Messenger

Using scratch project, we will create messenger that will pretend to send emails using custom template generator.

The implementation of the template generator should be your own. You are not allowed to use any third-parties libraries. Use TDD approach for template generator. Each step should be a separate commit.

Set of requirements for the mail template generator, you should create list of TDD tests for these requirements:

The system replaces variable placeholders like #{subject} from a template with values provided at runtime.

If at least for one placeholder value is not provided at runtime – template generator should throw an exception

Template generator ignores values for variables provided at runtime that aren’t found from the template.

System should support values passed in runtime with #{…}. E.g. template is “Some text: #{value}” and at runtime #{value} passed as #{tag}. Output should be “Some text: #{tag}”.

The system supports the full Latin-1 character set in templates and in variables.

Messenger should work in two modes: console and file mode.

* In console mode the application takes expression from console and prints result to console. No application parameters should be specified to use this mode.
* In file mode application takes expression from file and output results to file. To use this mode user should specify input and output file names as application parameters.

Tasks:

1. You should create tests for Messenger with the following condition (In scope of this task you can create more than one test to cover the same functionality):

TDD approach – (2 points)

@Parameterized runner and Dynamic tests (0.3 points)

Implement meta annotations and filtering (0.3 points)

TemporaryFolder rule (0.3 points)

Mock reading from file/console (0.3 points)

Use partial mock (0.3 points)

Use spy (0.3 points)

Create custom extension (jUnit5) to output test execution information to file (0.3 points)

Using ExpectedException rule to check exceptions + Assertion mechanism (0.3 points)

Implement Disabling test on condition (0.1 points)

Test quality and adequate coverage will be assessed as (0.5 points)

Don't forget about good tests, checkstyle and other staff that already included into your build phase

Extra mile:

Write BDD tests for messenger. You may use any BDD framework (Spock, JBehave, Cucumber,…). Negotiate framework with your mentor – (2 points)