

Grundlegende Programmierkonzepte: Variablen, Methoden-Parameter, Rückgabewerte

Erklärungen zu Methoden-Parametern

Kara soll vor sich eine Spur von x Kleeblättern legen. Die Anzahl x sei variabel.

Karas Auftrag soll durch eine **Methode legeKleeblattzeile** erledigt werden. Man muss der Methode dazu die gewünschte Anzahl als **Parameter** übergeben.

```
void legeKleeblattzeile(int anzahl) {  
    for (int i=1; i<=anzahl; i++) {  
        kara.putLeaf();  
        kara.move();  
    }  
}
```

```
public void myProgram() {  
    legeKleeblattzeile(7);  
    kara.turnRight();  
    legeKleeblattzeile(5);  
}
```

Erläuterungen

1. Bei den bisherigen Programmen war es mitunter lästig, die Methoden immer mit einem **Klammerpaar ()** zu versehen. Nun wird klar, dass die bisher eingesetzten Methoden nur Spezialfälle darstellen, bei denen **kein Parameter** übergeben wird.
2. In den runden Klammern nach dem Methodennamen wird der Parameter (hier anzahl) mit einem Typ angegeben, hier int. **Mehrere Parameter werden durch Kommata getrennt**. Beispiel: void zeichneRechteck(int breite, int hoehe)
3. Beim Aufruf einer Methode wird der beim Aufruf in der Klammer stehende Wert in die Variable (hier anzahl) kopiert.

Erklärungen zu Methoden-Rückgabewerten

Kara steht am Anfang einer langen Spur (mit Unterbrechungen) von Kleeblättern. Er soll herausfinden, wie viele Blätter in der Spur auf den ersten *laenge* Plätzen liegen.

Karas Auftrag soll durch eine **Methode zaehle()**, die das **Ergebnis des Zahlvorgangs zurückgeben** soll, erledigt werden.

```
int zaehle(int laenge) {
    int anzahl=0;
    for (int i=1; i<=laenge; i++) {
        if (kara.onLeaf()) {
            anzahl++;
        }
        kara.move();
    }
    return anzahl; // Rückgabe
}

public void myProgram() {
    int wieWeit=0;
    int kleeblattAnzahl=0;
    wieWeit = tools.intInput("Wie weit soll ich suchen?");
    if (wieWeit>0) {
        kleeblattAnzahl = zaehle(wieWeit);
        tools.showMessageDialog("Ich habe "+kleeblattAnzahl+
            " Blaetter gefunden.");
    }
    else {
        tools.showMessageDialog("Ich habe nichts zu tun.");
    }
}
```

Erläuterungen

1. Bei den bisherigen Programmen war es mitunter lästig, den Methoden immer das Wort **void**, was so viel wie leer bedeutet, voran zu stellen. Nun wird klar, dass die bisher eingesetzten Methoden nur **Spezialfall** darstellen, bei denen **kein Wert zurückgegeben** wird.
2. Bei der Methode *zaehle* wird deutlich, dass Methoden in Java als Funktionen (wie in der Mathematik) angesehen werden können. Der Parameter *laenge* ist das Argument der Funktion (in der Mathematik meistens *x*), die einen Funktionswert (hier *zaehle*) eines bestimmten Typs (hier *int*) zurückliefert, den man einer Variablen zuweisen kann (hier *kleeblattAnzahl = zaehle(wieWeit)*). In der Mathematik wurde man $y = f(x)$ schreiben.

3. Dem Namen der Methode wird der **Typ des Rückgabewertes voran** gestellt. Die **Rückgabe** eines Funktionswertes innerhalb einer Methode **geschieht mit return**, gefolgt vom Funktionswert. Man muss darauf achten, dass eine Methode mit einem Rückgabewert immer mit einem entsprechenden return verlassen wird.

