

## Task 1 – Jenkins Parameterized

### Setup

The main class I used **Mathematica**

```
J Matematica.java X
A: > Programe2 > SQM_HW > J Matematica.java
1 package SQM_HW;
2 // Marius-Remus Dumitrel
3
4 import java.util.ArrayList;
5 import java.util.List;
6
7 public class Matematica {
8
9     public int add(int a, int b) {
10         return a + b;
11     }
12
13     public int subtract(int a, int b) {
14         return a - b;
15     }
16
17     public int multiply(int a, int b) {
18         return a * b;
19     }
20
21     public int divide(int a, int b) {
22         return a / b;
23     }
24
25     public int sum(List<Integer> numbers) {
26         int sum = 0;
27         for (int number : numbers) {
28             sum += number;
29         }
30         return sum;
31     }
32
33     public boolean isEven(int number) {
34         return number % 2 == 0;
35     }
36
37     public boolean isPrime(int number) {
```

My test cases:

Name	Date modified	Type	Size
AdvancedOperationsTest.class	1/15/2026 2:50 AM	CLASS File	2 KB
AdvancedOperationsTest.java	1/15/2026 2:46 AM	JAVA File	1 KB
BasicOperationsTest.class	1/15/2026 2:50 AM	CLASS File	1 KB
BasicOperationsTest.java	1/15/2026 2:46 AM	JAVA File	1 KB
hamcrest.jar	1/15/2026 2:35 AM	JAR File	44 KB
junit.jar	1/15/2026 2:31 AM	JAR File	376 KB
Matematica.class	1/15/2026 2:50 AM	CLASS File	2 KB
Matematica.java	1/15/2026 2:46 AM	JAVA File	2 KB
MatematicaTests.class	1/15/2026 2:40 AM	CLASS File	2 KB
StringOperationsTest.class	1/15/2026 2:50 AM	CLASS File	1 KB
StringOperationsTest.java	1/15/2026 2:46 AM	JAVA File	1 KB

And a **run\_tests.bat** file to check tests work.

```
run_tests.bat ✘ x |  
1  @echo off  
2  if "%~1"==""  
3      echo Usage: run_tests.bat [TestClassName]  
4      echo Example: run_tests.bat BasicOperationsTest  
5      exit /b 1  
6  )  
7  
8  echo Compiling...  
9  javac -cp "SQM_HW\junit.jar;SQM_HW\hamcrest.jar;." SQM_HW\*.java  
10  if %errorlevel% neq 0 exit /b %errorlevel%  
11  
12  echo Running Tests: %1  
13  java -cp "SQM_HW\junit.jar;SQM_HW\hamcrest.jar;." org.junit.runner.JUnitCore SQM_HW.%1  
14  pause
```

Proof tests work the classic way:

```
Microsoft Windows [Version 10.0.22631.6199]
(c) Microsoft Corporation. All rights reserved.

A:\Programme2>run_tests.bat BasicOperationsTest
Compiling...
Running Tests: BasicOperationsTest
JUnit version 4.13.2
....
Time: 0.007

OK (4 tests)

Press any key to continue . . .

A:\Programme2>run_tests.bat AdvancedOperationsTest
Compiling...
Running Tests: AdvancedOperationsTest
JUnit version 4.13.2
....
Time: 0.009

OK (4 tests)

Press any key to continue . . .

A:\Programme2>run_tests.bat StringOperationsTest
Compiling...
Running Tests: StringOperationsTest
JUnit version 4.13.2
..
Time: 0.006

OK (2 tests)

Press any key to continue . . .

A:\Programme2>
```

