BATCH : Batch 107

LESSON : Jenkins

DATE :

SUBJECT : Jenkins Day 4

techproeducation

techproeducation

techproeducation

techproeducation

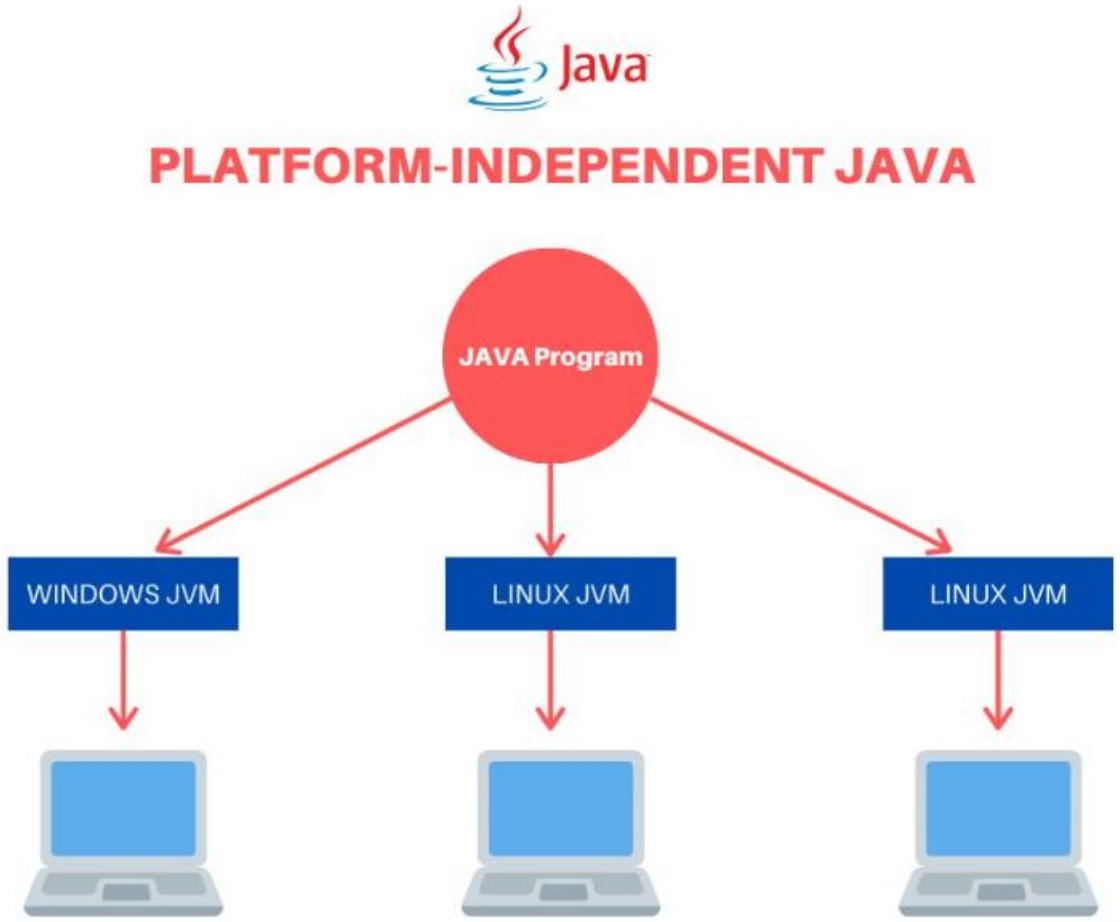
techproedu

Table of Contents

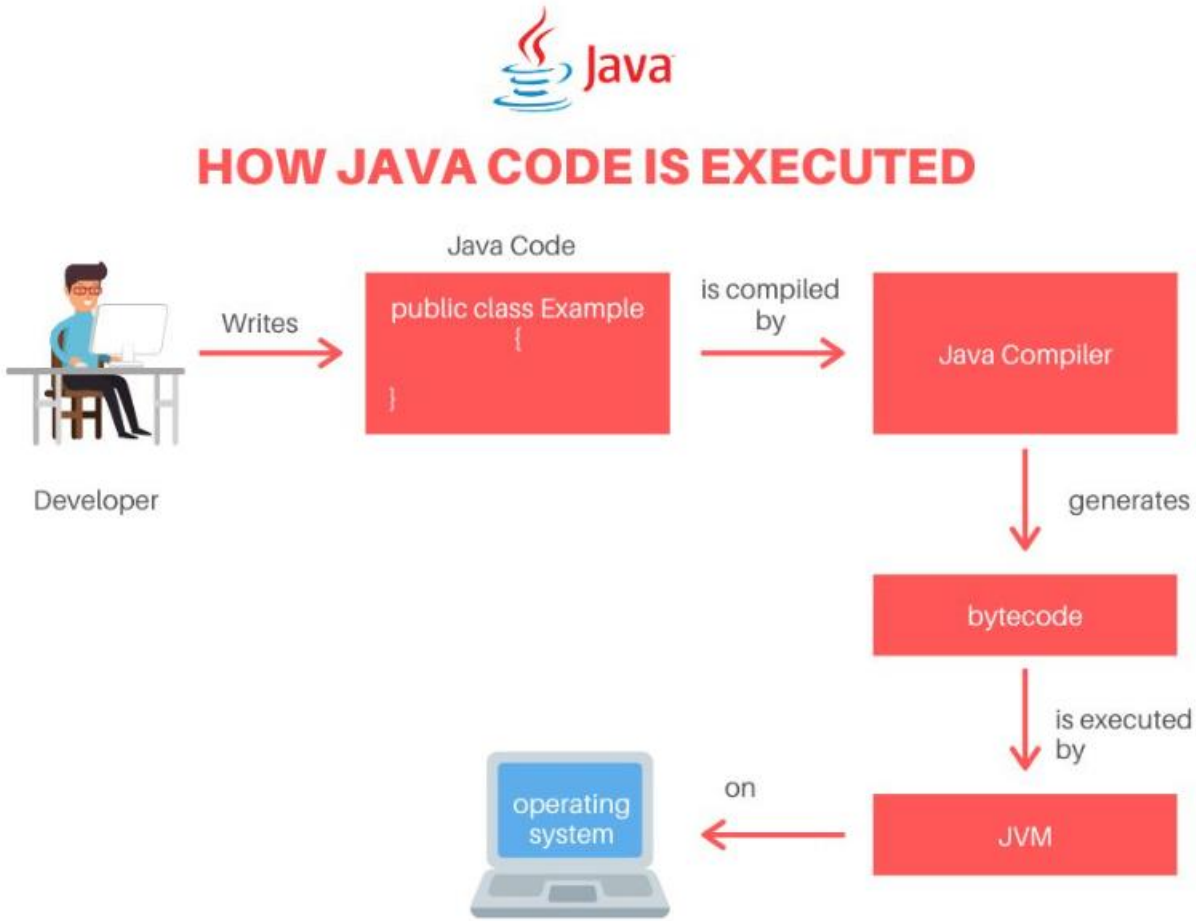
Jenkins Day 3

maven

- Maven began its life in Apache's Jakarta Alexandria Project in 2001.
- Apache Maven helps to,
  - build, multiple projects easily,
  - publish documentation for the projects,
  - accomplish an easy deployment,
  - share JARs across several other projects and
  - help in collaboration with development teams.



In the above diagram you will see that on the top of every operating system there is a Java Virtual Machine which is needed to run the Java program. **JVM** or **Java Virtual Machine** enables the computer to run a java program, since every operating system has its own JVM so it is a platform independent environment that simply converts Java into code that a computer understands.



In the above figure you can clearly see how Java code is compiled and converted to bytecode. Then it is executed through the Java virtual machine in the operating system. In short it is the working of an application made using Java programming language.

- Maven can:
  - manage a software project's builds with various versions,
  - compile source code into binary,
  - download dependencies,
  - add documentation,
  - run tests,
  - package compiled code,
  - deploy artifacts to an application server or a repository.

TECHPROED

Features of Maven

- It's easy to start with Maven.
- variety of options according to your needs.
- It has the same structure across a variety of different projects.
- It's easy to integrate into a developing team when they are working on Maven.
- It has a powerful dependency management tool.

