Java SE and Dalvik Virtual Machine (DVM) are both virtual machines that can run on different platforms. However, they are not the same thing.

Java SE is the standard edition of the Java platform, which includes a set of APIs (Application Programming Interfaces) for developing desktop and server applications. Java SE is designed to be portable across multiple platforms, allowing developers to write code once and run it on different operating systems.

On the other hand, the Dalvik Virtual Machine is the virtual machine that runs Android applications. Dalvik was specifically designed for mobile devices and is optimized for low memory and CPU usage. Unlike Java SE, which uses bytecode, Dalvik uses a different format called Dalvik Executable (DEX). DEX files are optimized for the limited resources available on mobile devices and are more compact than Java bytecode.

One of the reasons for the development of Dalvik was to avoid licensing fees associated with the Java platform. While Java SE is developed and maintained by Oracle, Dalvik is developed and maintained by Google as part of the Android Open Source Project (AOSP). In summary, while both Java SE and Dalvik are virtual machines that can run Java code, they are designed for different purposes and use different formats for their bytecode. Java SE is designed for desktop and server applications, while Dalvik is optimized for mobile devices and is used to run Android applications.

**Difference Table**

| **JVM(Java Virtual Machine)** | **DVM(Dalvik Virtual Machine)** |
| --- | --- |
| Stack-based VM that performs arithmetic and logic operations through push and pop operands. The result of operations is stored in stack memory. | Register-based VM that uses registers located in the CPU to perform arithmetic and logic operations. |
| Java source code is compiled into Java bytecode format(.class file) that further translates into machine code. | Source code files are first of all compiled into Java bytecode format like JVM. Further, the **DEX compiler(dx tool)** converts the Java bytecode into Dalvik bytecode(classes.dex) file that will be used to create the **.apk file**. |
| More information is required to the VM for data loading and manipulation as well as method loading in the stack data structure. | Instruction size is larger as it needs to encode the source and destination register of the VM. |
| Compiled bytecode size is compact because the location of the operand is implicitly on the operand stack. | Compiled bytecode size is larger as each instruction needs all implicit operands. |
| The executable file for the device is **.jar file**. | The executable file for the device is **.apk file**. |
| A single instance of JVM is configured with shared processes and memory space in order to run all deployed applications. | The device runs multiple DVM instances with a separate process in shared memory space to deploy the code of each application. |
| Supports multiple operating systems like Linux, Windows, and Mac OS. | Support only the Android operation system. |