Prof. Dr.-Ing. Ralf Schenkel
Tobias Zeimetz
Trier University

Wintersemester 22/23

Exercises for the Class

# Elements of Computer Science: Programming

## Assignment 07

Submission of solutions until 3:30 p.m. at 19.12.2022
at `moodle.uni-trier.de`

- Every task needs to be edited in a meaningful way in order to get a point!
- Please comment your solutions, so that we can easy understand your ideas!
- If you have questions about programming or the homeworks, just ask you teachers!
- **Submission that can't be compiled are rated with 0 points!**

### Exercise 1 (Evaluation: predefined `main` method)

Consider the recursively defined sequence $(f_n)_{n=0,1,...}$ of integers $f_n$:

$$\begin{aligned} & f_0 := 0 \\ & f_1 := 2 \\ & f_2 := 4 \\ \text{and in case } n \geq 3: \quad & f_n := (f_{n-1} + f_{n-2}) \cdot f_{n-3} \end{aligned}$$

1. Implement a "`static double frec(int n)`" method to calculate the $n$" th value $f_n$ of this sequence in the form of a *recursive algorithm*.
2. Implement a "`static double fdyn(int n)`" method to calculate $f_n$ using *dynamic programming*[1].

You should write both partial solutions into the given file. A suitable `main` method for testing your two implementations is already predefined there.

[1] See the slides of chapter 3 part 4

### Exercise 2 (Evaluation: Predefined `main` Method)

This task deals with two-dimensional (**`double`**) arrays and how they can be utilised as matrices. Their task is to implement methods that make it possible to multiply two matrices together. In case you have lost track of how matrix multiplication is performed, you can find guides under following links[2][3].

For this task you have to implement the following methods:

```
boolean isValid(double[][] matrix1, double[][] matrix2)
double[][] mulMatrices(double[][] matrix1, double[][] matrix2)
```

The method `isValid()` is used to determine if a matrix multiplication is possible. Remember, that you can only multyply two matrices if the number of collumns of `matrix1` is equal to the number of rows of `matrix2`.

The method `mulMatrices()` is used to implement the actual matrix multiplication. It returns a two-dimensional **`double`** array, containing the result of the multiplication. In case that both matrices cannot be multiplied because `isValid()` returns **`false`**, simply return the **`null`** reference

A suitable `main` method for testing your two implementations is already predefined there.

### Exercise 3 (Evaluation: Text)

Write a Java program that reads in two sets of numbers as input and generates patterns (consisting of $n \times n$ characters) of the following type as output (similar to example `K3B14E_Flag.java`).

Where $f$ specifies the format or pattern of the output (here only 1, 2 or 3, other values may be treated as you wish); $n \geq 5$ is always odd. Write a separate function for each of the three patterns (as already specified).

Format $f = 1$ as $7 \times 7$ pattern:

```
1 7
X.X.X.X
.X.....
X.X....
...X...
X...X..
.....X.
X.....X
```

Format $f = 2$ as $9 \times 9$ pattern:

[2] `https://www.mathsisfun.com/algebra/matrix-multiplying.html`
[3] `https://en.wikipedia.org/wiki/Matrix_multiplication`

```
2 9
....x....
...xxx...
..xxxxx..
.xxxxxxx.
xxxxxxxxx
.xxxxxxx.
..xxxxx..
...xxx...
....x....
```

Format $f = 3$ as $7 \times 7$ pattern:

```
1 7
......x
.....xx
....xxx
...xxxx
..xxxxx
.xxxxxx
xxxxxxx
```

The patterns should be created with nested loops. `System.out.print` may only be used in the following three forms (but of course multiple times):

- `System.out.print(".")`,
- `System.out.print("X")` und
- `System.out.println()`.

You must therefore build the output from individual "`.`", "`X`" and line separators. Also make sure that no line endings are missing and that you do not create unnecessary blank lines.