

Bootcamp Java Profesional

Acceso a Recursos en el Classpath

Javier Ramírez

Software Professional since 2001

Java Developer / Tech Lead

@javaMexico Founder/Co-Leader



@_benek | @javamexico

linktree.com/javamexico

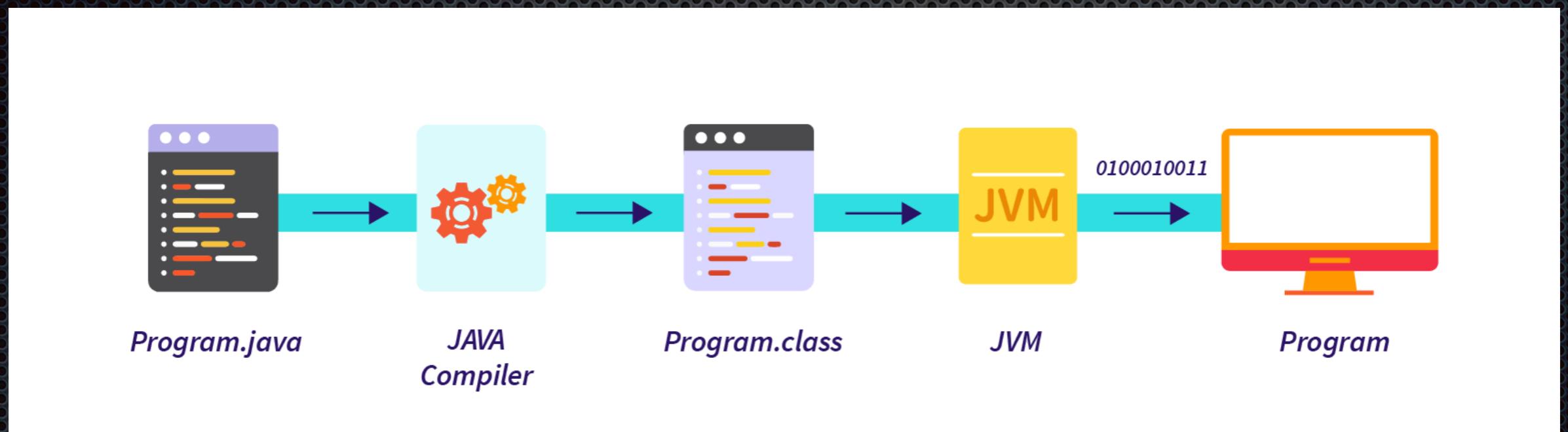


Source Code

- El código fuente de esta clase estará disponible en:
- <https://github.com/benek/Bootcamp-Java-CodigoFacilito>

Acceso a Recursos en el Classpath

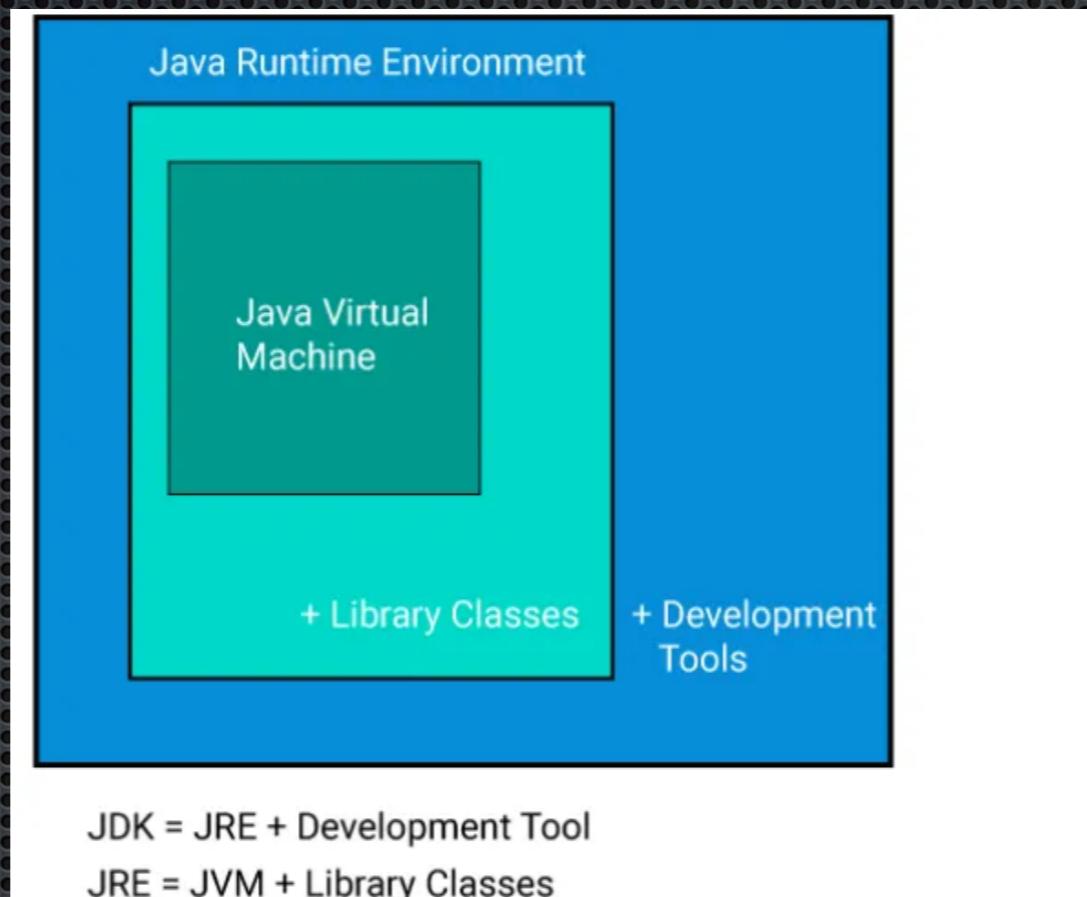
¿Cómo corren las apps Java?



