konkreten Beispiel links: $0, 1, 2, 3, 4, 5, 6, 7, 8$.

Für jeden Wert von y :

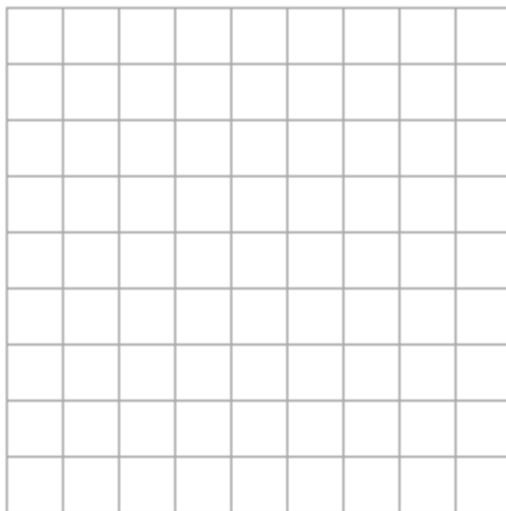
x nimmt die Werte $0, 2, \dots, \text{world.getSizeX()}-1$ an.
Im konkreten Beispiel links: $0, 2, 4, 6, 8$.

Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```

Methode myProgram

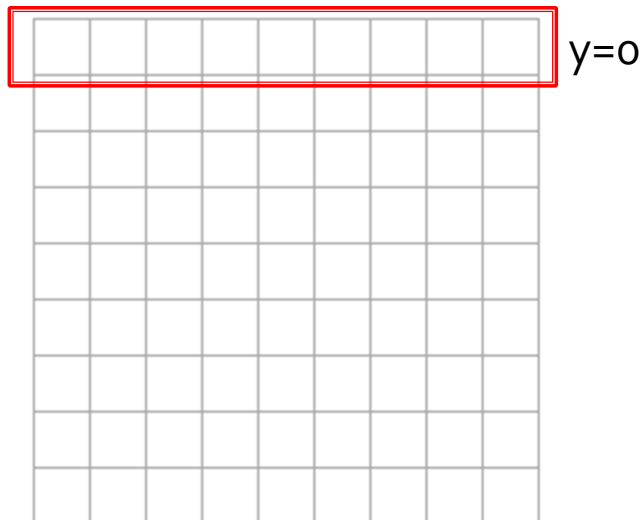


Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```

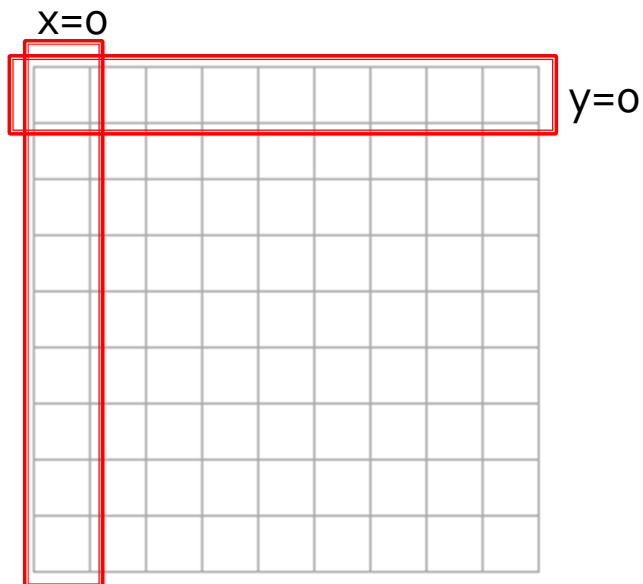
int y=0
Methode myProgram



Kara soll Kleblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```

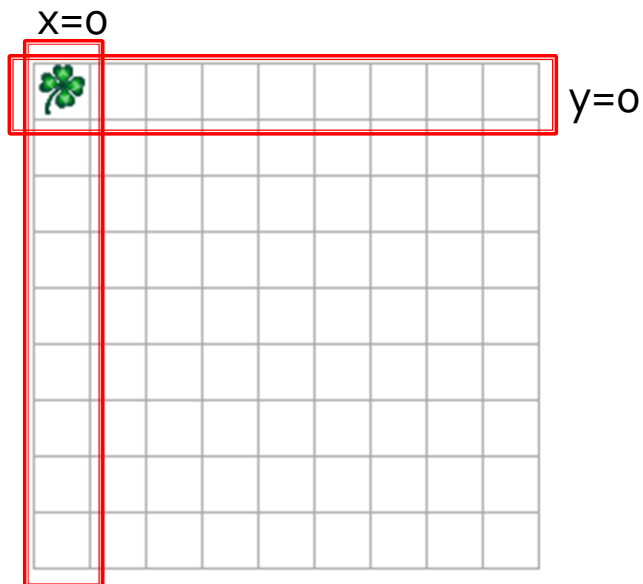


```
int x=0  
int y=0  
Methode myProgram
```

Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```



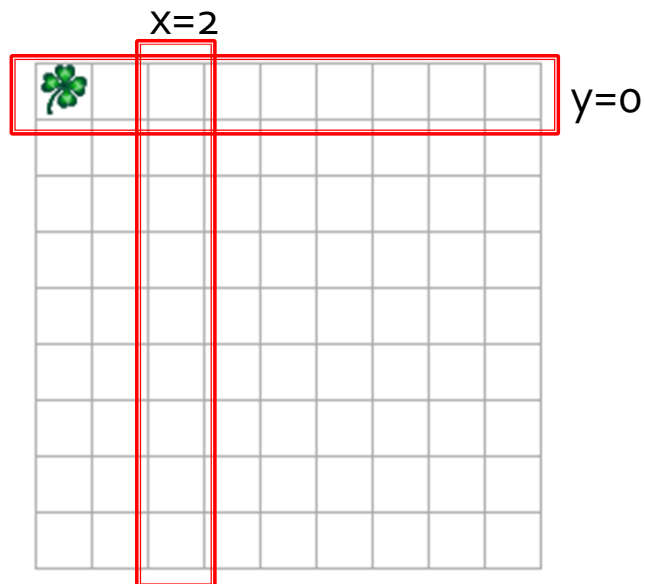
```
int x=0  
int y=0  
Methode myProgram
```

Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```

```
int x=2  
int y=0  
Methode myProgram
```

