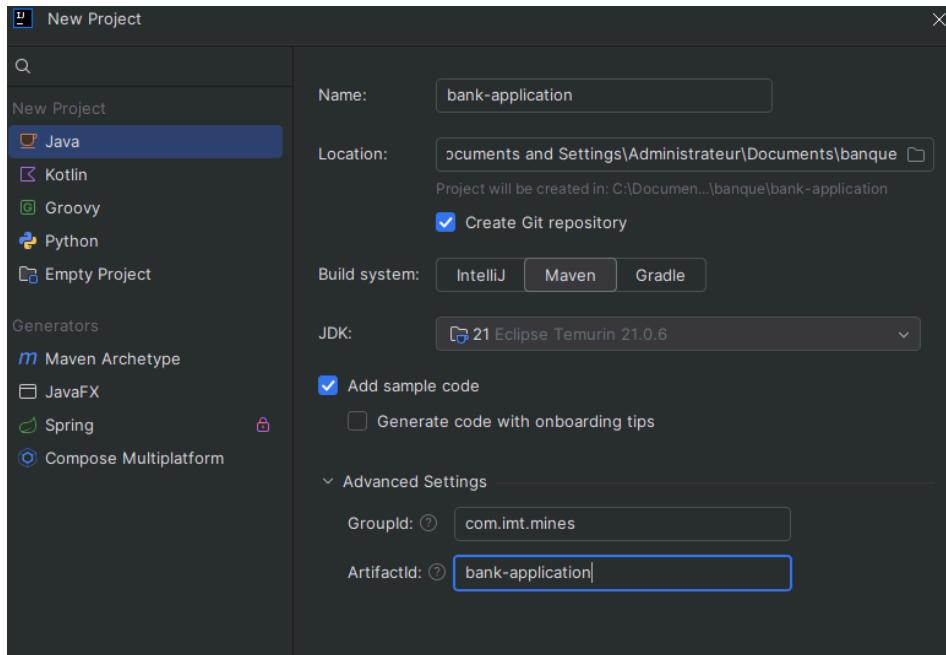
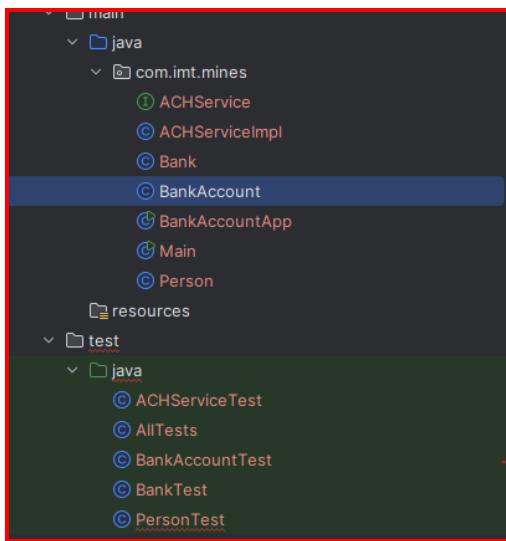


## Exercise 6 (Repo 2): Turn *BankApplication* into a Maven project



```
m pom.xml (bank-application) x  ⓘ ACHService.java  ⓘ ACHServiceImpl.java  ⓘ Bank.java
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"
5      modelVersion="4.0.0">
6
7      <groupId>com.imt.mines</groupId>
8      <artifactId>bank-application</artifactId>
9      <version>1.0-SNAPSHOT</version>
10
11     <properties>
12         <maven.compiler.source>17</maven.compiler.source>
13         <maven.compiler.target>17</maven.compiler.target>
14         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15     </properties>
16
17 </project>
```



```
PS C:\Users\Administrateur\Documents\banque\bank-application> mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.imt.mines:bank-application >-----
[INFO] Building bank-application 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ bank-application ---
[INFO] Copying 0 resource from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ bank-application ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 7 source files with javac [debug target 17] to target\classes
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  2.447 s
[INFO] Finished at: 2025-11-19T15:31:34+01:00
[INFO]
[INFO] -----
```

## Exercise 7 (Repo 2):

```
<dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.10.2</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

**mvn clean :**