Java Architecture

- JDK - Entorno para desarrollo
- JRE - Entorno para ejecución
- JVM - Intérprete
- JIT - Compilación nativa en tiempo real

Java Architecture



¿Cómo corren las apps Java?

- Dos pasos principales, el primero mediante un compilador independiente de OS, el segundo mediante una VM específica para cada OS

Compilation



Execution



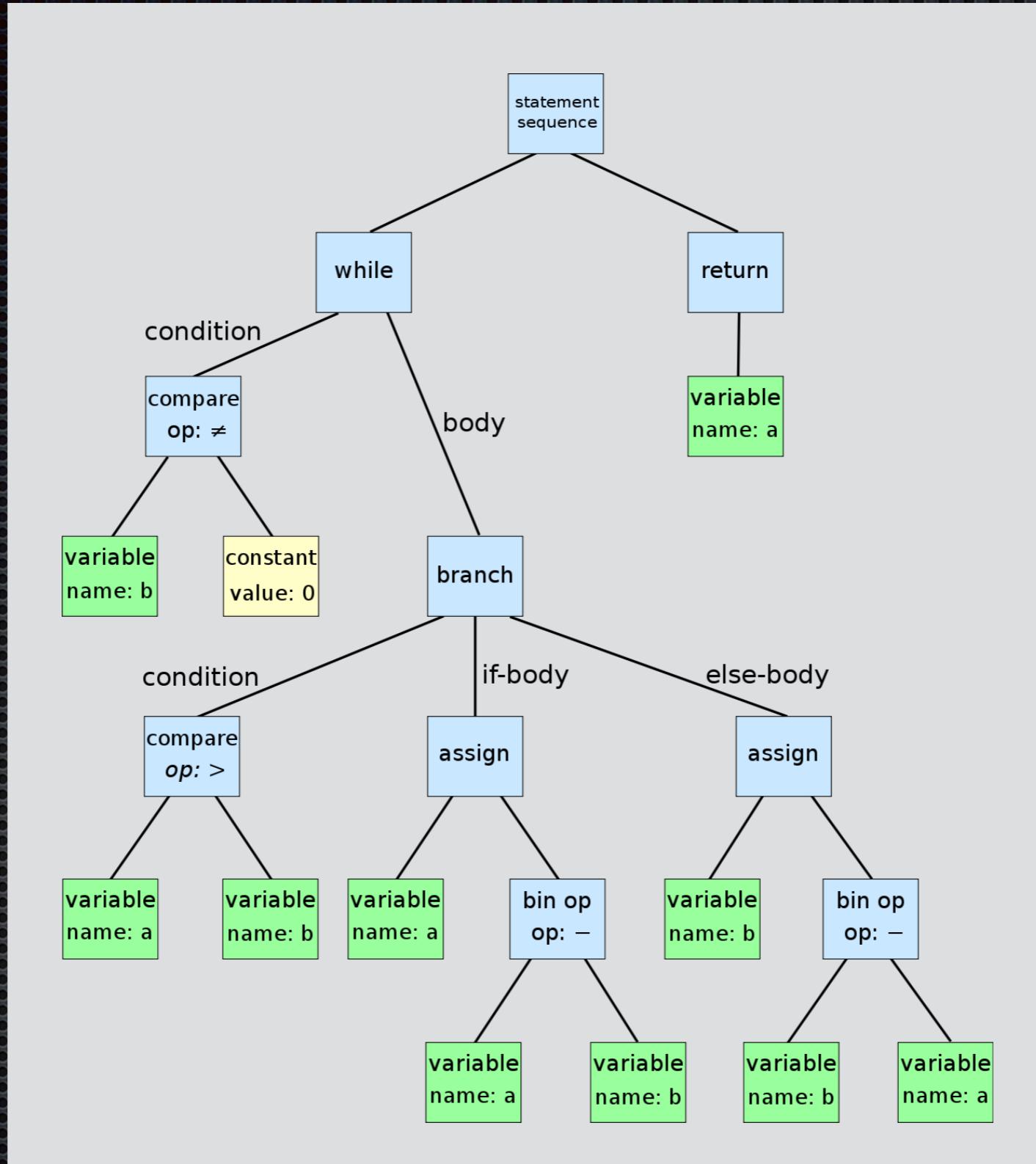
Compilation

- Step 1: Parse into AST Nodes (Abstract Syntax Tree)
- Step 2: Enter symbols into the symbol table
- Step 3: Process annotations (if requested)
- Step 4: Attributes the AST (name resolution, type check)
- Step 5: Dataflow analysis (check assignments, reachability)
- Step 6: Desugar (rewrite) AST
- Step 7: Generate .class files

Execution

- Step 1: Classloader
- Step 2: Bytecode Verifier per class (check variable initialization, method call types vs object refs, access, stack overflows)
- Step 3: Just-In-Time Compiler, final stage, converts to machine code

AST



Classloaders

- Se encargan de cargar dinámicamente clases hacia la JVM
- Evita que el runtime Java se tenga que preocupar por archivos o filesystems
- Las clases en Java no son todas cargadas en un mismo momento, o al mismo tiempo, sino cuando son requeridas

Classloaders

- No todas las clases son cargadas por el mismo Classloader
- Dependiendo del tipo y ubicación de cada clase, se decidirá qué Classloader es el encargado de cargarla
- Existen dos tipos de classloaders
 - Primordial - Embebido en todas las JVMs (default)
 - Non-primordial - User-defined, para personalizar el proceso de class-loading, muchas bibliotecas y frameworks ocupan sus propios classloaders

The Java Classloader

- Es parte del JRE (Java Runtime Environment)
- Todas las clases son cargadas en base a su *qualified name*
- Si alguna no se encuentra, implicará una de las siguientes excepciones
 - *ClassNotFoundException*
 - *NoClassDefFoundError*

The Java Classloader

- El classloader de Java tiene 3 tipos por defecto
 - BootStrap Classloader
 - Extension Classloader
 - System Classloader

BootStrap Classloader

- El classloader inicial, está hecho en código nativo
- Se encarga de iniciar el proceso cuando la JVM lo invoca
- Carga el primer Java Classloader y las clases presentes en *rt.jar*

