Kapitel 1: Die ersten Schritte (Lösungen)

Lösung zu Aufgabe 7:

public class MyKara extends Kara {

 public void act() {

 move();

 turnLeft();

 move();

 turnRight();

 move();

 move();

 turnRight();

 move();

 turnLeft();

 turnLeft();

 move();

 turnRight();

 move();

 move();

 turnRight();

 move();

 turnLeft();

 move();

 turnLeft();

 move();

 turnRight();

 move();

 move();

 turnRight();

 move();

 turnLeft();

 removeLeaf();

 stop();

 }

}

**Erläuterungen:**

1. Die ***Kommentare*** im Quelltext wurden weggelassen (= der Text, welcher im Greenfoot-Editor entweder grau oder blau dargestellt wird).
Kommentare schreibt man in den Quelltext als zusätzliche Informationen. Die Kommentare sind für Menschen und werden vom Computer ignoriert. Es gibt **drei Möglichkeiten**, um Kommentare in den Quelltext zu schreiben:
	1. Mit zwei Schrägstrichen **//** (nach den Strichen gilt der Rest der Zeile als Kommentar)
	2. Längere Kommentare über mehrere Zeilen schliesst man mit **/\*** und **\*/** ein.
	3. Kommentare für Methoden und Klassen werden zwischen **/\*\*** und **\*/** geschrieben.
2. Die Methode ***stop()*** bewirkt, dass nach der act()-Methode gestoppt wird, auch wenn der Run-Knopf gedrückt wurde.

Lösung zu Aufgabe 8:

public class MyKara extends Kara {

 public void act() {

 move();

 goAroundTree();

 goAroundTree();

 move();

 goAroundTree();

 removeLeaf();

 stop();

 }

 public void goAroundTree() {

 turnLeft();

 move();

 turnRight();

 move();

 move();

 turnRight();

 move();

 turnLeft();

 }

}

**Erläuterungen:**

1. Zur besseren Übersicht und um zu vermeiden, dass wir dreimal den gleichen Code schreiben müssen, haben wir eine neue Methode **goAroundTree()** eingeführt.
2. Dem Namen der Methode werden hier zwei Schlüsselwörter **public void** vorangestellt.
**public** bedeutet, dass die Methode auch von ausserhalb aufgerufen werden kann.
**void** (engl. Leer) bedeutet, dass die Methode keinen Wert zurückliefert.
3. Hinter jeder Methode steht das Klammerpaar **()**, was bedeutet, dass die Methode keine Parameter übergeben bekommt. Später werden wir Methoden mit Parametern kenne lernen.