# 4 Instruction – Methods

## 4.1 Stars and Letters

All the methods you have to write in this exercise should be without parameters. This will probably not result in a good design. It is done for the purpose of the exercise only, i.e., understanding methods, local variables, and instance variables.

1. Write a class `Stars` with a demo method that calls a method `printStars` that prints a line with 10 stars, like this:

   `**********`

2. Modify this method such that the number of stars that it can print an arbitrary number of stars. The actual number is obtained from the instance variable `numstars`.

   Hint: the method call `System.out.print("*");` prints one star *without a newline*. The call `System.out.println();` prints only a newline. Try out this printStars method.

3. Probably, your method printStars of the previous exercise uses a counter to keep track of the actual number of stars printed when you're building the output. If it doesn't, modify the method such that it uses such a counter and make it a *local variable*.

4. Modify the method such that it doesn't print a newline after the stars. Add a method `void nl()` that just prints a newline.

5. Add a method that prints a number of spaces, the actual number depending on the instance variable `numspaces`.

6. Print a big letter H consisting of stars, using several calls to the method printStars, printSpaces, and nl.

7. Add a method `void printH()` that prints a big letter H.

8. Add methods for E, L, and some other letters.

9. Add a method that prints a word.

10. Modify the method printStars such that it prints a sequence of the character that is stored in an instance variable. Write a method that prints a big H, consisting of H characters. Call the method `printH` created above.

## 4.2 Put your money where your mouth is

You are going to develop a program that "pronounces" an amount of money in Dutch. I.e., given an amount in digits, it should output the same amount in words. The amount is specified as a number of euros and a number of cents. E.g., when the input is `12 56`, it will output "twaalf euro en zesenvijftig cent". The maximum amount to be handled is 1000000.99. The minimum amount is 0.00.

The program will repeatedly ask for an amount in the form of two integers, one representing the amount of euros and the other the amount of cents, until the word "stop" is given instead of the two integers. When an amount outside the specified interval is input, it will issue a message and ask again for an amount. Your program does not have to take other input into account.

**Considerations**

1. Decompose the problem in subproblems and write a function for each of these subproblems. Such a function typically takes an integer, sometimes two, as parameter and produces a String with Dutch text.

   These functions should not modify instance variables and should deal with input or output. In other words, they are *side-effect free*.

   You may need about five to ten functions and at least one method that takes care of input and output and calling the functions.

2. Formulate for every function what should hold before the call and what is assumed about the parameter(s) and what it returns. Add this in comment before the method, preceded with `@pre` (for precondition) and `@return` (for return value).

   For example, the header of a function that returns the text of the numbers below 20 could be:

   ```
   // @pre 0<= n < 100
   // @return "text of number n"
   String textBelow20(int n)
   ```

3. First design the functions and other methods you need. Enter the headers and comment as described above and show your design to the instructor before you start to implementing.

4. To choose between the various parts of the text, such as, “een”, “twee”, it is not necessary to use the somewhat bulky construction of multiple if statements. Thinks about one or more arrays of Strings to deal with this. Another possibility is to use the `switch` statement. It is allowed to use this construct but it is not covered in this course.

5. The use of diacritical marks, as in `drieëntwintig` is not required. You are nevertheless challenge to use them.

---

Kees Huizing – contact2ip65@gmail.com – 21 Sep, 2010