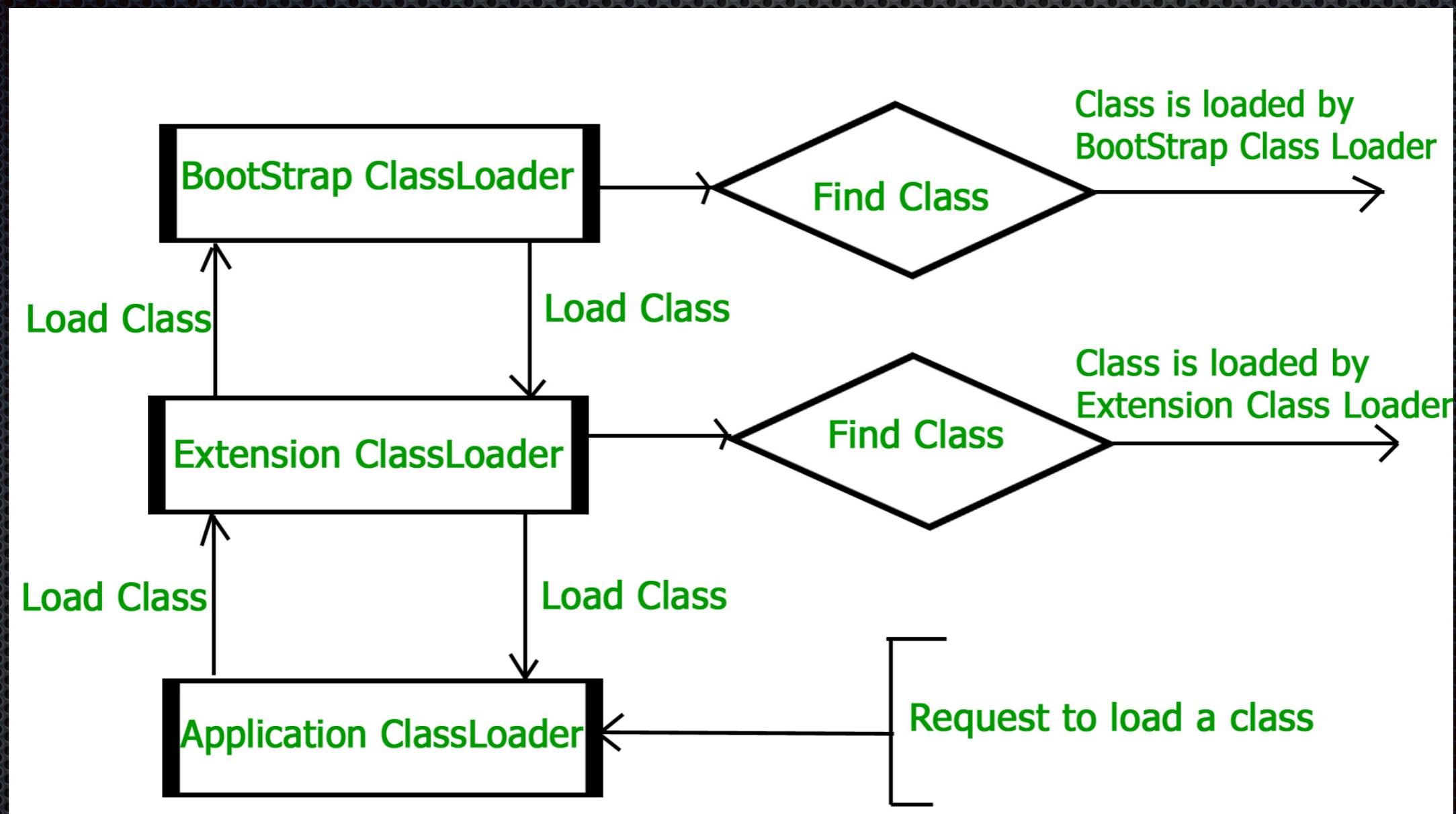
Extension Classloader

- *Child* del BootStrap Classloader
- Carga las extensiones de las clases *core* de Java
- Localiza las extensiones presentes en *jre/lib/ext*, o de algún directorio configurado por la propiedad *java.ext.dirs*

System Classloader

- También llamado *Application Classloader*
- *Child* del *Extension Classloader*
- Carga las clases para la aplicación, presentes en la variable de entorno **CLASSPATH**, o en presentes en las opciones de la línea de comandos
 - `-classpath`
 - `-cp`

The Java Classloaders



java.lang.ClassLoader

- Clase Abstracta para los Classloaders

- `loadClass(String name, boolean resolve)`
- `defineClass()`
- `findClass(String name)`
- `findLoadedClass(String name)`
- `Class.forName(String name, boolean initialize, ClassLoader loader)`

Classpath

- Es la ruta o rutas en las que el JRE buscará por clases y archivos de recursos
- Le indica a Java en donde encontrar las clases definidas por los usuarios, que no son extensiones ni forman parte de la plataforma Java

Classpath

- El classpath requiere encontrar clases que fueron compiladas por el compilador *javac*
- Por defecto, busca en el directorio actual

