# General instructions

You have to complete any of the exercises within 8 hours.

You have to privde full source code, published on a GitHub public repository (or other form of public repository).

The project MUST use Maven as project model, other project models (like Gradle, ANT, Eclipse built-in) will not be accepted.

The project MUST contain all files needed for configuration, usually it is better to have a README.txt file which explains the project and the configuration. Usually the configuration will reside in a src/main/resources/configuration.properties file (but this is not required)

For exercises which use a Web UI a mvn clean install MUST create a WAR file, to be deployed on a simple Tomcat (no full JavaEE stack, like TomEE, Pajara, Glassfish). An “executable war” will be appreciated as well.

# SMTP Benchmarks

You have to implement a simple Java application which executes a bench on a SMTP service.

Requirements:

* The application MUST work with an SMTP service with username/pasword autentication
* The application MUST measure min/max/avg time of message delivery
* The application MUST measure the impact of the size of the message
* The application MAY evaluate the impact of sending multiple messages on the same SMTP connection
* The application MUST use JavaMail API

# Monitoring a Java application using JMX

You have to implement a simple command-line program which monitors values of a Mbean attribute published on a remote JVM using the JMX API.

Optional requirements:

- Open a GUI which shows a chart

- Configuration of an alert if the value is not within a given range

Suggestion:

it is better to use the Java Heap Memory size as sample attribute

# DB Benchmarks

You have to implement simple java application which executes benchmarks on a JDBC compliant database.

The application MUST evaluate the min/max/avg time of INSERT statements.

The application MUST evaluate the min/max/avg time of SELECT statements (using the PK of the COLUMN).

The application MUST issue DML requests using the PreparedStatement API and issuing “commits” every X statements.

The application will be run on a PostGRE or MS SQLServer database.

The script to create the table MUST be included in the source code.

# JCache Benchmarks

You have to implement simple java application which executes benchmarks on a JCache compliant service.

The application MUST evaluate the min/max/avg time of PUT/INVALIDATE statements.

The application MUST evaluate the min/max/avg time of GET statements.

The application MUST use the JCache API (JSR107) and compare at least two JCache Providers within this list: Hazelcast, BlazingCache, Infinispan.

The application MUST take into account the cost of serialization/deserialization of Java Beans.

https://www.google.com/settings/security/lesssecureapps