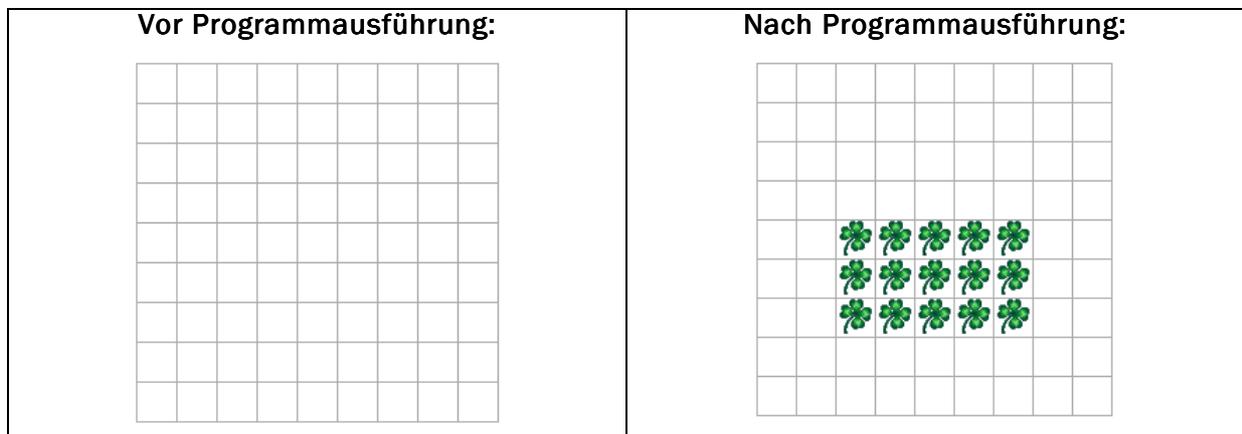
Einfache Methoden ohne einen Rückgabewert können auch mit einem return verlassen werden. Dann steht nur ein Stichpunkt hinter return: „return ;“

4. In der Programmzeile `wieWeit = tools.intInput("Wie weit soll ich suchen?");` wird der Text „Wie weit soll ich suchen?“ als Parameter vom Typ String an die Methode `tools.intInput` übergeben. Diese Methode öffnet ein neues Fenster mit dem Text und der Möglichkeit, eine Ganzzahl (int) einzugeben. Der Funktionswert ist dann die eingegebene Zahl, die der Variablen `wieWeit` zugewiesen wird. Der Methodenkopf dazu: `int intInput(String title)`. Weitere Informationen dazu in der Dokumentation innerhalb von Javakara (oben rechts auf das Fragezeichen klicken).

Aufgabe: Rechteck an beliebigen Koordinaten zeichnen

Schreiben Sie ein Programm, das ein ausgefülltes Rechteck zeichnet, und zwar an vom Benutzer eingegebenen Koordinaten und in vom Benutzer eingegebener Grösse.

Angenommen, Sie geben ein `x=2`, `y=4`, `Breite=5` und `Höhe=3`, dann soll Ihr Programm das folgende Rechteck zeichnen:

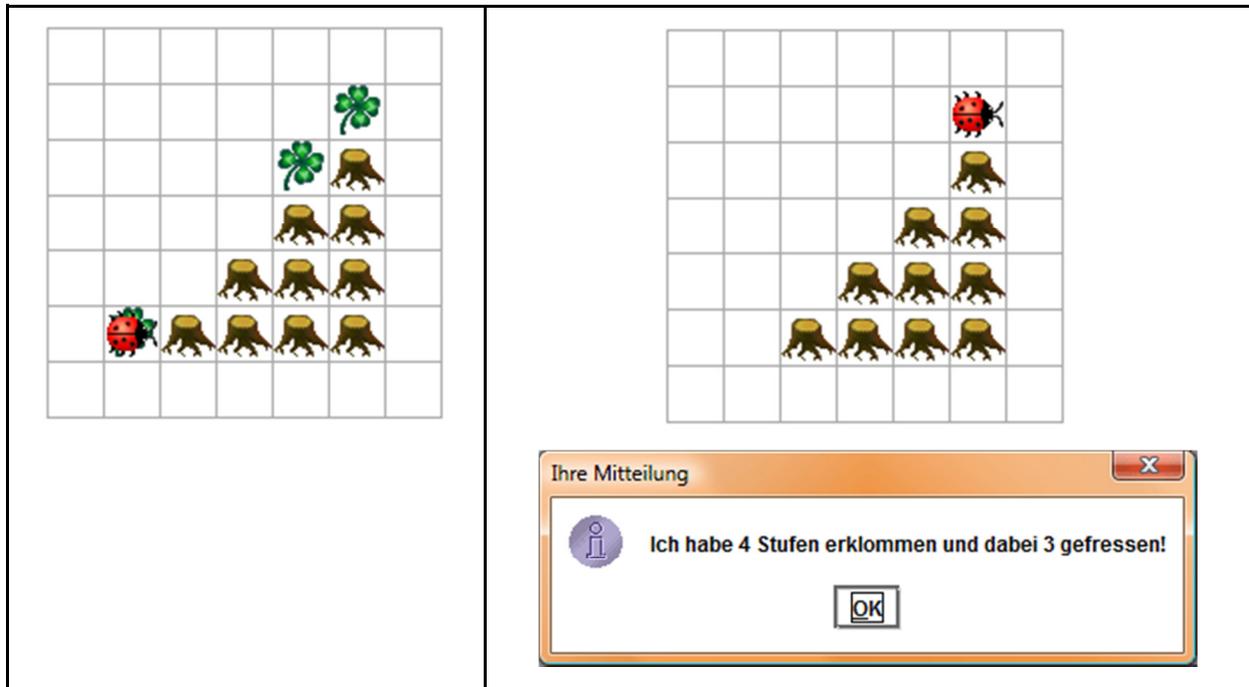


Zur Erinnerung: Eine Ganzzahl können Sie vom Benutzer wie folgt erfragen:

```
int xLinksOben = tools.intInput("x");
```

Aufgabe: Treppensteigen und Kleeblätter zählen

Kara soll eine Treppe aus Bäumen besteigen und dabei alle Kleeblätter fressen. Wenn er das Ende der Treppe erreicht hat, soll er die Anzahl Treppenstufen und die Anzahl Kleeblätter ausgeben:



Für das Zählen der Kleeblätter und Stufen ist der Anfang gemacht. Ergänzen Sie das Programm unten so, dass Kara die Treppe hochläuft und dabei die **Stufen zählt** und die **Kleeblätter frisst und zählt**.

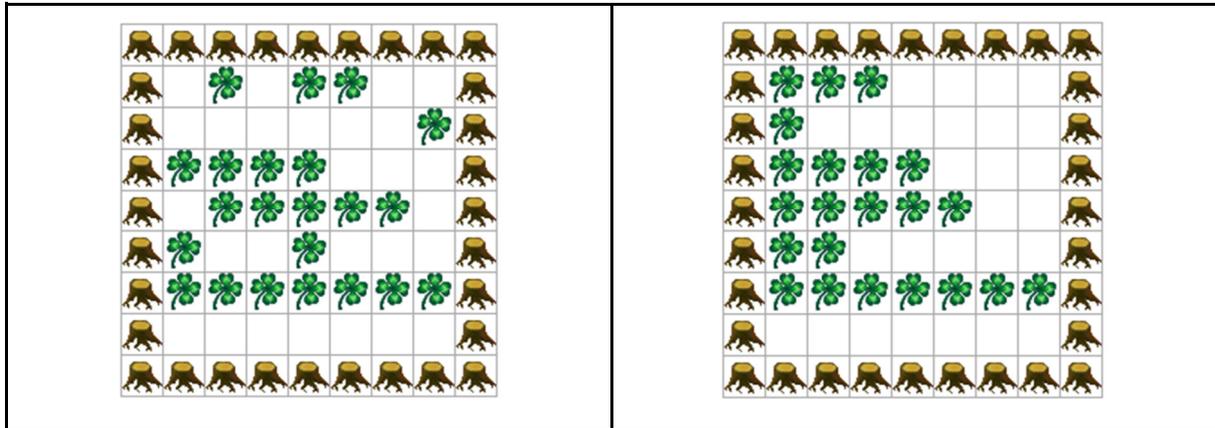
```
public void myMainProgram() { // ohne Eclipse: myProgram
    int kleeblaetter = 0;
    int stufen = 0;

    // ...

    tools.showMessageDialog("Ich habe " + stufen +
        " Stufen erklommen und dabei " +
        kleeblaetter + " Kleeblätter gefressen!");
}
```

Aufgabe: Kleeblätter an den Rand schieben

Schreiben Sie ein Programm, das in jeder Zeile die Kleeblätter an den linken Rand „schiebt“ (Sie können die Kleeblätter natürlich auch in jeder Zeile nach rechts schieben, oder in jeder Spalte nach oben oder unten):