Classpath en JDK Commands

- Los comandos *java*, *javac*, *jdb*, *javah* y *jdeps*, contienen una opción disponible para setear el classpath:
 - *-classpath*
 - *-cp* (abreviación)
- Ejemplo:
 - `java -classpath <path> package.Class`

Variable de entorno

- Alternativamente, los comandos de la JDK buscarán también definiciones de Classpath en una variable de entorno
 - CLASSPATH
- Usar la variable de entorno no es el método recomendado oficialmente
 - `set CLASSPATH=path1;path2`

Ejemplos de uso

- `java -classpath /Users/benek/MyClasses org.benek.Main`
- `java -classpath /Users/benek/MyClasses/myclasses.jar org.benek.Main`
- `java -classpath /java/MyClasses;/java/OtherClasses ...`
- `java -classpath mydir;mydir/*;mydir/other org.benek.Main`

Problemáticas

- *ClassNotFoundException* y *NoClassDefFoundError* son dos problemas comunes en aplicaciones Java
- Ambas refieren al proceso de class loading
- *ClassNotFoundException*
 - Cuando la clase no se encontró en el classpath en tiempo de ejecución
- *NoClassDefFoundError*
 - Cuando la definición de la clase estuvo presente en la compilación, pero la clase no se encuentra en tiempo de ejecución

Q&A