\*\*WAR (Web Application Archive)\*\* и \*\*JAR (Java Archive)\*\* - это архивные форматы, используемые в мире Java для упаковки и распространения приложений и библиотек. Каждый из этих форматов имеет свои особенности и предназначение:

1. \*\*WAR (Web Application Archive)\*\*:

- Формат WAR используется для упаковки веб-приложений, предназначенных для развертывания на веб-серверах.

- WAR-файл содержит в себе веб-компоненты, такие как сервлеты, JSP-страницы, HTML, CSS, JavaScript и другие ресурсы, необходимые для работы веб-приложения.

- WAR-файлы обычно развертываются на серверах приложений, таких как Apache Tomcat, WildFly, и служат для обслуживания веб-приложений.

2. \*\*JAR (Java Archive)\*\*:

- Формат JAR используется для упаковки и распространения Java-библиотек (библиотек классов и ресурсов), а также самостоятельных приложений.

- JAR-файлы содержат Java-байт-код, ресурсы, манифест (manifest), который содержит метаданные о JAR-файле, и могут содержать файлы манифеста.

- JAR-файлы используются для организации и упаковки библиотек и приложений так, чтобы они могли быть легко переносимыми и повторно используемыми на разных платформах и в разных проектах.

Интересный факт: WAR-файлы на самом деле являются специфическими JAR-файлами с определенной структурой каталогов и содержат специфические файлы манифеста, которые указывают на точку входа в веб-приложение.

Таким образом, WAR-файлы предназначены для веб-приложений и их развертывания на серверах, в то время как JAR-файлы используются для упаковки и распространения Java-кода и библиотек, которые могут быть использованы другими Java-приложениями.

my-java-project/

├── src/

│ ├── main/

│ │ ├── java/

│ │ │ ├── [Ваши пакеты и исходные Java-классы]

│ │ ├── resources/

│ │ │ ├── [Файлы конфигурации и ресурсы]

│ │ ├── webapp/

│ │ │ ├── META-INF/

│ │ │ │ ├── MANIFEST.MF

│ │ │ ├── WEB-INF/

│ │ │ │ ├── web.xml

│ │ │ │ ├── lib/

│ │ │ │ │ ├── [JAR-файлы библиотек]

│ │ │ │ ├── classes/

│ │ │ │ │ ├── [Скомпилированные классы Java]

│ │ │ ├── [Статические ресурсы и JSP/HTML-файлы]

├── target/

│ ├── [Скомпилированные файлы исходного кода и библиотеки]

├── [Другие каталоги и файлы, по необходимости]

1. **src/main/java/ (Исходные Java-коды приложения)**:
   * Этот каталог содержит все исходные Java-классы вашего приложения. Вы организуете их в соответствующих пакетах (packages) внутри этого каталога.
2. **src/main/resources/ (Ресурсы)**:
   * Здесь располагаются ресурсы, такие как файлы конфигурации, файлы свойств (properties), XML-файлы и другие ресурсы, которые используются вашим приложением.
3. **src/main/webapp/ (Веб-ресурсы)**:
   * Этот каталог содержит веб-ресурсы вашего веб-приложения, включая статические файлы (HTML, CSS, JavaScript), JSP-страницы и другие веб-ресурсы.
4. **src/main/webapp/META-INF/ (Мета-информация)**:
   * Каталог **META-INF** используется для размещения мета-информации о приложении, включая файл манифеста (**MANIFEST.MF**), который может содержать информацию о версии, авторстве и зависимостях приложения.
5. **src/main/webapp/WEB-INF/ (Информация о веб-приложении)**:
   * Каталог **WEB-INF** содержит важные компоненты веб-приложения:
     + **web.xml**: Файл конфигурации веб-приложения, в котором определяются сервлеты, фильтры, слушатели и другие параметры.
     + **lib/**: Каталог, в котором находятся JAR-файлы библиотек, необходимых для работы вашего веб-приложения.
     + **classes/**: Каталог, в который складываются скомпилированные классы Java вашего веб-приложения.
6. **target/ (Результаты сборки)**:
   * Этот каталог создается системой сборки (например, Maven) и содержит результаты компиляции и упаковки вашего проекта. Сюда помещаются скомпилированные классы, JAR-файлы и другие файлы, готовые к выполнению.
7. **Другие каталоги и файлы, по необходимости**:
   * Вы можете создавать другие каталоги и файлы в корне проекта в соответствии с требованиями вашего приложения. Например, это может быть каталог для тестов, документации, скриптов, изображений и так далее.

