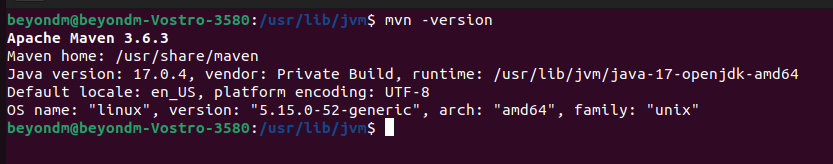
**4. Maven**

1. **Why do we need a build tool like maven?**

Maven is a project management tool based on project object model(POM). Maven can be used to build,dependency and documentation of java projects. Main advantages of the maven are,

* + - * 1. Project building is easy.
        2. Maven helps to add jars and dependencies to the project easily.
        3. Maven provides project information (log document, dependency list, unit test reports etc.)
        4. Maven update a central repository of jars and dependencies.
        5. With the help of Maven we can build any number of projects into output types like the JAR, WAR etc without doing any scripting.
        6. We can easily integrate with version control systems like git.

1. **Install maven**
2. **Display output of maven version**

****

1. **What is the pom.xml file?**

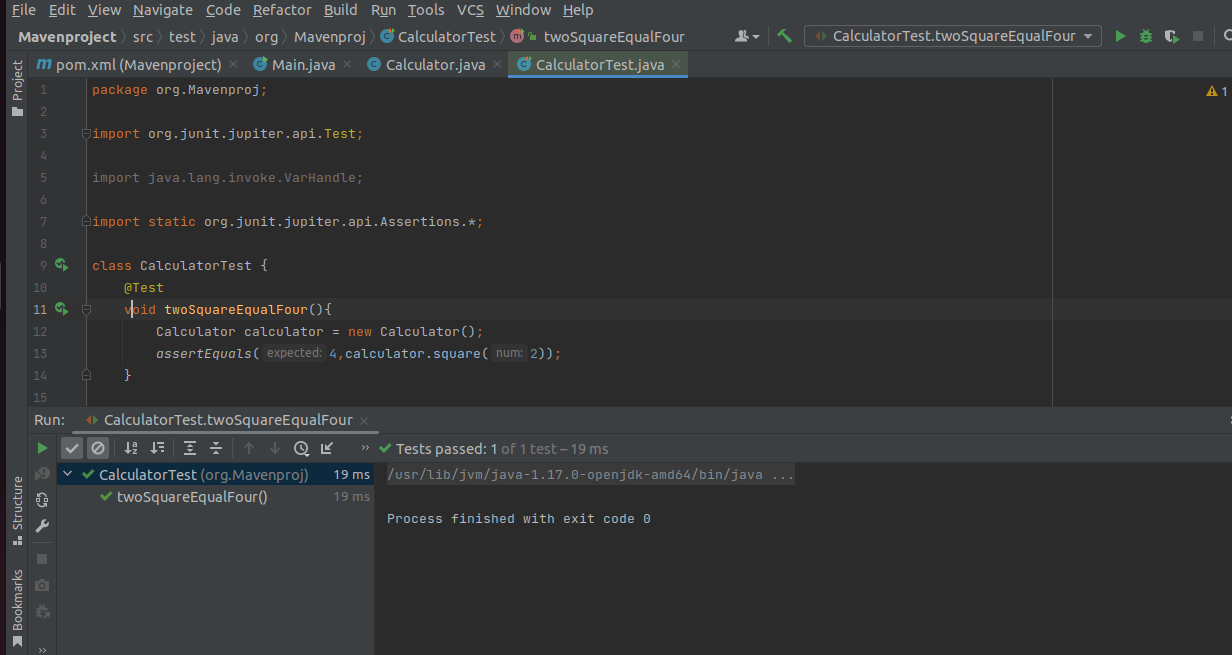
POM is an acronym for Project Object Model. The pom.xml file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc.

1. **Explain these tags found in pom.xml files?**

|  |  |
| --- | --- |
| groupId | It is the sub element of project. It specifies the id for the project group. |
| artifactId | It is the sub element of project. It specifies the id for the artifact (project). An artifact is something that is either produced or used by a project. |
| version | It is the sub element of project. It specifies the version of the artifact under given group. |
| packaging | defines packaging type such as jar, war etc. |
| dependencies | defines dependencies for this project. |
| dependency | defines a dependency. It is used inside dependencies. |
| properties | Maven properties are value placeholders. Their values are accessible anywhere within a POM by using the notation ${X}, where X is the property or they can be used by plugins as default values. |

1. **Create a method which accept an integer as parameter and returns the square of it**
2. **Add junit (v5) dependency**
3. **Add a unit test to test the method**
4. **Run unit tests with maven. What is the command you used?**

Ctrl +Shift+F10



1. Create a Student class with following attributes and add getters/setters

* id: int
* name: String
* age: int
* subjects: List<String>

1. In your main method create and student object and set these values

id: 1

name: “john”

age: 20

subjects: [“Maths”, “English”, “History”]

1. Then print these student values using getters (e.g: s.getName())
2. Add lombok dependency and remove getter/setter methods from Student class
3. **Explain the usage of these commands**

|  |  |
| --- | --- |
| mvn clean | This command I used to clean the maven project by deleting the target directory. |
| mvn install | This command builds the maven project and installs the project files (JAR, WAR, pom.xml, etc) to the local repository. |
| mvn package | This command builds the maven project and packages them into a JAR, WAR, etc. |

1. **Explain 3 types of maven repositories**

A maven repository is a directory of packaged JAR file with pom.xml file. Maven searches for dependencies in the repositories. There are 3 types of maven repository:

* 1. Local Repository - Maven local repository is located in your local system. It is created by the maven when you run any maven command.
  2. Central Repository - Maven central repository is located on the web. It has been created by the apache maven community itself.
  3. Remote Repository - Maven remote repository is located on the web. Most of libraries can be missing from the central repository such as JBoss library etc, so we need to define remote repository in pom.xml file.

Maven searches for the dependencies in the following order: Local repository then Central repository then Remote repository.

1. Add your codes and answer sheet to a directory named “maven-basic-training” and push it to your training github repository.