TECHPROED

Features of Maven

- extra features with plugins in Java or scripting languages.
- Maven can give different outputs like a jar, ear, war, or metadata for the same project.
- Maven can generate a website and a PDF with the documentation in the project.
- Maven can integrate with your source control system such as CVS and manages the release of a project.
- run tests, deploy artifacts to an application server or a repository.


```
└──maven-project
    ├──pom.xml
    ├──README.txt
    ├──NOTICE.txt
    ├──LICENSE.txt
    └──src
        ├──main
        │   ├──java
        │   ├──resources
        │   ├──filters
        │   └──webapp
        ├──test
        │   ├──java
        │   ├──resources
        │   └──filters
        ├──it
        ├──site
        └──assembly
```

TECHPROED

POM File

- POM can define
  - the **project dependencies,**
  - **the plugins or goals to be executed,**
  - **the build profiles, and more**

POM File

The POM contains all necessary information about a project, as well as configurations of plugins to be used during the build process. It is the declarative manifestation of the "who", "what", and "where", while the build lifecycle is the "when" and "how".

POM files need at least the project tag and four other inner tags named as modelVersion, groupId, artifactId, and version

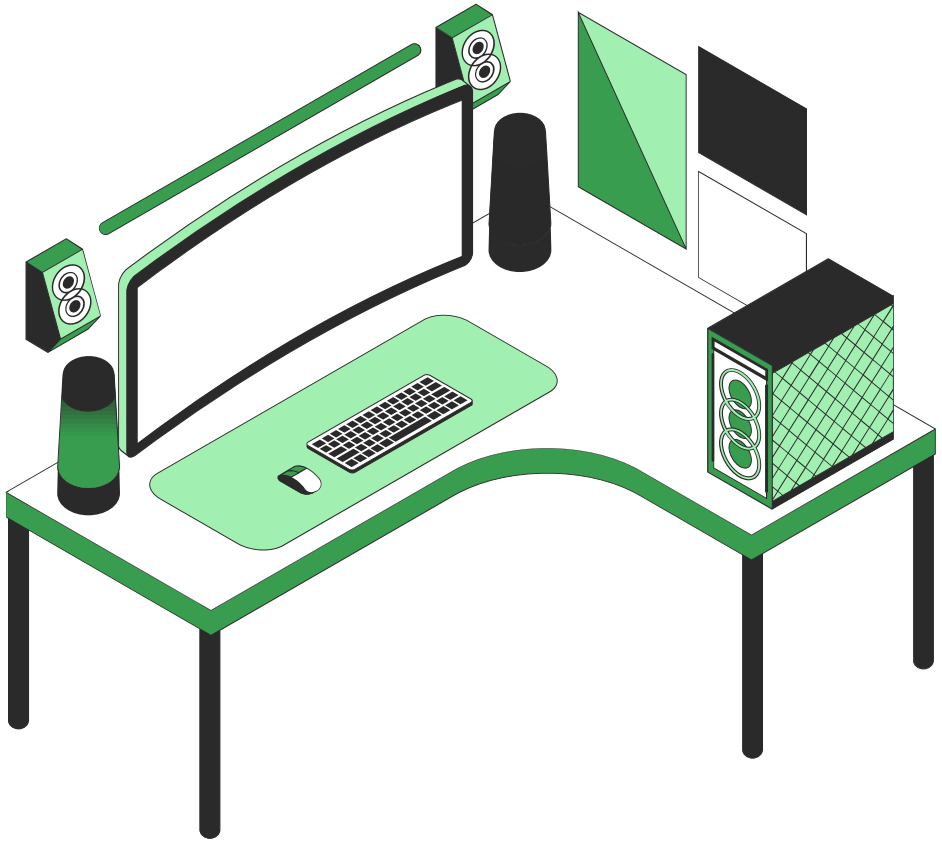
TECHPROED

POM File

POM files need at least the project tag and four other inner tags named as modelVersion, groupId, artifactId, and version

```
1. <project>
2.   <modelVersion>4.0.0</modelVersion>
3.
4.   <groupId>com.mycompany.app</groupId>
5.   <artifactId>my-app</artifactId>
6.   <version>1</version>
7. </project>
```

TECHPROED



Do you have any questions?

Send it to us! We hope you learned something new.

TECHPROED