Эта структура проекта предоставляет основу для организации и разработки веб-приложения на Java с использованием технологии Java EE или сервлетов. Вы можете настраивать и расширять структуру в зависимости от специфики вашего проекта и требований.

На сегодняшний день имеется три семейства Java-технологий (и соответственно JSR-спецификаций):

- Java Platform, Standard Edition (Java SE);

- Java Platform, Enterprise Edition (Java EE);

- Java Platform, Micro Edition Specification (Java ME).

В составе Java SE содержится два основных продукта: Java Runtime Environment (JRE) и Java Development Kit (JDK). JRE представляет собой библиотеки, виртуальную Java-машину (JVM) и другие компоненты для исполнения приложений, разработанных на языке Java. Кроме того, в JRE включены технологии Java Plug-In (для запуска апплетов в web-браузерах) и Java Web Start (для разворачивания внешних приложений из сети Internet). JDK включает в себя все, что содержится в JRE, а также компиляторы языка Java, отладчик и дополнительные библиотеки. Спецификация Java SE определяет Java SE Application Interface (Java SE API) – программный интерфейс, позволяющий вызывать методы java-ядра и стандартные функции для создания настольных (desktop) и клиентских приложений, использовать графику. Более подробно с составом и назначением технологий Java SE можно ознакомиться на сайте [10]

Структура Tomcat

В структуре каталогов Tomcat папка `lib` предназначена для хранения библиотек (JAR-файлов), которые являются частью самого Tomcat или используются для его работы. Эти библиотеки обеспечивают функциональность сервера приложений и общие ресурсы, которые могут быть использованы веб-приложениями, развернутыми на Tomcat.

Вот некоторые из библиотек и JAR-файлов, которые обычно находятся в каталоге `lib` Tomcat:

1. \*\*servlet-api.jar\*\*: Этот JAR-файл содержит классы и интерфейсы, определенные в спецификации Java Servlet API. Он необходим для выполнения сервлетов на Tomcat.

2. \*\*jsp-api.jar\*\*: Этот JAR-файл содержит классы и интерфейсы, определенные в спецификации JavaServer Pages (JSP) API. Он необходим для выполнения JSP-страниц на Tomcat.

3. \*\*catalina.jar\*\*: Этот JAR-файл содержит классы, отвечающие за ядро Tomcat, включая обработку HTTP-запросов и управление жизненным циклом сервлетов.

4. \*\*jasper.jar\*\*: Этот JAR-файл отвечает за компиляцию JSP-страниц в Java-код и их выполнение.

5. \*\*commons-logging.jar\*\*: Библиотека для ведения журнала и логирования событий.

6. \*\*tomcat-dbcp.jar\*\*: Библиотека для поддержки пула соединений с базой данных (DBCP) на Tomcat.

7. \*\*websocket-api.jar\*\*: Содержит классы, связанные с веб-сокетами, которые позволяют веб-приложениям взаимодействовать в режиме реального времени.

Эти библиотеки предоставляют базовую инфраструктуру и функциональность для работы веб-приложений на Tomcat. Ваше веб-приложение может зависеть от некоторых из этих библиотек, и Tomcat автоматически предоставляет доступ к ним, чтобы обеспечить выполнение вашего приложения.