Chapter 1: First Steps (Solutions)

Solution to Task 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();
     }
}
```

Notes:

1. The ***Comments*** in the code have been omitted (= the text that is in the Greenfoot editor either gray or blue).
   Comments in the source code are written as additional information. The comments are for people and are ignored by the computer. There are **three ways** to post comments in the code:
   a. With a double slash **//** (after the double slash the rest of the line is a comment)
   b. Longer comments over several lines are enclosed between **/*** and ***/**.
   c. Comments on methods and classes are written between **/**** and ***/**.
2. The method ***stop()*** makes sure the simulation stops after the act()-Method even when the Run-button was pressed.

Solution to Task 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();
    }
}
```

Notes:

1. For greater clarity and to avoid that we need to write the same code three times, we have developed a new method **`goAroundTree()`**.

2. The name of the methods here are preceded by two keywords: **`public void`**.
   `public` means that the method can be called from outside.
   `void` means that the method returns no value.

3. Behind each method is a pair of parentheses **`()`** which means that the method gets passed no parameters. Later we will learn how to write methods with parameters.