Sie können dazu folgendes Programmgerüst erweitern:

```
public void myMainProgram() { // ohne Eclipse: myProgram
    for (int y = 1; y < world.getSizeY() - 1; y++) {

        // ...

    }
}

int zaehleUndEntferneBlaetter (int y) {
    int anzahl = 0;

    // zähle Blätter in Zeile y, und entferne sie

    return anzahl;
}

void legeBlaetter(int y, int anzahl) {

    // lege von links her anzahl Blätter in Zeile y

}
```

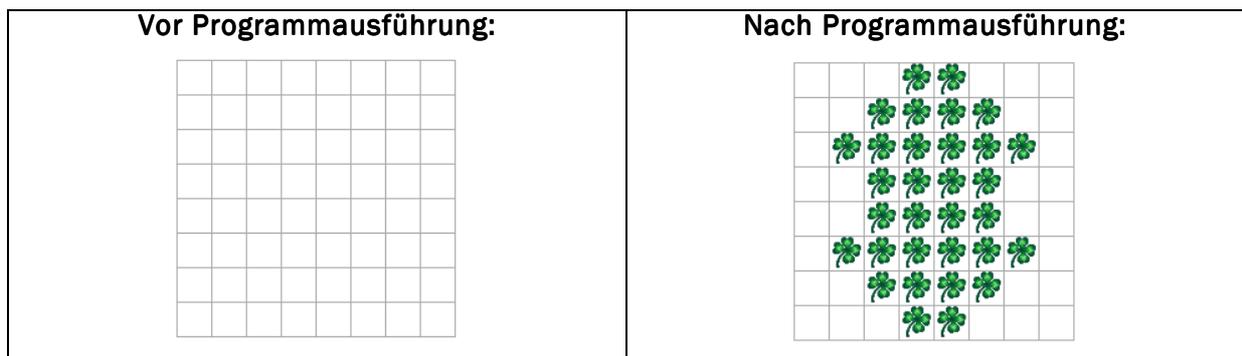
Zur Erinnerung: `world.getSizeY()` liefert die Anzahl Zeilen der Welt; `world.getSizeX()` liefert die Anzahl Spalten der Welt.

Aufgabe: Kleeblätter legen – aber symmetrisch

Schreiben Sie ein Programm, das in jeder Zeile eine vom Benutzer eingegebene Anzahl Kleeblätter legt bzw. doppelt so viele Kleeblätter wie eingegeben, symmetrisch zur vertikalen Mitte und zur horizontalen Mitte.

Angenommen, Sie geben nacheinander die vier Zahlen 1,2,3,2 ein, und die Welt ist 8x8 Felder gross: Dann soll Ihr Programm

- in den Zeilen 0 und 7 je 2 Blätter,
- in den Zeilen 1 und 6 je 4 Blätter,
- in den Zeilen 2 und 5 je 6 Blätter,
- in den Zeilen 3 und 4 je 4 Blätter legen:



Damit Ihr Programm funktioniert: Verwenden Sie Welten, deren Breite und Höhe eine gerade Zahl (2, 4, 6, ...) ist.

Zur Erinnerung: `world.getSizeY()` liefert die Anzahl Zeilen der Welt; `world.getSizeX()` liefert die Anzahl Spalten der Welt.

Hinweis: Sie können diese Aufgabe schrittweise lösen:

1. Schreiben Sie zunächst ein Programm, das einfach im linken oberen Quadrant die eingegebene Anzahl Kleeblätter auf die korrekten Felder legt.
2. Erweitern Sie das Programm so, dass auch im rechten oberen Quadrant die eingegebene Anzahl Kleeblätter auf die korrekten Felder gelegt wird.
3. Erweitern Sie das Programm so, dass auch in der unteren Hälfte die Kleeblätter gelegt werden.

Sie können, wenn Sie möchten, das folgende Gerüst als Startpunkt verwenden:

```
public void myMainProgram() { // ohne Eclipse: myProgram
    int halbeBreite = world.getSizeX() / 2;
    for (int y = 0; y < halbeBreite; y++) {
        // ...
    }
}

void legeBlaetter(int y, int anzahl) {
    // ...
}
```