Java In Depth
Oefeningen Streams

Peter Hardeel

cvo leerstad

volwassenenonderwijs
www.leerstad.be
info@leerstad.be

1. Write a program that puts the words “Hello world! This is my first file input/output exercise” to a text file `first.txt`.

2. Redirect the standard input so that it comes from `first.txt`, and the standard output so that it goes to `output.txt`. Read from standard input and output what you read to standard output.

3. Write a program that displays the contents of *one* file on the screen. The name of the file is given by a command-line argument.
    a. What control structure do you have to add when you want to display the contents of *one or more* file(s) that are given as command-line arguments?

4. Write a program that writes the integer values 1 to 100 to an output file `integerdata1.dat`.

5. Write a program that reads the integer values stored in `integerdata1.dat` and shows them on the screen in small groups of 10 integers.

6. Write a program that reads the integer values stored in `integerdata1.dat` and writes them to two output files named `integerdata2.dat` and `integerdata3.dat`. The first output file should contain the values 1 to 10, 21 to 30, 41 to 50, ... while the second output file should contain the values 11 to 20, 31 to 40, ...

7. The same program as the program above, but now each group of 10 values should be delimited with the special value 9999 (*sentinel value*). After you have made the two files, read the contents of `integerdata2.dat` in again, and store the values in a list by checking for the sentinel value while reading. Display the contents of that list.

8. Write a program that appends a line of text to a file. Look at the constructors for a `FileWriter` to solve this problem.

9. Log some text to a file (a default log file named `messagelog.txt`). The logged data must be appended to the file and each log line must be timestamped. For this timestamp, you can use a `Date` object that comes from the `java.util` package.

10. Make a class, named `FormatWriter`, that extends `PrintWriter` so that it contains a method named `print(int value)` that prints out the value of an argument, using a specified width. The width can be specified at the moment of constructing a `FormatWriter` or it is set to a default value (which is chosen 10). It can also be changed with a method `setWidth(int width)`.

11. Write a program that, using a filter, reads the values of integers stored in a file and that gives also the sum of those values.

12. Write a program that uses a filtered reader that reads a file where all the characters are put in lowercase and gives those characters in uppercase.

13. Write a program that outputs the double values, stored in an array, called `prices`, and expressed in BEF to a file by constructing a filtered `outputstream` that converts those values first to Euro.
    1 Euro equals 40.3399 BEF.
    The contents of `prices` is {2500, 1400, 1655, 512, 4000, 12300}

14. Write a small program that checks if a file exists. The filename is given as an argument.

15. Write a small program that makes a directory. The directory name is given as an argument.

16. Write a program that determines if one file is older than another file. The two filenames are given at the command-line.

17. Write a program that inserts a line into a file at a certain linenumber. There are three arguments to be given: the filename, a line number and the string to be inserted at the line number specified.

18. Get the content of a directory with subdirectories. This program uses a recursive algorithm.

19. Get the content of a directory with a filter. First create a class that implements `java.io.FilenameFilter` and then code the `accept()` method. The program should show all the files in a directory that is given as a first command-line argument with an extension that is given as a second argument.

20. Write a program that puts the integer values 50, 60 and 70, the double values 50.0, 60.0, 70.0 in a random access file called `values.dat`. Then you have to change the values 70 and 70.0 into 80 and 80.0.

21. Write a program that puts the following personal data into a properties file called `personal.properties`. Your first name with key `name.firstname`, your last name with key `name.lastname`, your phonenumber with key `phonenumber`, your home address with key `home_address`.
    Afterwards, you read back these properties, and add your age with key `age` and save this property to the same file. Then you read back the same properties with another program, and display them on the screen.

22. Change exercise 7 in such a way that the names of the input file and the outputfiles are stored in a properties file named `input_output.properties` with the key values `input`, `output1` and `output2`. The value of the sentinel is also retained in that same properties file.