1. **Что такое веб-профиль Java EE**

Веб-профиль Java EE (или Jakarta EE) представляет собой набор спецификаций и API, которые ориентированы на создание веб-приложений и веб-сервисов. Этот профиль предоставляет базовые возможности для разработки веб-приложений, включая работу с сервлетами, JSP (JavaServer Pages), JSTL (JavaServer Pages Standard Tag Library), JTA (Java Transaction API), CDI (Contexts and Dependency Injection), JPA (Java Persistence API) и другими технологиями.

Веб-профиль был создан для того, чтобы облегчить разработку небольших и средних веб-приложений, не требующих всех возможностей Java EE. Это позволило разработчикам создавать более легковесные и быстрые приложения, а также уменьшило накладные расходы, связанные с развертыванием и управлением серверами приложений.

1. **Отличие веб-севера от сервера приложений в java**

Основные отличия между веб-сервером и сервером приложений:

1. **Функция:**

Веб-сервер обрабатывает HTTP-запросы и служит статическим содержимым.

Сервер приложений выполняет Java-приложения с бизнес-логикой и динамическим содержимым.

1. **Технологии:**

Веб-сервер ориентирован на HTTP и веб-технологии.

Сервер приложений предоставляет Java-специфичные API и сервисы.

1. **Примеры:**

Веб-: WildFly, TomEE.

1. **Масштабируемость:**

Веб-серверы масштабируются для обработки большего числа запросов вертикально.

Серверы приложений масштабируются для обработки бизнес-логики как вертикально, так и горизонтально.

1. **Web.xml в Java EE это** ***д***ескриптором развертывания web-приложения
2. **Java SE жизненный цикл**

Давайте рассмотрим жизненный цикл Java-приложения, начиная с создания файла Java с программой "Hello, World!" и заканчивая готовым для выполнения приложением:

1. **Создание Java-кода:**

Создайте файл с расширением .java, например, HelloWorld.java.

Откройте этот файл в текстовом редакторе или интегрированной среде разработки (IDE).

1. **Написание кода:**

Введите Java-код, который выводит "Hello, World!" на экран. Например:

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}}

1. **Компиляция Java-кода:**

Откройте командную строку или терминал.

Перейдите в каталог, где находится файл HelloWorld.java.

Выполните команду компиляции Java для создания файлов .class:

javac HelloWorld.java

Это создаст файл HelloWorld.class, содержащий байт-код вашей программы.

1. **Запуск Java-приложения:**

Выполните вашу программу с помощью команды java, указав имя класса с методом main (без расширения .class):

java HelloWorld

В результате выполнения на экране должно появиться "Hello, World!".

1. **Что нужно для запуска Jakarta (Java EE)?**

Для запуска Jakarta EE (ранее известной как Java EE) вам потребуется следующее:

* Среда выполнения Java (JRE) или Java Development Kit (JDK): Jakarta EE приложения требуют Java виртуальной машины (JVM) для выполнения. Вы можете использовать либо JRE, если вы только хотите выполнить Jakarta EE приложение, либо JDK, если вы планируете разрабатывать приложения на Jakarta EE.
* Сервер приложений Jakarta EE: Jakarta EE приложения обычно развертываются на серверах приложений, таких как Apache Tomcat, WildFly, GlassFish, Payara и другие. Вам нужно установить и настроить сервер приложений, который поддерживает Jakarta EE, чтобы развернуть и запустить ваше приложение.
* Jakarta EE приложение: Вам также понадобится Jakarta EE приложение, которое вы хотите развернуть и выполнить на сервере приложений. Это может быть WAR-файл (Web Application Archive) для веб-приложений или EAR-файл (Enterprise Archive) для корпоративных приложений.

1. **Как между собой связаны JDK и JRE?**

JDK (Java Development Kit) и JRE (Java Runtime Environment) связаны следующим образом:

* JDK содержит JRE: JDK включает в себя все компоненты JRE и дополнительные инструменты, и библиотеки, необходимые для разработки Java-приложений. Это включает компилятор Java (javac), отладчик (jdb), утилиты для создания JAR-файлов и другие инструменты разработки.
* JRE представляет собой минимальное окружение для выполнения Java-приложений: JRE содержит виртуальную машину Java (JVM), необходимую для запуска Java-приложений, и базовую библиотеку классов Java (Java Standard Library). JRE предназначен только для выполнения приложений и не содержит инструментов разработки.

Вкратце, JDK включает в себя JRE, но также добавляет к нему инструменты и библиотеки для создания и компиляции Java-приложений. Если вы разрабатываете Java-приложения, вам нужно использовать JDK. Если вы просто хотите выполнить готовое Java-приложение, JRE будет достаточно.