

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

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

Lösung:

```
1 import javakara.JavaKaraProgram ;
2 public class ParameterBeispiel extends JavaKaraProgram{
3     void legeKleeblattzeile(int anzahl) {
4         for (int i =1; i<=anzahl ; i++) {
5             kara.putLeaf();
6             kara.move();
7         }
8     }
9 }
10 public void myProgram() { // Anfang von myProgram
11     legeKleeblattzeile(7);
12 } // Ende von myProgram
13 } // Ende
```

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 der Klammer 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.

1. Aufgabe: (1 Parameter)

- a) Tippt das Programm ab und testet es mit unterschiedlichen Parametern.
- b) Verändert **myProgram()** so, dass Kara ein Rechteck aus Kleeblättern legt. Die Kleeblätter der Kanten des Rechtecks sollen mit Hilfe der Methode **legeKleeblatt(...)** ausgelegt werden.
- c) Programmiert eine Methode **legeQuadrat(int breite)**, die ein Quadrat mit einer variablen Kantenlänge legt.

2. Aufgabe: (mehrere Parameter)

Kara soll Muster aus Kleeblättern legen: Quadrate, Rechtecke, usw.
Jedes Muster soll durch eine Methode realisiert werden, z. B.

void legeRechteck(int breite, int hoehe)

Folgende Methoden sollen vorher zur Verfügung gestellt werden:

1. **void turnAround()**
dreht Kara um 180 Grad.
2. **void legeX(int anzahl)**
Kara startet auf der Position des ersten abzulegenden Blattes und legt danach in seiner aktuellen Blickrichtung anzahl Kleeblätter ab. Am Ende steht er einen Schritt hinter dem letzten Blatt.
3. **void geheX(int anzahl)**
Kara startet auf seiner aktuellen Position und geht danach in seiner aktuellen Blickrichtung Anzahl Schritte vorwärts.

a) Programmiert eine Methode **legeRechteck(int breite, int hoehe)** , sodass Kara ein mit Kleeblättern umrandetes Rechteck auslegt. Der Methodenaufruf `legeRechteck(5,3)` soll demnach ein Rechteck mit der Breite 5 und der Höhe 3 erzeugen.

b) Programmiert eine Methode **legeAusgefüelltesQuadrat(int seitenlaenge)** , sodass Kara ein mit Kleeblättern gefülltes Quadrat auslegt. Der Methodenaufruf `legeAusgefüelltesQuadrat(4)` soll demnach ein mit Kleeblättern ausgefülltes Quadrat mit Seitenlänge 4 auslegen.

c) Programmiert eine Methode **legeAusgefüelltesRechteck(int breite, int hoehe)** , sodass Kara ein mit Kleeblättern gefülltes Rechteck auslegt. Der Methodenaufruf `legeAusgefüelltesRechteck(5,3)` soll demnach ein ausgefülltes Rechteck der Breite 5 und der Höhe 3 erzeugen.

Zusatzaufgabe:

d) Überlegt Euch ähnliche Methoden für andere geometrische Figuren, bspw. Rauten, Parallelogramme, Dreiecke... und programmiert diese.

3. Aufgabe: (Rückgabetypen und Parameter)

Kara kann mit den Methoden der ersten beiden Aufgaben, umrandete und gefüllte Quadrate und umrandete und gefüllte Rechtecke variabler Größe legen.

Auf der Welt von Kara befindet sich bereits ein gefülltes (bzw. umrandetes) Quadrat (bzw. Rechteck). Kara soll erkennen, um welches Muster es sich handelt und anschließend unter dem Muster eine originalgetreue Kopie auslegen.

Diese Aufgabe ist recht umfangreich. Überlegt Euch zunächst schriftlich eine Strategie und zerlegt die Aufgabe in Teilaufgaben, die ihr dann nacheinander programmiert und testet.