Vývoj Aplikácií s Viacvrstvovou Architektúrou

JVM

IT ACADEMY

STU FIIT

Lektor: PhDr. Ing. Mgr. Miroslav Reiter, DiS., MBA, MPA
Kontakt: miroslav.reiter@it-academy.sk

Čo JVM a ako ho efektívne využívať?

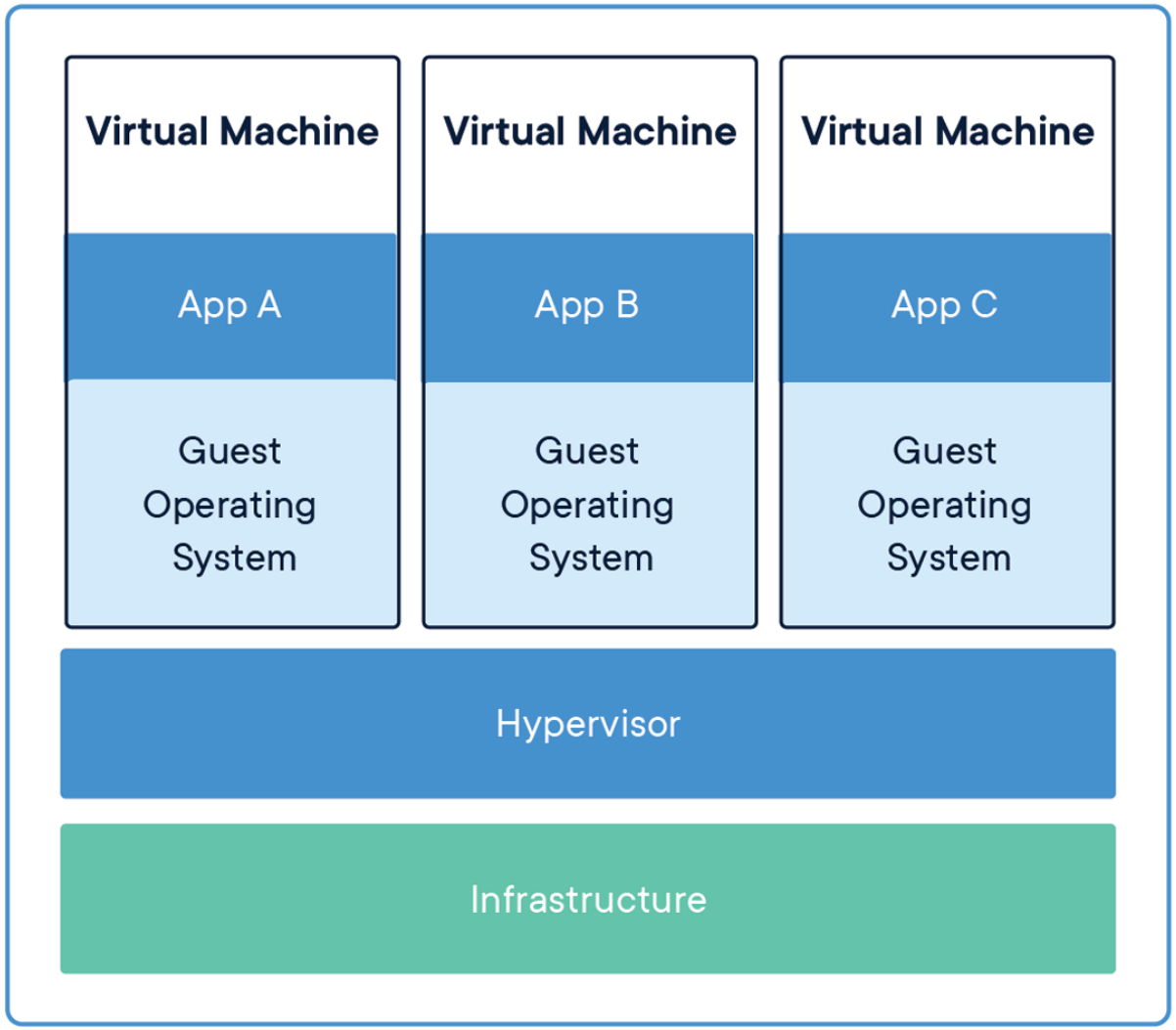
Koľko zariadení používa Javu?