```
PS C:\Users\Administrateur\Documents\banque\bank-application> mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.imt.mines:bank-application >-----
[INFO] Building bank-application 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ bank-application ---
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.3.4/maven-shared-utils-3.3.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-shared-utils/3.3.4/maven-shared-utils-3.3.4.jar (153 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/commons-io/commons-io/2.6/commons-io-2.6.jar
Downloaded from central: https://repo.maven.apache.org/maven2/commons-io/commons-io/2.6/commons-io-2.6.jar (215 kB at 1.6 MB/s)
[INFO] Deleting C:\Users\Administrateur\Documents\banque\bank-application\target
[INFO] -----
[INFO] BUILD SUCCESS
```

#### maven test :

```
PS C:\Users\Administrateur\Documents\banque\bank-application> mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.imt.mines:bank-application >-----
[INFO] Building bank-application 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ bank-application ---
[INFO] Copying 0 resource from src\main\resources to target\classes
Downloaded from central: https://repo.maven.apache.org/ (3 at 4.5 MB/s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  3.902 s
[INFO] Finished at: 2025-11-19T16:11:45+01:00
```

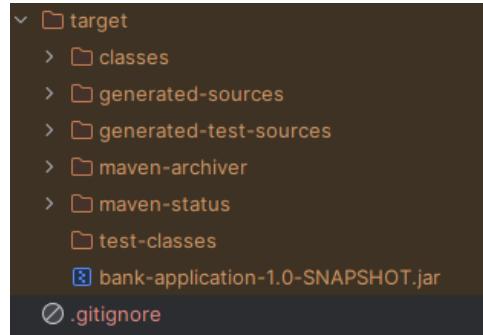
#### mvn package :

```
PS C:\Users\Administrateur\Documents\banque\bank-application> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.imt.mines:bank-application >-----
[INFO] Building bank-application 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ bank-application ---
[INFO] Copying 0 resource from src\main\resources to target\classes
Downloaded from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.9/xz-1.9.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/commons/commons-compress/1.22/commons-compress-1.22.jar
Downloaded from central: https://repo.maven.apache.org/maven2/com/github/luben/zstd-jni/1.4.3/zstd-jni-1.4.3.jar
[INFO] Building jar: C:\Users\Administrateur\Documents\banque\bank-application\target\bank-application-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  5.397 s
[INFO] Finished at: 2025-11-19T16:13:01+01:00
```

- Which **phases** of the Maven lifecycle are executed?

the phases that are executed are : **validate , compile , test compile , test , package**

- What **new files or folders** appear in target /?



maven verify :

```
PS C:\Users\Administrateur\Documents\banque\bank-application> mvn verify
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.imt.mines:bank-application >-----
[INFO] Building bank-application 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
```

```
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time:  3.087 s
[INFO] Finished at: 2025-11-19T16:23:34+01:00
[INFO] -----
```

mvn verify goes further than both test and package

verify ensures the project is *ready for release* by running extra validation steps after packaging, while test only runs tests and package only builds the JAR.

## Exercise 8 (Repo 2): Write unit tests for the bank domain

chosen class : bankaccount

methods : depositMoney(double amount)

withdrawMoney(double amount)

```

import com.imt.mines.BankAccount;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

class BankAccountTest {

    @Test
    void testDepositMoney() {
        BankAccount acc = new BankAccount();
        acc.depositMoney(depositAmount: 100);
        assertEquals(expected: 100, acc.getBalance());
    }

    @Test
    void testWithdrawTooMuch() {
        BankAccount acc = new BankAccount();
        acc.depositMoney(depositAmount: 50);

        boolean result = acc.withdrawMoney(withdrawAmount: 100);

        assertFalse(result);
        assertEquals(expected: 50, acc.getBalance());
    }
}

```

```

PS C:\Users\Administrateur\Documents\banque\bank-application> mvn test
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.imt.mines:bank-application >-----
[INFO] Building bank-application 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ bank-application ---

[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running BankAccountTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.085 s -- in BankAccountTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 6.678 s

```

## Exercise 9 (Repo 2): Increase coverage with JaCoCo

## Ajout du plugin jacoco dans pom.xml

