CS1 — Practical Session 5: Loops & Arrays

Kurt Driessens

September 24th, 2016

In this fifth practical session you will get some personal experience with loops and arrays in Java. During this lab, complexity goes up rapidly, so you might not finish all exercises today. That's okay: pick up during tomorrows lab where you left off today.

Exercise 1

Write a method that sums all the elements of an array with an even index. Also write a method to sum all the elements with an odd index. Use these methods to see whether the even or odd indexed elements have a larger sum for an array of length 6 entered by the user. (**Advanced -** Adapt your solution to work for any length array, as decided by the user at runtime. As in Ex 2 from the previous lab, the user should enter all values on 1 line, separated by spaces and terminated with a letter.

Exercise 2

Write a method that inserts a new value into a sorted array. (Keep in mind that this will increase the size of the array). Test your method by sorting a number of numeric values read from the prompt using the Scanner class and printing out the sorted list.

Exercise 3

Implement a method that performs matrix addition and returns the sum of its two parameters as a result. Make the method check whether the sum is legal, i.e. the width and height of the matrices match, and warn the user if they don't. Write a main method to test it. To make testing easy, you can instantiate, for example, a 3 by 3 matrix as follows:

```
double[][] matrix = {{1.0,2.0,3.0},{4.0,5.0,6.0},{7.0,8.0,9.0}};
```

Assignment!

Implement a method that performs matrix multiplication and returns the product of its two parameters as a result. If you don't know how to multiply two matrices, ask Google. Make the method check whether the multiplication is legal, i.e. the width of the first matrix should be equal to the height of the second, and warn the user if it isn't. Again, test your method in a main method.

When it all works as expected, upload your `.java` under the 5^{th} lab assignment.

Bonus — Your first computer game!

Implement the game Tic-tac-toe. Don't worry, when undertaken step-by-step, this is easier than it sounds. Your program should read in moves from human players, show the resulting board, check rules and congratulate the winner.

Accomplish this task by taking the following steps:

1. Design a board representation. You should be able to represent and recognise empty squares and squares marked by two different players. Don't take too long with this step ... :-)
2. In each step of the game, your program should print out the current board. Write a method that does this.
3. In each step of the game, your program should ask the user for the coordinates (ranging from 1-3 and 1-3 for each dimension) of the next move. Your program needs to check the legality of the move and ask for another move if it is not legal (this includes checking if the square is empty). Write a method that does this.
4. After each step of the game, your program should execute the move on the board and check whether the played move wins the game, i.e., it should check if the player created a horizontal, vertical or diagonal line of his symbol only. Write a method that does this.
5. The game and your program should end when one player wins or when the board is full. Write a method that checks these conditions.
6. Now combine all of the written methods with a loop to turn them into the Tic-tac-toe game. When the loop ends (i.e. when the game ends and thus just before your program ends: print the result, i.e. the winner if there is one.