## **JDK VS JRE VS JVM**

* JDK Environment to Develop & Run/Execute the code
* JRE Environment to Run/Execute the code
* JVM : Interpreter line by line && exeute the code

## **What is JAR ??**

* JAR file stands for Java Archive with .jar file extension
* JAR file allows Java Runtime Environment (JRE) to deploy an entire application including the classes and related resources in a single request.
* A JAR file is a file with Java classes, associated metadata, and resources such as text and images aggregated into one file.

**To execute a JAR file, simple JDK is enough**

A JAR file has all the components required to make a self-contained executable Java application. It consists of Java source codes, XML based configuration data, manifest file, JSON based data files, etc. In other words, a JAR file is a combination of all these files as a single compressed file. The compressed file reduces the size of the application. It is also easier to send that file through the network to various systems or platforms.

To develop Java-based applications, the programmer should install Java Development Kit (JDK) into the machine. Each JDK has JAR utility to support JAR files. It allows creating new JAR files with a manifest file and extracting all the content of a JAR file onto the file system. Furthermore, this utility is also helpful for updating the existing JAR files.

## **What is WAR ??**

* WAR file stands for Web Application Resource or Web Application Archive with .war file extension.
* WAR file allows testing and deploying web applications easily
* To execute a WAR file, a Web server or Web container is required, for example, Tomcat or Weblogic or Websphere
* A WAR file is a file that is used to distribute a collection of JAR files, JSP, Servlet, XML files, static web pages like HTML and other resources that constitute a web application

A WAR file contains the files of a web project. It can have servlet, JSP, XML, HTML, CSS and JavaScript files. These files can be deployed on servlet/JSP container. The WAR files are inside the WEB-INF folder of the project. As a WAR file combines all the files into a single unit, it takes less time to exchange a file from the client to server.

It is possible to deploy a WAR file by using the server control panel. Another method is by manually locating a WAR file in a specific folder of the server. In order to deploy a WAR file in a server such as Tomcat manually, the programmer can go to the webapps directory of Tomcat and paste the WAR file in that directory. Otherwise, the server extracts the WAR file internally, when executing the web project.

## **What is EAR ??**

* EAR stands for Enterprise Application Archive with .ear file extension
* EAR file allows deploying different modules onto an application server
* EAR file is deployed in an **Application server**
* An EAR file is a standard JAR file that represents the modules of the application, and a metadata directory called META-INT which contains one or more deployment descriptors.

An EAR file is a Java EE file. It packs one or more modules into one archive. Also, this helps to deploy various modules onto an application server simultaneously and coherently. EAR file consists of deployment descriptors that describe how to deploy the modules. These deployment descriptors are XML files. Moreover, applications such as Ant, Maven, and, Gradle help to build EAR files.

