CROSSWORD
CYBERSECURITY

Roman numerals exercise

Scenario

You are a member of a development team working on a project building mass report generation software in Java 8. A client asked that some numbers in reports should be formatted as Roman numerals (https://en.wikipedia.org/wiki/Roman_numerals), instead of usual Arabic numerals, e.g. XII instead of 12. The client needs an initial working implementation very quickly, but also expressed an interested of having more customisations options later. Unfortunately, the client is not sure at the moment what kind of customisations can be useful.

You are asked to design and write code that converts integer values in Java ("int" type) to their Roman numeral representations as strings, to support the client's request. First you need to write a simple to use, but inflexible implementation. Then you need create another implementation that is more flexible and allows customising behaviour of your code. It is up to you to think and decide what kind of flexibility can be useful.

Expectations

In both cases you are expected to deliver production ready code, with unit tests, comments, proper names and structure. You are not allowed to use any third-party libraries outside of Java SDK, except related to unit testing. All code must be your original creation, and we suggest you should not try to check how this functionality is implemented elsewhere. All comments and names should in English.

Delivery

Please put your code on a public repository hosted on GitHub, Bitbucket or Gitlab (according to your preference) and send us a link to the repository.

Evaluation criterias

This exercise is designed to allow you to demonstrate how good you are at:

- Coding - producing working code.
- Software Engineering - understanding of good practices of writing maintainable code.
- Research - finding information not included in specifications.
- Design - defining public part of your code that is easy to use by other developers.
- Java language - knowledge and proper use of Java features.
- Innovation - inventing new approaches and anticipating future requirements.
- Communication - understanding requirements and explaining intent of your code.