

Chapter 5: Methods

Once upon a time, there was a little girl who always wore a red riding cloak. So everyone called her Little Red Riding Hood. One morning the mother said to the child: "Red Riding Hood, today is your grandmother's birthday. Bake her favorite cake, take a bottle of good old wine from the cellar, put everything in one basket, and go visit her."



Help Little Red Riding Hood bake a cake! Carefully read the theory first.

Method with Parameters

With parameters values can be passed to a method. In the following method, you can specify how many steps Kara should take:

```
public void multiMove(int steps) {  
    int i = 0;  
  
    while (i < steps) {  
        move();  
  
        i = i + 1;  
    }  
}
```

To go 5 steps just put the number 5 between the parentheses:

```
multiMove(5);
```

To invoke a method with multiple parameters, the values are separated by commas.

```
drawRectangle(21, 4);
```

Explanation:

- (1) In previous programs it was rather cumbersome to always write the parentheses () for each method. Now it is clear that the previously used methods were merely special cases where no parameter is passed.
- (2) In the brackets the parameter (in this case `steps`) is specified with a type (here `int`). Multiple parameters are separated by commas as seen in this example:
`public void drawRectangle(int width, int height)`
- (3) When such a method is called, the values are copied into the variables (in this case into `width` and `height`).

Baking the Cake

We will now use Kara to “bake” a birthday cake for the grandmother. The cake will be made of leafs.

Open the **scenario35...** from the project **scenarios-chapter-5**.

In preparation for baking, the following methods are provided:

1. `public void turnAround()`

Kara rotates by 180 degrees.

2. `public void multiMove(int steps)`

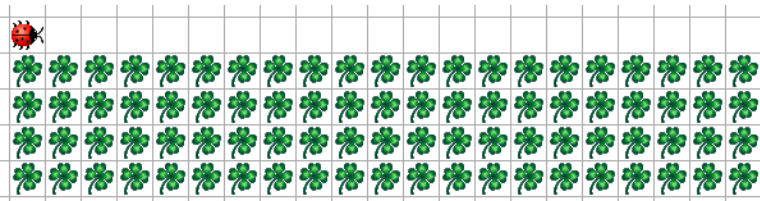
Kara takes the number of steps in the current direction (see example in theory section above).

3. `public void putLeafs(int count)`

Kara puts the specified number of leafs. The first one he places at the current position, the other following in his line of sight.

TASK 35: BAKING A CAKE

Kara draws a rectangle with leafs symbolizing the cake. Kara starts in the lower left corner and looks to the right.

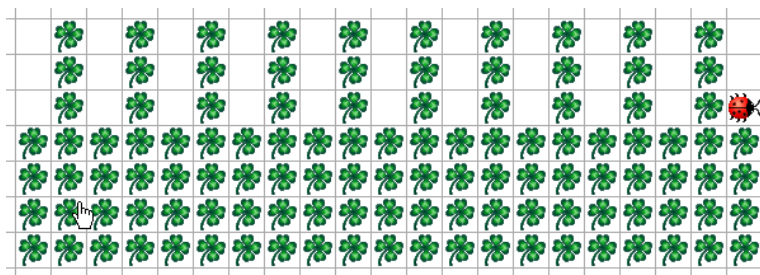


Kara should be able to make a **rectangle with variable width and height!**

The method call **drawRectangle(21, 4)** should therefore create a rectangle with width 21 and height of 4.

TASK 36: CANDLES ON CAKE

To make the cake look like a birthday cake it needs a few candles, of course. Extend your program with an additional method **drawCandles(int count)** which sets the specified number of candles on the cake.



Methoden mit Return Values

A method can return a value as a result. The following method shall calculate a random number (between 0 and 9) and return the result:

```
public int randomNumber() {
    int random = Greenfoot.getRandomNumber(10);
    return random;
}
```

The result can either be stored in a variable or directly be used as follows:

```
multiMove(randomNumber());
```

Explanation:

- (1) The type of the return value (in this case **int**) is specified before the name of the method. If the method does not return a value, we write **void**.
- (2) To return a value we write **return** followed by the value.

TASK 37: CANDLES FOR AGE

We want to put on the cake a **candle for every decade** of grandmother's age. As Little Red Riding Hood does not know exactly how old the grandmother is, she must first ask her mother.

Your program should place the right number of candles on the cake after asking about the how old grandmother is.

Note: The following method can be used to ask for the age. It returns an **int**:

```
intInput("Message...")
```

ADDITIONAL TASK 38: LAYERED CAKE

Let Kara make a layered cake for her grandmother. Kara should add a layer for every decade that her grandmother is over 50. Each layer should have two lines and is indented on both sides by two leaves.