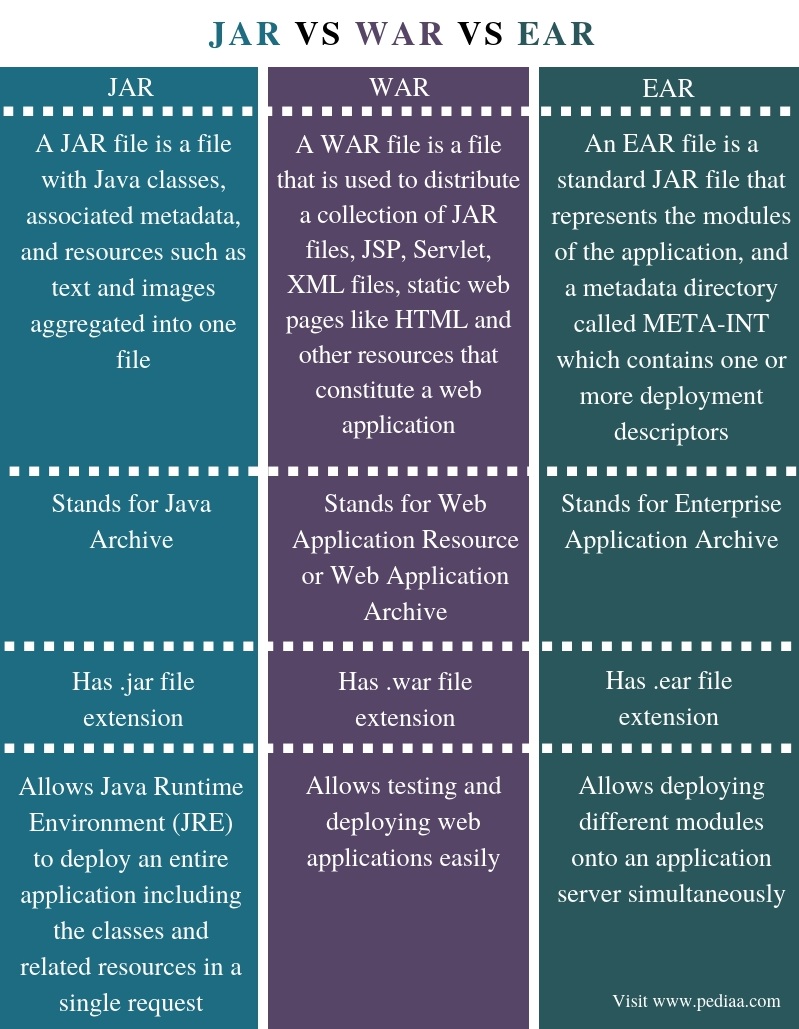
## **JAR VS WAR VS EAR**

* JAR: EJB modules which contain enterprise java beans (class files) and EJB deployment descriptor are packed as JAR files with .jar extension
* WAR: Web modules which contain Servlet class files, JSP Files, supporting files, GIF and HTML files are packaged as a JAR file with .war (web archive) extension
* EAR: All the above files (.jar and .war) are packaged as a JAR file with .ear (enterprise archive) extension and deployed into Application Server.

JAR >>JAVA ARCHIVE :: Group of jar .class files.

WAR >> WEB ARCHIVE :: Its for webapplication (10 jsp 100 servlets html css js xml files) Each war file represents one web application. Only web related technologies like servlets jsp

EAR >> Enterprise Archive



## **Java Compile Process**

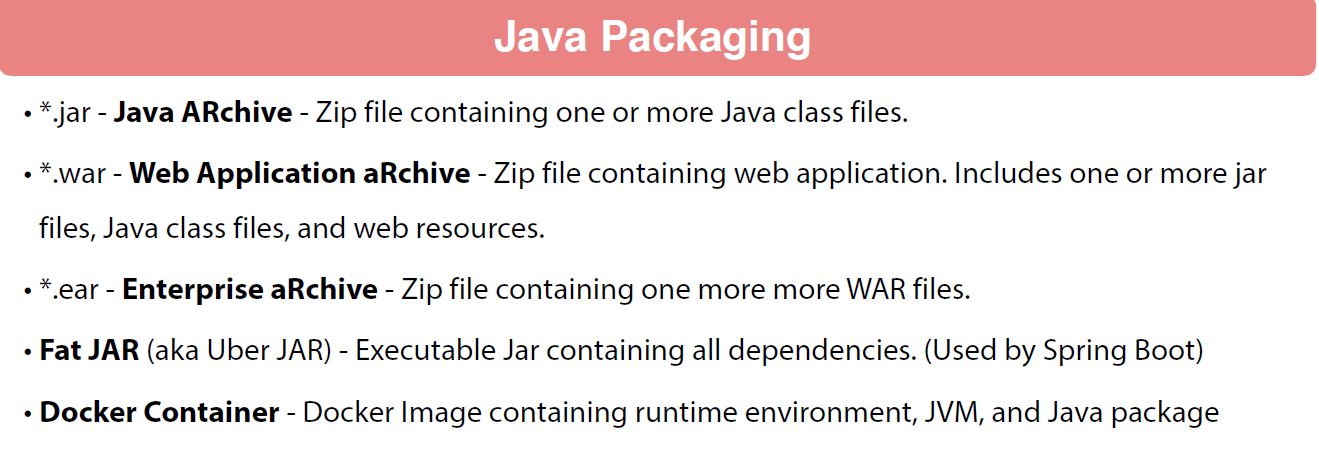
A picture containing diagram

Description automatically generated

## **Packaging Java Applications**

Shape

Description automatically generated



## **Java Deployment**

Graphical user interface, text, application, email

Description automatically generated

## **Java Hello World**

Step 1 : Create java application.  


Step 2 : Complilation

* $ javac HelloWorld.java It creates HelloWorld.class file

Step 3: Execution

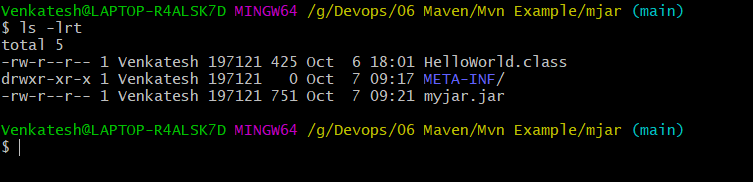
* $ java Helloworld It gives output

## **Creating Java JAR files**

Text

Description automatically generated

**Extracted Jar file**



It consist a MANIFEST.MF file inside META-INF folder, this is part of jar specification. We didn’t specify what to do with it. We could made this an executable jar and specify the main class .