- **Desktopové aplikácie (Java SE)**
- **Java a Android**
  - Smartfóny
  - Tablety
  - Notebooky
  - Set-top boxy
  - Navigačné a parkovacie systémy
  - Terminály
  - Lekárske prístoje
  - Tlačiarne, webkamery, vysávače, chladničky
- **Čipové karty (Java Card)**
- **Enterprise a serverové aplikácie (Java EE)**
  - Podniková architektúra (EE)
  - Weby (Liferay, Spring)
  - Testovanie softvéru (Junit, Selenium)

Virtual Machine

- Virtuálne stroje (VM) sa bežne používajú na **distribúciu programov v architektonicky neutrálnom formáte**, ktorý je možné **ľahko interpretovať** alebo **kompilovať**

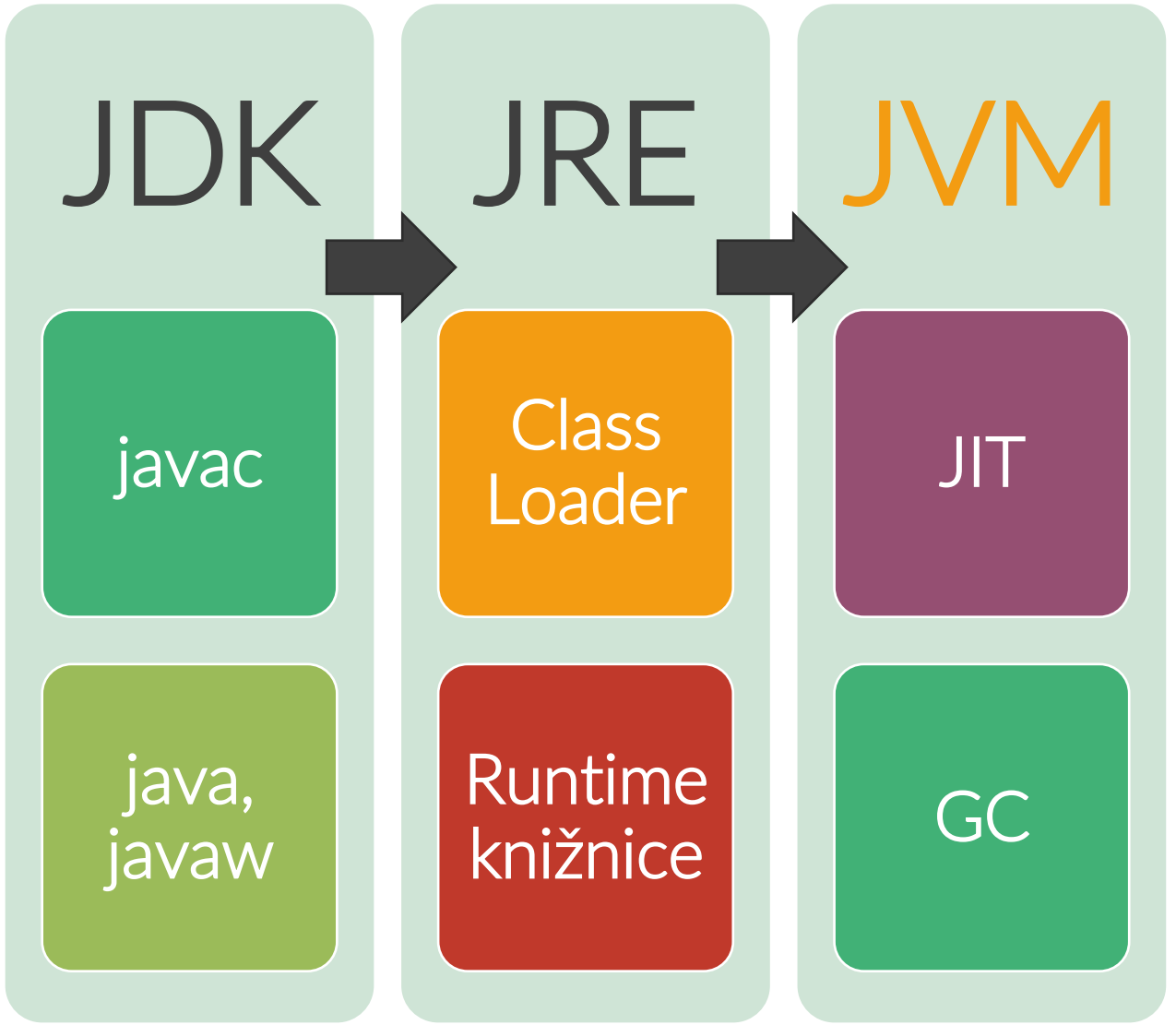
- **Hardvérový virtuálny stroj**
- **Aplikačný virtuálny stroj**



