Lab 1

[valid 2015-2016]

Graeco-Latin square
Write a Java application that generates and prints on the screen Graeco-Latin squares. A Graeco-Latin Square is a *n x n* matrix. Each cell of the matrix is a pair of elements taken from the carthesian product of two given sets: *S X T*; for example, S may be a subset of the Latin alphabet and T a subset of the Greek alphabet.

The constraints are:

1. In each row, and in each column, all the *2n* elements are different.
2. No two pairs are the same in the matrix.

Example:

Aα	Bγ	Cβ
Bβ	Cα	Aγ
Cγ	Aβ	Bα

The main specifications of the application are:

- (0.5p) Create and display the square from the example. The Unicode character for α is "\u03B1".
- (0.25) Parse the command line arguments - they are given in the following format:

```
[n:number of elements] [S:"elements of the first set"] [T:"elements of the second set"]
```

If command line arguments are missing, the application will generate a random value for *n*, *S* contains the first *n* characters of the Latin alphabet (upper case) and *T* the first *n* characters of the Greek alphabet (lower case).
- Generate Graeco-Latin squares of size *n x n* and display them on the screen:
  - (0.5p) at least one...
  - (1.25p) all of them (display only the first, count them instead).
- The running time of the application will be displayed on the screen (try n=5).

Please consult the API documentation to learn more information about the classes and methods used!

Resources:

- Language Basics
- Numbers and Strings
- NetBeans IDE Java Quick Start Tutorial

Objectives:

- Get used with an integrated development environment (IDE): Netbeans.
- Get used with the Java language syntax.
- Create and run a simple application.
- Understand the following concepts: compiler, interpreter, byte-code, Java Virtual Machine (JVM), portability.
- Use the Unicode alphabet and Java special characters.
- Work with primitive data types and strings (String, StringBuilder).
- Work with one- and multi-dimensional arrays.
- Parse command line arguments.
- Perform conversions between strings and numbers.
- Split a string into tokens.
- Generate random numbers.
- Create multiple methods in the main class of the application.