```
<build>
    <plugins>

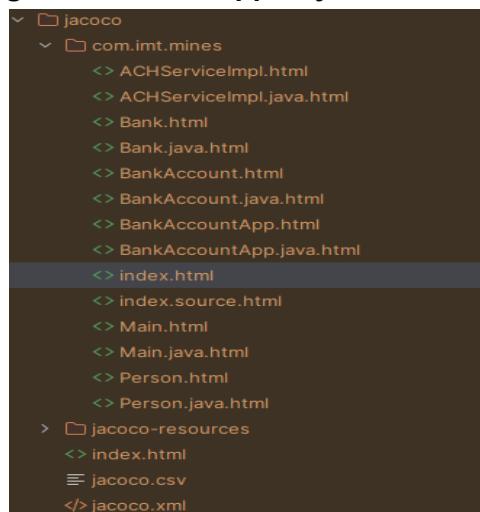
        <!-- Plugin Jacoco -->
        <plugin>
            <groupId>org.jacoco</groupId>
            <artifactId>jacoco-maven-plugin</artifactId>
            <version>${jacoco.version}</version>
            <executions>

                <execution>
                    <goals>
                        <goal>prepare-agent</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
```

## Mvn clean test

```
[INFO] --- jacoco:0.8.10:report (report) @ bank-application ---
[INFO] Loading execution data file C:\Users\Administrateur\Documents\bank-application\target\surefire-reports\jacoco.exec
[INFO] Analyzed bundle 'bank-application' with 6 classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 11.258 s
```

## génération du rapport jacoco



## the index.html

bank-application > com.imt.mines

### com.imt.mines

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cqty	Missed	Lines	Missed	Methods	Missed	Classes
BankAccountApp	██████████	0 %	██████████	0 %	39	39	134	134	2	2	1	1
Bank	██████████	0 %	██████	0 %	32	32	93	93	14	14	1	1
BankAccount	██████████	20 %	████	20 %	26	32	78	100	16	22	0	1
Person	██████████	0 %	████	0 %	34	34	80	80	23	23	1	1
Main	██████████	0 %	n/a	n/a	2	2	3	3	2	2	1	1
ACHServiceImpl	██████████	0 %	n/a	n/a	3	3	3	3	3	3	1	1
Total	1334 of 1409	5 %	148 of 152	2 %	136	142	391	413	60	66	5	6

Alors maintenant je vais ajouter 2 autres tests pour la class bankaccount pour verifier si mon coverage va augmenter dans index.html

```

    @Test
    void testDepositNegativeAmount() {
        BankAccount acc = new BankAccount();
        acc.depositMoney(depositAmount: 100); // soldé = 100

        acc.depositMoney(depositAmount: -50); // dépôt négatif → ignoré

        assertEquals(expected: 100, acc.getBalance()); // soldé reste inchangé
    }

    @Test
    void testCumulativeWithdrawExceedsLimit() {
        BankAccount acc = new BankAccount();
        acc.depositMoney(depositAmount: 500);
        acc.setWithdrawLimit(100); // limite totale

        boolean first = acc.withdrawMoney(withdrawAmount: 60);
        boolean second = acc.withdrawMoney(withdrawAmount: 50); // doit échouer (60 + 50 > 100)

        assertTrue(first);
        assertFalse(second);
        assertEquals(expected: 500 - 60, acc.getBalance()); // soldé reste après 1er retrait
    }

```

après l'ajout des tests

bank-application > com.imt.mines

### com.imt.mines

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cqty	Missed	Lines	Missed	Methods	Missed	Classes
BankAccountApp	0 %	0 %	0 %	0 %	39	39	134	134	2	2	1	1
Bank	0 %	0 %	0 %	0 %	32	32	93	93	14	14	1	1
BankAccount	30 %	70 %	50 %	50 %	21	32	72	100	15	22	0	1
Person	0 %	0 %	0 %	0 %	34	34	80	80	23	23	1	1
Main	0 %	0 %	n/a	n/a	2	2	3	3	2	2	1	1
ACHServiceimpl	0 %	0 %	n/a	n/a	3	3	3	3	3	3	1	1
Total	1 298 of 1 409	7 %	142 of 152	6 %	131	142	385	413	59	66	5	6

## Exercise 10 (Repo 2): Extend a CI workflow (GitHub Actions)