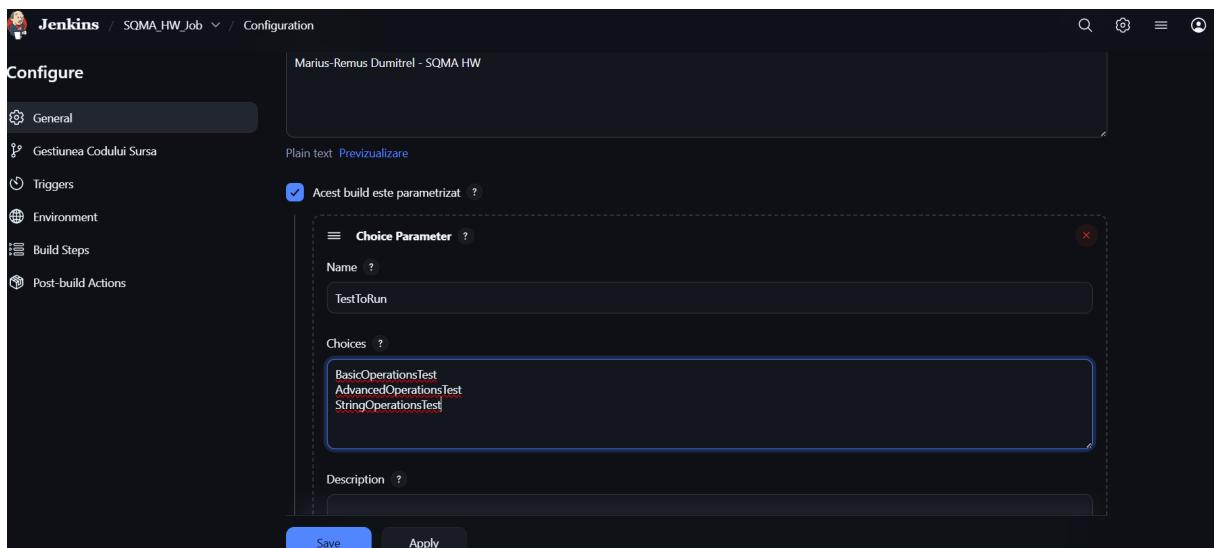
The Git commit and origin main:

```
A:\Programe2>git branch -M main
A:\Programe2>git remote add origin https://github.com/Vincitur/SQMA_Dumitrel_Marius-Remus.git
A:\Programe2>git push -u origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 748 bytes | 748.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Vincitur/SQMA_Dumitrel_Marius-Remus.git
  f3c5068..c5f439b main -> main
branch 'main' set up to track 'origin/main'.
```

### Jenkins Job Setup:

New Item -> SMQA\_HW\_Job -> Freestyle project

- Configure Parameters:  
Check **This project is parameterized.**
- Click **Add Parameter** -> **Choice Parameter**.
- **Name:** TestToRun



## Select Git in Source Code Management:

The screenshot shows the Jenkins configuration interface for a job named "SQMA\_HW\_Job". Under the "Source Code Management" section, the "Git" option is selected. The "Repository URL" field contains the URL "https://github.com/Vincitur/SQMA\_Dumitrel\_Marius\_Renrus". The "Branch Specifier" field is set to "Branch Specifier (blank for 'any')". The "Configure" sidebar on the left lists "General", "Gestiunea Codului Sursa" (which is currently selected), "Triggers", "Environment", "Build Steps" (which is also selected), and "Post-build Actions".

## Build steps: run\_tests.bat in our Windows batch command

The screenshot shows the Jenkins configuration interface for the same job. Under the "Build Steps" section, a "Execute Windows batch command" step is defined with the command "run\_tests.bat %testToRun%". The "Configure" sidebar on the left lists "General", "Gestiunea Codului Sursa", "Triggers", "Environment", "Build Steps" (selected), and "Post-build Actions".

## After we Build with Parameters:

The screenshot shows the Jenkins project page for "Project SQMA HW Job". It displays a "Build with Parameters" step. The "TestToRun" parameter is set to "BasicOperationsTest". At the bottom, there are "Build" and "Cancel" buttons. The sidebar on the left includes links for "Stare", "Schimbări", "Spațiu de lucru", "Build with Parameters" (selected), "Configurează", "Sterge Proiect", "Job Config History", and "Rename".

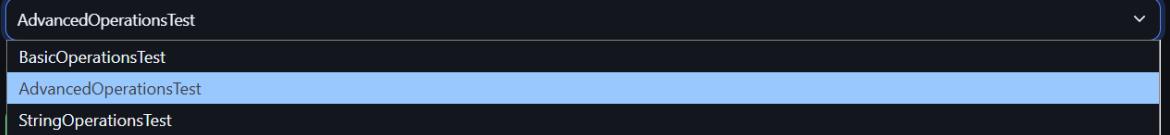
Thus the **Choice parameter** helps decide which tests to run.

