

2IP65/2IP70 – Instruction 1

Preliminaries

Material for this course can be found on the wiki pages at <http://huizing.name/javacourse>.¹

To do these instructions, you need a working Java development system. We recommend that you use a recent version of NetBeans that is not a *beta* version (i.e., slightly experimental), preferably **6.5** or later. NetBeans is a so-called Integrated Development Environment (IDE) from Sun, the designers of Java. These instructions are based on **Java 1.6**, which is automatically installed with NetBeans if you follow the *NetBeans tutorial* that can be found on the wiki. Java 1.5 can also be used.

Rules about the exercises

- Start every file you write with a few lines of comment with at least the date and your name(s).
- Work in pairs, if possible. Do not work in groups of three or more.
- Only one member of the pair has to write the program, but switch roles regularly (at least once per week).
- Keep your programs neat from the start on:
 - Use clear, sensible names for variables etc.
 - Let class names always begin with an uppercase letter, variable names with a lowercase letter; use all capitals for constants.
 - Use indentation rules consistently. After an opening brace (`{`), indent every line by two or more positions and put the closing brace (`}`) on a single line, not indented. See the program below for an example.
 - Don't make lines longer than the width of your screen; otherwise it is rather hard for the instructor to help you with your program. Don't hesitate to break a line; in Java, a line can be broken anywhere a space could appear (except in strings).
- Ask the instructor to inspect your exercise when you have finished it.

Exercise 1 – Dag, wereld

Make sure that Netbeans is installed on your computer. See the *Netbeans tutorial*.

Following this tutorial, create a project and an *Empty Java File* with the class name `Groet`. The file `Groet.java` is now open in your edit window; enter the text below. If the file is not in your edit window, look for it under the `Projects` or the `Files` tab and double-click it.

¹redirecting link to http://www.win.tue.nl/~keesh/dokuwiki/doku.php?id=2ip65_en_2ip70

```

/* Instruction 1 - Ex 1
 * Taas Daamde en Surdie Finnis
 * 26 augustus 2008
 */

class Groet {
    public void doedegroeten() {
        System.out.println("Dag, wereld");
    }

    public static void main(String[] args) {
        new Groet().doedegroeten();
    }
}

```

Run the project as described in the tutorial.

Alternatively, you can first compile your program by choosing **Compile Groet.java** from the menu **Build** or from the pop-up menu (right mouse button) in the editor. When compilation is successful, you can run the file from the **Run**-menu or the pop-up menu.

If there are errors, check whether all spaces, parentheses (haakjes), curly braces (accolades) and semicolons (;), are in the right places, whether you have the uppercase and lowercase right, etc. Remove all errors and try again until the compilation gives no errors anymore. You should see as output

```
Dag, wereld
```

probably surrounded by messages from NetBeans; on my screen it looks like this:

```

init:
deps-jar:
compile-single:
run-single:
Dag, wereld
BUILD SUCCESSFUL (total time: 0 seconds)

```

Exercise 2 – Input and choice

Read the following program. The file `Interest.java` can be downloaded from the wiki.

Copy the text from the file into an empty Java file or download the file into your source folder (see the *Netbeans tutorial*, section 3). In the latter case, note that the file will be named `fetch.php` if you download it directly. Rename the file to `Interest.java` afterwards or choose this name at the download.

```

import java.util.Scanner;

class Interest {
    Scanner scanner;
    double saldo;
    double rente; // percentage

    public void calcInterest() {
        scanner = new Scanner(System.in);

        saldo = 250;
        System.out.println("Geef de rente in procenten ");
        rente = scanner.nextDouble();
        saldo = saldo + saldo * rente/100;

        System.out.println("Het saldo na een jaar is " + saldo);
    }

    public static void main(String[] args) {
        new Interest().calcInterest();
    }
}

```

1. Change this program in such a way that it also inputs the initial amount (saldo) instead of working with a fixed amount.
2. The bank has the policy that amounts below 100 will not receive interest. Change the program to accommodate this policy.

Exercise 3 – Leap year

In our Gregorian calendar, a leap year (schrikkeljaar) is a year that is divisible by 4, unless it is a century year (divisible by 100) and since this would make the year a little bit too short, the century years 1600, 2000, etc. are nonetheless leap years.

Hint; the operator % gives the remainder after division, hence, for n an integer, $n \% 4$ is equal to 0 if n is divisible by 4 and vice versa.

1. Write a program that takes a number as an input and outputs whether it is a leap year or not. Use if-statements.
2. Write a variant that uses a boolean expression that is true exactly when the number is a leap year.

* Exercise 4 – more ifs

Exercises with a * are optional.

Write a program that plays a guessing game. The computer gives the player a list of eight dishes (or persons, animals, sports, words for being drunk, or whatever) and she picks one of them. Then the computer has to find out which one she picked by asking at most three questions that can be answered by yes or no.

Example (input from the user is in bold face, the empty line was caused by the user pressing the Enter key):

```
Choose a dish from (press enter when ready):  
Pizza, Moussaka, Dame Blanche, Sabayon, Pêche Melba, Lasagna,  
Yoghurt-In-The-Plate-You-Had-Your-Macaroni, Hutspot.
```

```
Is it a dish from the oven?  
yes  
Does it contain potatoes?  
no  
Can you throw it like a frisbee?  
no  
It is Lasagna!
```

Note that the questions in the example are not optimal. E.g., when the player answers “no” after the first question, there are 5 possible dishes left and it is not always possible to distinguish these by only two yes/no questions.

Write a program that performs this guessing game. You may use the eight dishes from the example or use some of your own choice (with appropriate questions).