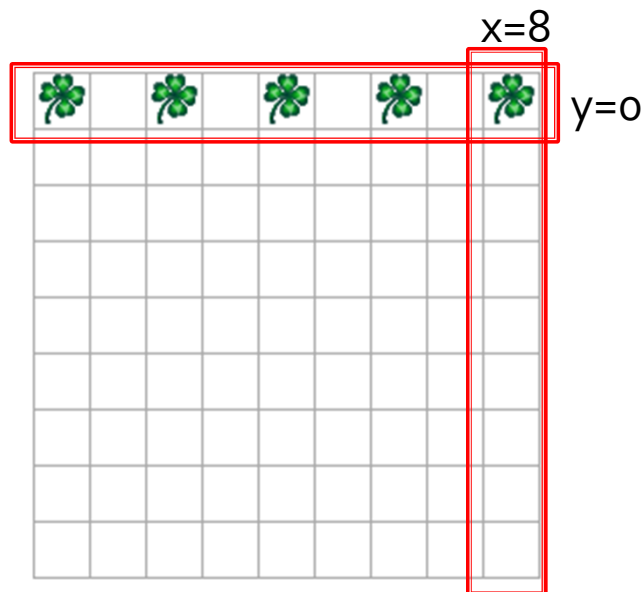


Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```

int x=8
int y=0
Methode myProgram

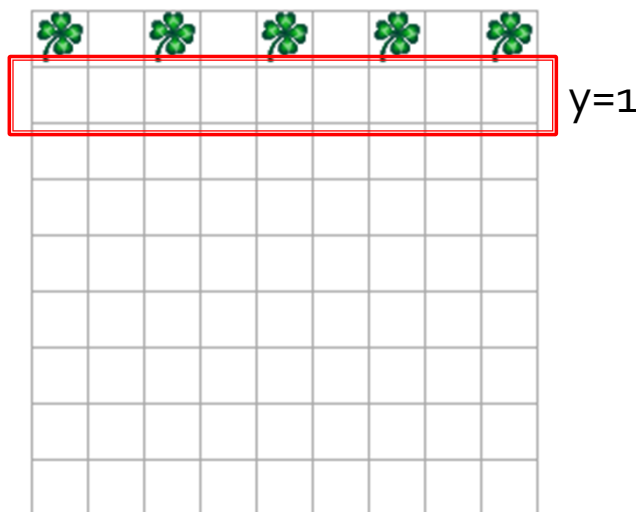


Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```

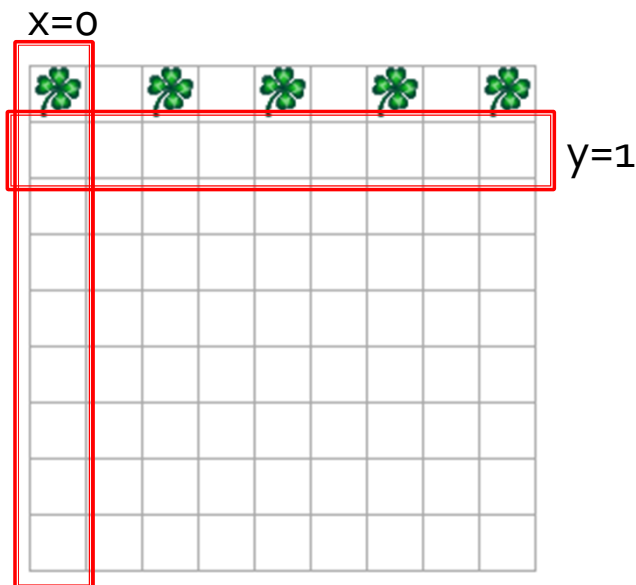
int y=1
Methode myProgram



Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```

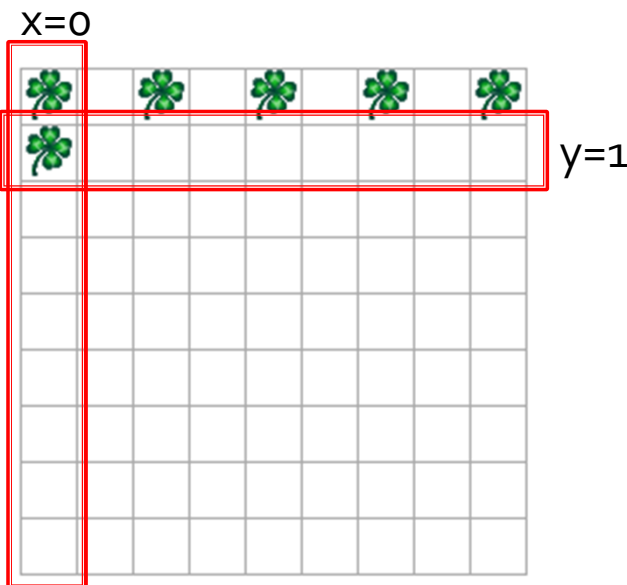


```
int x=0  
int y=1  
Methode myProgram
```

Kara soll Kleeblattstreifen legen: Programmausführung



```
public void myMainProgram() {  
    for (int y = 0; y < world.getSizeY(); y++) {  
        for (int x = 0; x < world.getSizeX(); x = x + 2)  
            world.setLeaf(x, y, true);  
        }  
    }  
}
```



```
int x=0  
int y=1  
Methode myProgram
```

Kara soll Kleeblattstreifen legen: Alternative Lösung: Methode

```
@Override
public void myMainProgram() {
    for (int y = 0; y < world.getSizeY(); y++) {
        eineZeile(y);
    }
}

void eineZeile(int y) {
    for (int x = 0; x < world.getSizeX(); x = x + 2) {
        world.setLeaf(x, y, true);
    }
}
```

Kara soll Kleeblattstreifen legen: Alternative Lösung: Spalten zuerst

```
@Override
public void myMainProgram() {
    for (int x = 0; x < world.getSizeX(); x = x + 2) {
        for (int y = 0; y < world.getSizeY(); y++) {
            world.setLeaf(x, y, true);
        }
    }
}
```

Beliebig kombinierbar: Schleifen, Verzweigungen

```
if (...) {  
  while (...) {  
    if (...) {  
      for (...) {  
      }  
    }  
  }  
  else if {  
    while (...) {  
    }  
  }  
}
```



HILFE!

Programme lesbar halten mit überschaubaren Methoden

```
if (...) {  
  while (...) {  
    methode1();  
  }  
}
```

```
void methode1() {  
  if (...) {  
    methode2();  
  }  
  else if {  
    methode3();  
  }  
}
```

```
void methode2() {  
  for (...) {  
  }  
}
```

```
void methode3() {  
  while (...) {  
  }  
}
```