**Project SQMA\_HW\_Job**

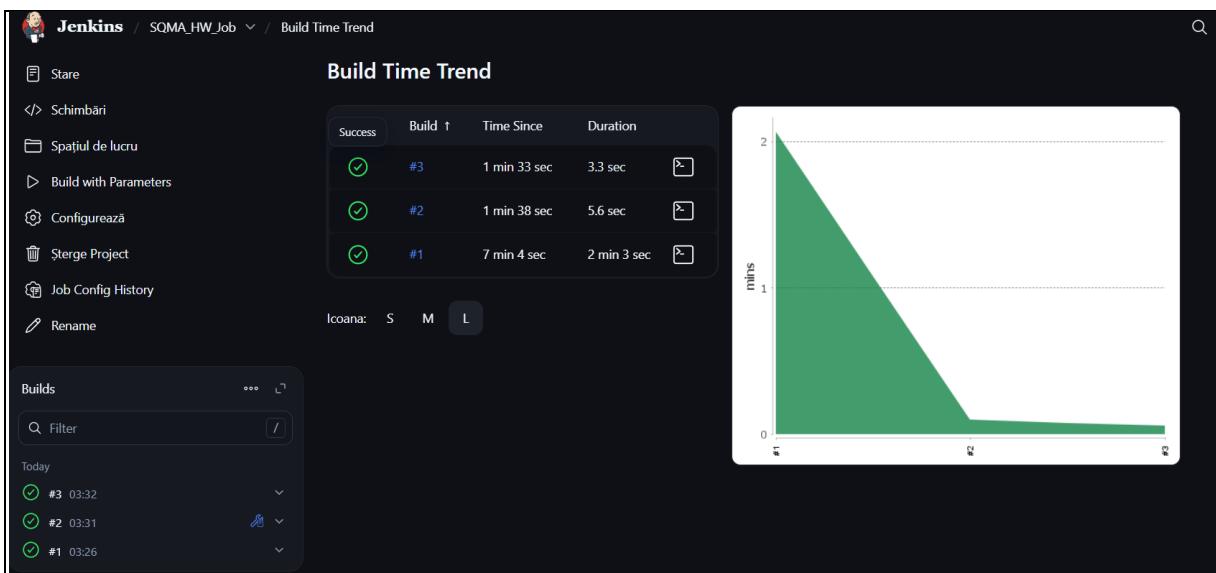
This build requires parameters:

TestToRun  
Select which test suite to run.

AdvancedOperationsTest  
BasicOperationsTest  
AdvancedOperationsTest  
StringOperationsTest



All builds were successful.



Console example (for build 3):

Started by user [Marius-Remus Dumitrel](#)

Running as SYSTEM

Building in workspace A:\Programe2\Jenkins Code\workspace\SQMA\_HW\_Job

The recommended git tool is: NONE

No credentials specified

> git.exe rev-parse --resolve-git-dir A:\Programe2\Jenkins

Code\workspace\SQMA\_HW\_Job\.git # timeout=10

Fetching changes from the remote Git repository

> git.exe config remote.origin.url

[https://github.com/Vincitur/SQMA\\_Dumitrel\\_Marius-Remus](https://github.com/Vincitur/SQMA_Dumitrel_Marius-Remus) # timeout=10

Fetching upstream changes from

[https://github.com/Vincitur/SQMA\\_Dumitrel\\_Marius-Remus](https://github.com/Vincitur/SQMA_Dumitrel_Marius-Remus)

> git.exe --version # timeout=10

> git --version # 'git version 2.45.1.windows.1'

> git.exe fetch --tags --force --progress --

[https://github.com/Vincitur/SQMA\\_Dumitrel\\_Marius-Remus](https://github.com/Vincitur/SQMA_Dumitrel_Marius-Remus)

+refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10

Checking out Revision c5f439b76ef53cb95cfbda541768ad8a2e2dd92c

(refs/remotes/origin/main)

> git.exe config core.sparsecheckout # timeout=10

```
> git.exe checkout -f c5f439b76ef53cb95cfbda541768ad8a2e2dd92c # timeout=10
Commit message: "Initial assignment submission"
> git.exe rev-list --no-walk c5f439b76ef53cb95cfbda541768ad8a2e2dd92c #
timeout=10
[SQMA_HW_Job] $ cmd /c call
C:\Users\Marius\AppData\Local\Temp\jenkins15143727660404099321.bat

A:\Programme2\Jenkins_Code\workspace\SQMA_HW_Job>run_tests.bat
StringOperationsTest
Compiling...
Running Tests: StringOperationsTest
JUnit version 4.13.2
..
Time: 0.007

OK (2 tests)

Press any key to continue . . .
Finished: SUCCESS
```

## Task 2 – Pipeline Job

### Setup:

First we should create a **Jenkinsfile** with our Declarative Pipeline, you will find the file in the Github repository, so I can keep this document clean.