```

25      # Run Maven tests
26      - name: Run tests
27          run: mvn clean test
28
29      # Build Maven package
30      - name: Build package
31          run: mvn package
32
33      # Upload target/ folder as artifact
34      - name: Upload build artifact
35          uses: actions/upload-artifact@v4
36          with:
37              name: java-build
38              path: target/

```

Triggered via push 1 minute ago

Status Success Total duration 35s Artifacts 1

maven.yml  
on: push

build 31s

Artifacts  
Produced during runtime

Name	Size	Digest
java-build	106 KB	sha256:330d02f85ff4a724b9bc459a814e3b7996... <a href="#">copy</a>

## Exercise 11 (Repo 2): Cucumber for BankAccount

- Add Cucumber Dependencies:

```
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-java</artifactId>
    <version>7.15.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>io.cucumber</groupId>
    <artifactId>cucumber-junit</artifactId>
    <version>7.15.0</version>
    <scope>test</scope>
</dependency>
</dependencies>
```

- Create a Feature File: Create the folder (if it does not exist):  
`src/test/resources/features`. Inside it, create the file:  
`src/test/resources/features/bank_account_basic.feature`

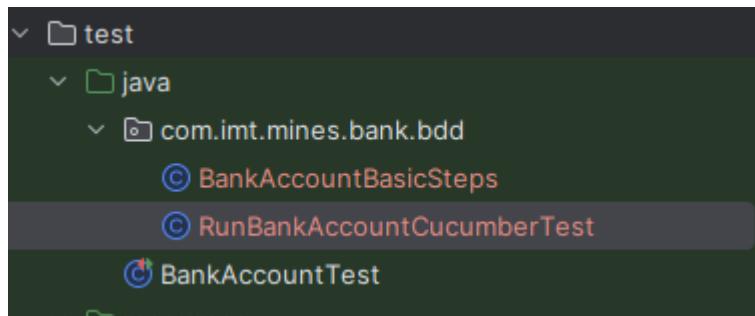
The screenshot shows a file tree on the left and a code editor on the right. The file tree includes a Person class, a resources folder, a test folder containing Java files (BankAccountBasicSteps, RunBankAccountCucumberTest, BankAccountTest) and a features folder containing the bank\_account\_basic.feature file. The code editor displays the contents of the feature file:

```

1 Feature: Basic bank account behavior
2 Scenario: A new account has zero balance
3 Given I have a new bank account
4 When I check its balance
5 Then the balance should be 0
6

```

- **Create the Cucumber Runner:** Create a package for BDD tests under `src/test/java`: `src/test/java/com/imt/mines/bank/bdd`. In that package, create the class: `RunBankAccountCucumberTest.java` (You do not need to modify this class)



- **Run the Tests**
  - From the project root (where `pom.xml` is): `mvn test`

The screenshot shows the Eclipse IDE's 'Run' view. The 'Feature' dropdown is set to 'bank\_account\_basic'. The results pane shows 'Test Results' with 1 scenario passed and 3 steps passed. The output pane displays the command used ('java -jar'), the start time ('Testing started at 21:30 ...'), and the summary ('1 Scenarios (1 passed), 3 Steps (3 passed)').

```

Run Feature: bank_account_basic
Test Results 8 ms
Done: Scenarios 1 of 1 (1sec 593 ms)
"C:\Program Files\Eclipse Adoptium\jdk-21.0.6.7-hotspot\bin\java.exe" ...
Testing started at 21:30 ...
1 Scenarios (1 passed)
3 Steps (3 passed)
0m0,393s

Process finished with exit code 0

```

but i dont see that on the output when i run maven test

```
[INFO]
[INFO] --- jacoco:0.8.10:prepare-agent (default) @ bank-application ---
[INFO] argLine set to -javaagent:C:\\Users\\Administrateur\\.m2\\repository\\org\\jacoco\\org.jacoco.agent-0.8.10.jar=C:\\Users\\Administrateur\\Documents\\banque\\bank-application\\target\\jacoco.exec
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ bank-application ---
[INFO] Copying 0 resource from src\\main\\resources to target\\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ bank-application ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ bank-application ---
[INFO] Copying 1 resource from src\\test\\resources to target\\test-classes
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ bank-application ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ bank-application ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
```