Prof. Ralf Schenkel   Wintersemester 23/24
Tobias Zeimetz
Trier University

Exercises for the Class

Elements of Computer Science: Programming

Live Assignment 3

Submission of solutions for group 1 until 2:00 p.m. and for group 2 until 3:00 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: Text)   (10 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`):

```
    Input: 6
    XXXXXX
    XXXXXo
    XXXXoo
    XXXooo
    XXoooo
    Xooooo
```

```
    Input: 7
    XXXXXXX
    XXXXXXo
    XXXXXoo
    XXXXooo
    XXXoooo
    XXooooo
    Xoooooo
```

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.

Exercise 2 (Evaluation: Numbers) (10 Points)

The given progam 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:

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

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 3 (Evaluation: predefined `main` method) (10 Points)

Implement a method

```
public static String rearrange(String word, int[] sequence)
```

that is able to rearrange the characters in a given word `word` according to the given index values in the sequence array `sequence`. In case `sequence` contains numbers < 0 or $>= word.length()$ return the String "Invalid Sequence".

Examples:

Word: juiceApple

Sequence: 5 6 7 8 9 0 1 2 3 4

Result: Applejuice

Word: juiceApple

Sequence: 5 6 7 8 9 0 1 2 3 10

Result: Invalid Sequence (since the highest index in the given word is 9)

Exercise 4 (Evaluation: predefined `main` method) (10 Points)

Implement a method

```
public static String abbreviateName(String name)
```

that is able to abbreviate all first names of a persons name. The format of the given name strings (i.e., `name`) will always follow the same format or order, i.e., the name string starts with first names and ends with a persons last name.

Examples:

Name: Peter Miller
Abbreviated Name: P. Miller

Name: Edgar Allan Poe
Abbreviated Name: E. A. Poe

Exercise 5 (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 (≥ 86400 seconds) or negative, the string `"xx:xx:xx"` can be returned.