[https://github.com/Vincitur/SQMA\\_Dumitrel\\_Marius-Remus/blob/main/Jenkinsfile](https://github.com/Vincitur/SQMA_Dumitrel_Marius-Remus/blob/main/Jenkinsfile)

```

A:\Programe2>git add .

A:\Programe2>git commit -m "Add Jenkinsfile"
[main 4ff91e2] Add Jenkinsfile
 19 files changed, 46 insertions(+), 1318 deletions(-)
 delete mode 100644 Jenkins Code/jobs/HelloRemus/builds/1/build.xml
 delete mode 100644 Jenkins Code/jobs/HelloRemus/builds/1/changelog.xml
 delete mode 100644 Jenkins Code/jobs/HelloRemus/builds/1/log
 delete mode 100644 Jenkins Code/jobs/HelloRemus/builds/permalinks
 delete mode 100644 Jenkins Code/jobs/HelloRemus/config.xml
 delete mode 100644 Jenkins Code/jobs/HelloRemus/nextBuildNumber
 delete mode 100644 Jenkins/Jenkins.war
 delete mode 100644 Jenkins/Update-JenkinsVersion.ps1
 delete mode 100644 Jenkins/jenkins.err.log
 delete mode 100644 Jenkins/jenkins.exe
 delete mode 100644 Jenkins/jenkins.exe.config
 delete mode 100644 Jenkins/jenkins.msi
 delete mode 100644 Jenkins/jenkins.out.log
 delete mode 100644 Jenkins/jenkins.wrapper.log
 delete mode 100644 Jenkins/jenkins.xml
 create mode 100644 Jenkinsfile
 create mode 100644 ~WRL3083.tmp

A:\Programe2>git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 610.77 KiB | 23.49 MiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/Vinciturn/SQMA_Dumitrel_Marius-Remus.git
 c5f439b..4ff91e2  main -> main

```

Now we need a new Jenkins job: **SQMA\_HW\_Pipeline**

### New Item

Enter an item name

Select an item type

 **Freestyle project**  
 Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

 **Pipeline**  
 Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **Multi-configuration project**

## Configure Pipeline:

The screenshot shows the Jenkins Pipeline configuration page for the 'SQMA\_HW\_Pipeline' job. The left sidebar has tabs for General, Triggers, Pipeline (which is selected), and Advanced. The main area is titled 'Configure' and 'Definition' is set to 'Pipeline script from SCM'. Under 'SCM', 'Git' is selected. In the 'Repositories' section, there is one repository defined with the URL 'https://github.com/Vincitur/SQMA\_Dumitrel\_Marius-Remus'. Below this, the 'Branch Specifier' field contains '/main'. Other settings include '(Auto)' for the Repository browser and 'Jenkinsfile' for the Script Path. A checked checkbox for 'Lightweight checkout' is also present. At the bottom are 'Save' and 'Apply' buttons.

## Run Pipeline

Once I clicked Build Now, the Pipeline ran all tests automatically:

SQMA\_HW\_Pipeline

Task 2: Jenkins Pipeline

Linkuri permanente

Builds

Today

#1 03:58

The build was sucessful:

Stages

15 ianuarie 2026

#1 03:58 - 3 m, 29 s

○ - ✓ - ✓ - ✓ - ✓ - ✓ - ○

The Pipeline gave the following result:

```
Started by user Marius-Remus Dumitrel
Obtained Jenkinsfile from git
https://github.com/Vincitur/SQMA\_Dumitrel\_Marius-Remus
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in A:\\Programme2\\Jenkins Code\\workspace\\SQMA_HW_Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
```

```
The recommended git tool is: git.exe
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Vinciturn/SQMA\_Dumitrel\_Marius-Remus
> git.exe init A:\Programme2\Jenkins Code\workspace\SQMA_HW_Pipeline #
timeout=10
Fetching upstream changes from
https://github.com/Vinciturn/SQMA\_Dumitrel\_Marius-Remus
> git.exe --version # timeout=10
> git --version # 'git version 2.45.1.windows.1'
> git.exe fetch --tags --force --progress --
https://github.com/Vinciturn/SQMA\_Dumitrel\_Marius-Remus
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url
https://github.com/Vinciturn/SQMA\_Dumitrel\_Marius-Remus # timeout=10
> git.exe config --add remote.origin.fetch
+refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 4ff91e26773f6d51a3d12be9ab9794aba75f59f4
(refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 4ff91e26773f6d51a3d12be9ab9794aba75f59f4 # timeout=10
Commit message: "Add Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Checkout)
[Pipeline] checkout
The recommended git tool is: git.exe
No credentials specified
> git.exe rev-parse --resolve-git-dir A:\Programme2\Jenkins
Code\workspace\SQMA_HW_Pipeline\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url
https://github.com/Vinciturn/SQMA\_Dumitrel\_Marius-Remus # timeout=10
Fetching upstream changes from
https://github.com/Vinciturn/SQMA\_Dumitrel\_Marius-Remus
> git.exe --version # timeout=10
> git --version # 'git version 2.45.1.windows.1'
> git.exe fetch --tags --force --progress --
https://github.com/Vinciturn/SQMA\_Dumitrel\_Marius-Remus
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 4ff91e26773f6d51a3d12be9ab9794aba75f59f4
(refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 4ff91e26773f6d51a3d12be9ab9794aba75f59f4 # timeout=10
Commit message: "Add Jenkinsfile"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test Basic Operations)
```

```
[Pipeline] bat

A:\Programme2\Jenkins Code\workspace\SQMA_HW_Pipeline>run_tests.bat
BasicOperationsTest
Compiling...
Running Tests: BasicOperationsTest
JUnit version 4.13.2
....
Time: 0.006

OK (4 tests)

Press any key to continue . . .
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test Advanced Operations)
[Pipeline] bat

A:\Programme2\Jenkins Code\workspace\SQMA_HW_Pipeline>run_tests.bat
AdvancedOperationsTest
Compiling...
Running Tests: AdvancedOperationsTest
JUnit version 4.13.2
....
Time: 0.006

OK (4 tests)

Press any key to continue . . .
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test String Operations)
[Pipeline] bat

A:\Programme2\Jenkins Code\workspace\SQMA_HW_Pipeline>run_tests.bat
StringOperationsTest
Compiling...
Running Tests: StringOperationsTest
JUnit version 4.13.2
..
Time: 0.008

OK (2 tests)

Press any key to continue . . .
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline execution finished.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
```

```
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The screenshot shows the Jenkins Pipeline Overview for the SQMA\_HW\_Pipeline #1. The pipeline stages are represented by a horizontal timeline: Start, Checkout SCM (green checkmark), Checkout (green checkmark), Test Basic Operations (green checkmark), Test Advanced... (green checkmark), Test String Operations (green checkmark), Post Actions (green checkmark), and End. Below the timeline, a summary of the execution details is provided:

- Manually run by Marius-Remus Dumitrel
- Pornit acum 7 min 14 sec
- Queued 6 ms
- Took 3 min 29 sec
- Post Actions: 92 ms
- Pipeline execution finished: 48 ms
- Total duration: 3 m, 45 s
- Jenkins version: Jenkins

## The End:

The screenshot shows the Jenkins dashboard for the SQMA\_HW\_Pipeline job. The build history section displays the following information:

S	W	Name	Ultimul Build efectuat cu succes	Ultimul Esec	Durata ultimului build
✓	☀️	SQMA_HW_Job	34 min #3	Nu este aplicabil	3.3 sec
✓	☀️	SQMA_HW_Pipeline	8 min 3 sec #1	Nu este aplicabil	3 min 29 sec

Other sections visible include 'Listă de așteptare' (Empty) and 'Starea builder-ului Executant' (0 of 2 executors busy). The bottom right corner shows the REST API and Jenkins version (2.528.3).