

Exercises for the Class  
**Elements of Computer Science: Programming**  
Live Assignment 3

Submission of solutions for group 1 until 1:00 p.m. and for group 2 until 2:00 p.m.  
at `moodle.uni-trier.de`

- Submission that can't be compiled are rated with **0** points!
- Please comment your solutions, otherwise you can lose points!
- If you try to cheat, you will lose your points and the classroom exercise will be over!

**Exercise 1 (Evaluation: Numbers)**

(10 Points)

The given program reads in two arrays. In this task the arrays should be interpreted as vectors. Your task is to determine the scalar product of both vectors. Use the following formula:

$$\vec{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \text{ and } \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \text{ lead to } \vec{a} \circ \vec{b} = a_1b_1 + a_2b_2 + a_3b_3$$

An example of the application of the scalar product is:

$$\vec{a} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix} \text{ and } \vec{b} = \begin{pmatrix} 10 \\ 20 \\ 30 \end{pmatrix} \text{ lead to } \vec{a} \circ \vec{b} = 2 \times 10 + 3 \times 20 + 4 \times 30 = 200$$

You can assume, that both read in arrays have the same size.

**Exercise 2 (Evaluation: predefined main method)**

(10 Points)

Implement the method

```
static String contentsLine(String s,String t)
```

The method should return as result the concatenation of the parameters `s` and `t`, but inside filled with points `'.'` to a length of exactly 30 characters. (This could be used e.g. as formatting for a table of contents.)

**Example:**

```
contentsLine("Chapter_5", "123") ~ Chapter 5.....123
```

If the total length of the two strings is greater than 27 and there is not enough space for at least 3 separating points, the **null** reference should be returned.

### Exercise 3 (Evaluation: predefined main method)

(10 Points)

Implement the method

```
static String secToString(int k)
```

The method receives as integer parameter  $k$  a time duration in seconds and shall convert it into hours ( $h$ ), minutes ( $m$ ) and remaining seconds ( $s$ ) (with  $h < 24$ ,  $m < 60$  and  $s < 60$ ). The values should be converted as a string in the format " $hh:mm:ss$ " can be returned, for example:

- $121 \rightsquigarrow "00:02:01"$  (2 Minutes, 1 Second)
- $86399 \rightsquigarrow "23:59:59"$  (23 Hours, 59 Minutes, 59 Seconds)

If the passed parameter value  $k$  is too large for this format ( $\geq 86400$  seconds) or negative, the string " $xx:xx:xx$ " can be returned.

### Exercise 4 (Evaluation: predefined main method)

(5 Points)

Implement a method

```
static int[] decomposeDate(String s).
```

The method is to specify a date (given as `String`) in the form  $dd.mm.yyyy$  into the three individual components (but now as `int` in an array). The parts  $dd$ ,  $mm$  and  $yyyy$  are specified decimal (e.g. " $31.12.2017$ ") or " $01.01.2018$ "). You can assume that  $s$  contains only correct dates (no special characters or too large values etc.).

Example:

At `int[] x = decomposeDate("24.12.2017")`  $x$  contains after execution a `int` field with three values as follows:

	0	1	2
x	24	12	2017

**Exercise 5 (Evaluation: Text)****(15 Points)**

Write a program that reads a **int** number as input and generates a pattern (from  $n \times n$  characters) of the following type as output (similar to the example K3B14E\_Flag):

<pre>Input : 6 XXXXXX XXXXXo XXXXoo XXXooo XXoooo Xooooo</pre>	<pre>Input : 7 XXXXXXX XXXXXXo XXXXXoo XXXXooo XXXoooo XXooooo Xoooooo</pre>
--	--

Also make sure that no line ends are missing and that you do not create unnecessary blank lines, otherwise the evaluation will not work. The letters used in the pattern are the capital X and the lower o.