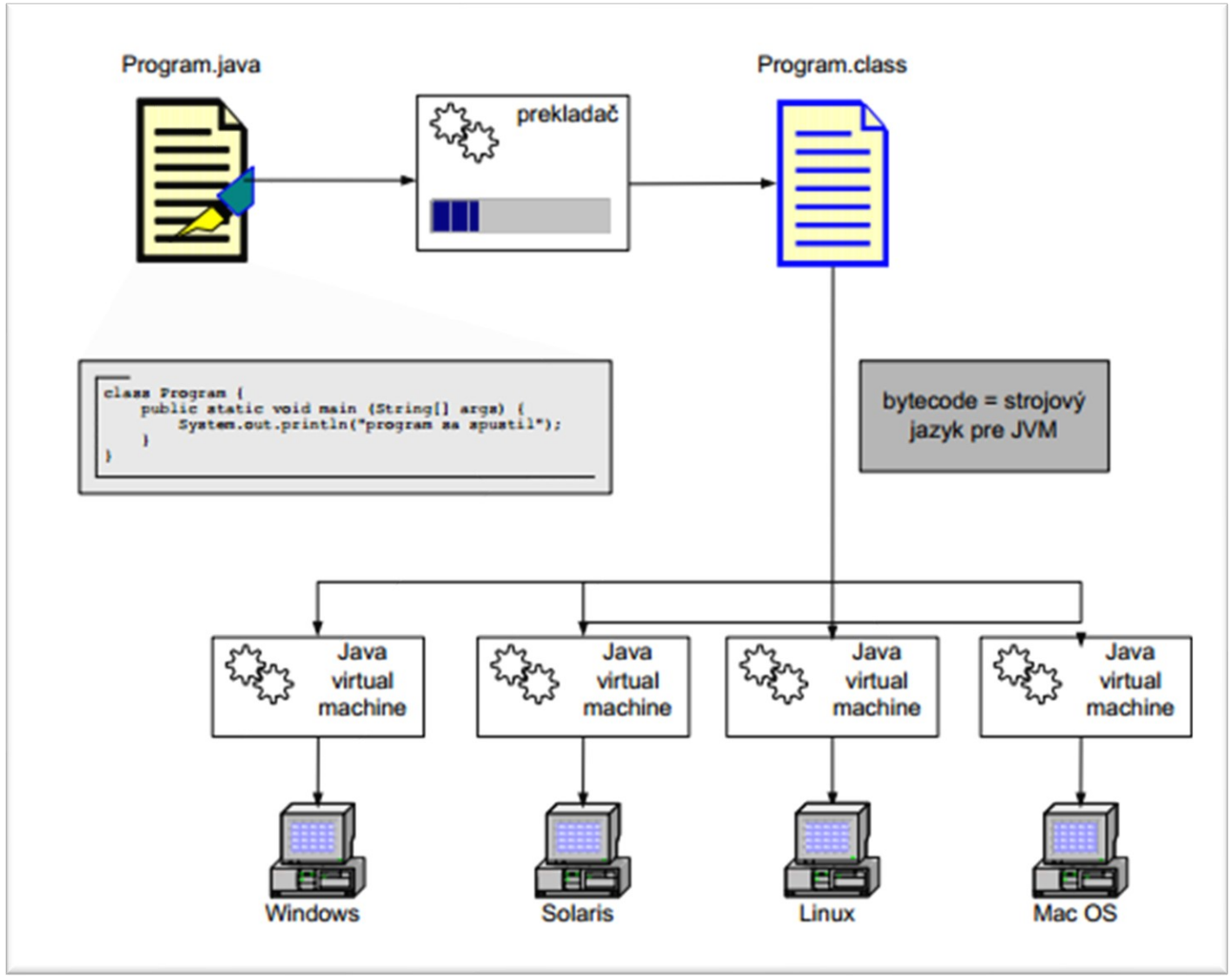
Java platforma

Zahrňuje 2 komponenty:

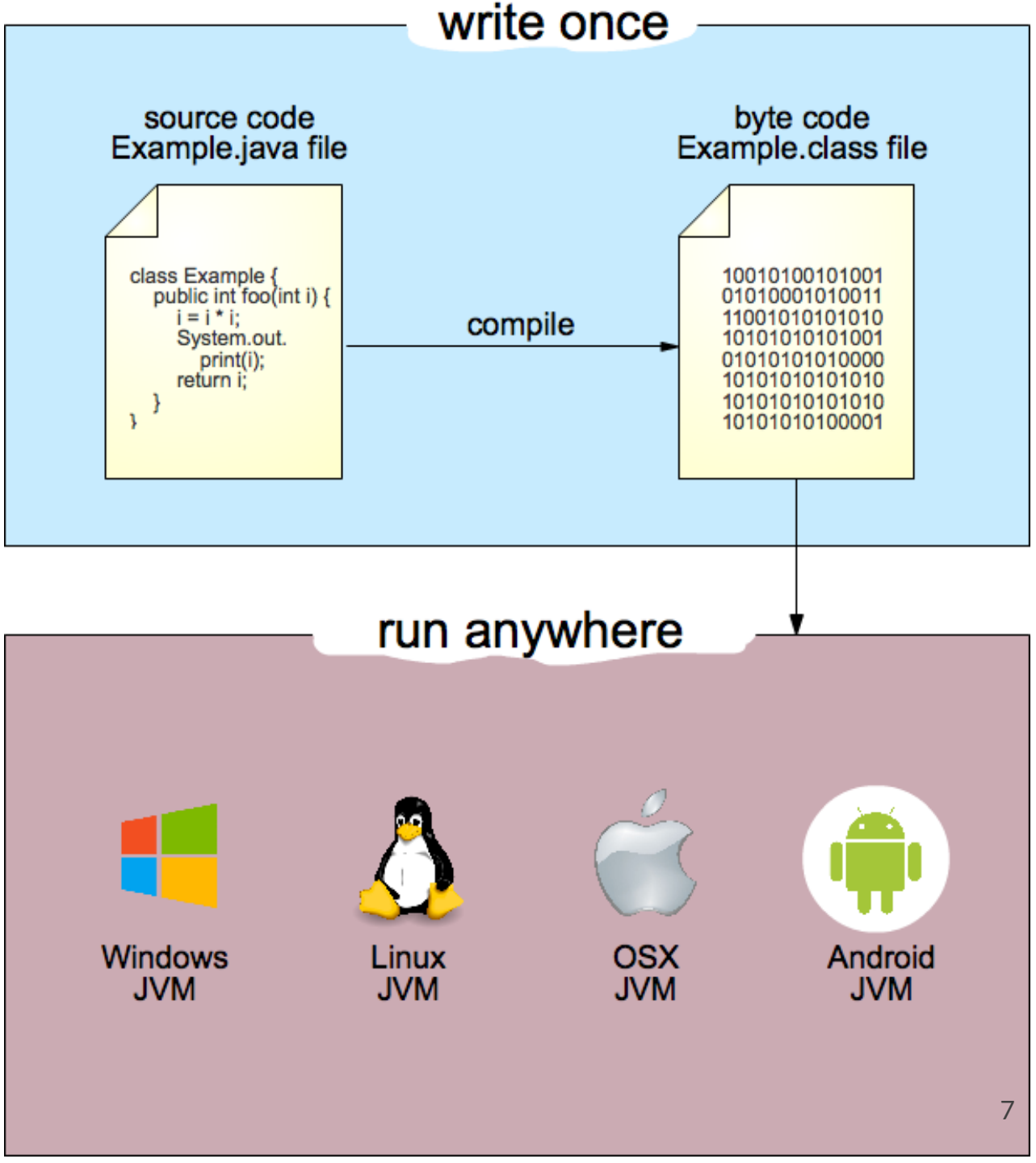
1. **Virtuálny stroj jazyka java JVM**
2. **Aplikačné programové rozhranie API**



Java platforma pre jazyk Java



Java platforma pre Javu



Základné Java CMD príkazy

- **javac** – java compiler
  - ❑javac PrvaTrieda.java
- **java** – spustenie preloženého programu
  - ❑java PrvaTrieda
  - ❑java -jar PrvyProgram.jar
- **jar** – práca so súbormi jar
  - ❑jar -cfe PrvyProgram.jar PrvaTrieda PrvaTrieda.class
  - ❑jar -cfm PrvyProgram.java Manifest.txt PrvaTrieda.class
- **javadoc** – vytvorenie dokumentácie zo zdrojových súborov

```
Manifest-Version: 1.0
Created-By: meno autora
Main-Class: PrvaTrieda
```

na konci prazdny riadok

Select Administrator: C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. Všetky práva vyhradené.

C:\WINDOWS\system32>cd C:\Users\Administrator\Desktop\Main

C:\Users\Administrator\Desktop\Main>javac Main
error: Class names, 'Main', are only accepted if annotation processing is explicitly requested
1 error

C:\Users\Administrator\Desktop\Main>javac Main.java

C:\Users\Administrator\Desktop\Main>java Main.class
Error: Could not find or load main class Main.class
Caused by: java.lang.ClassNotFoundException: Main.class

C:\Users\Administrator\Desktop\Main>java Main
Ahoj Java...

C:\Users\Administrator\Desktop\Main>fsutil file createnew manifest.man 0
File C:\Users\Administrator\Desktop\Main\manifest.man is created

C:\Users\Administrator\Desktop\Main>notepad manifest.man
```

manifest.man – Poznámkový blok

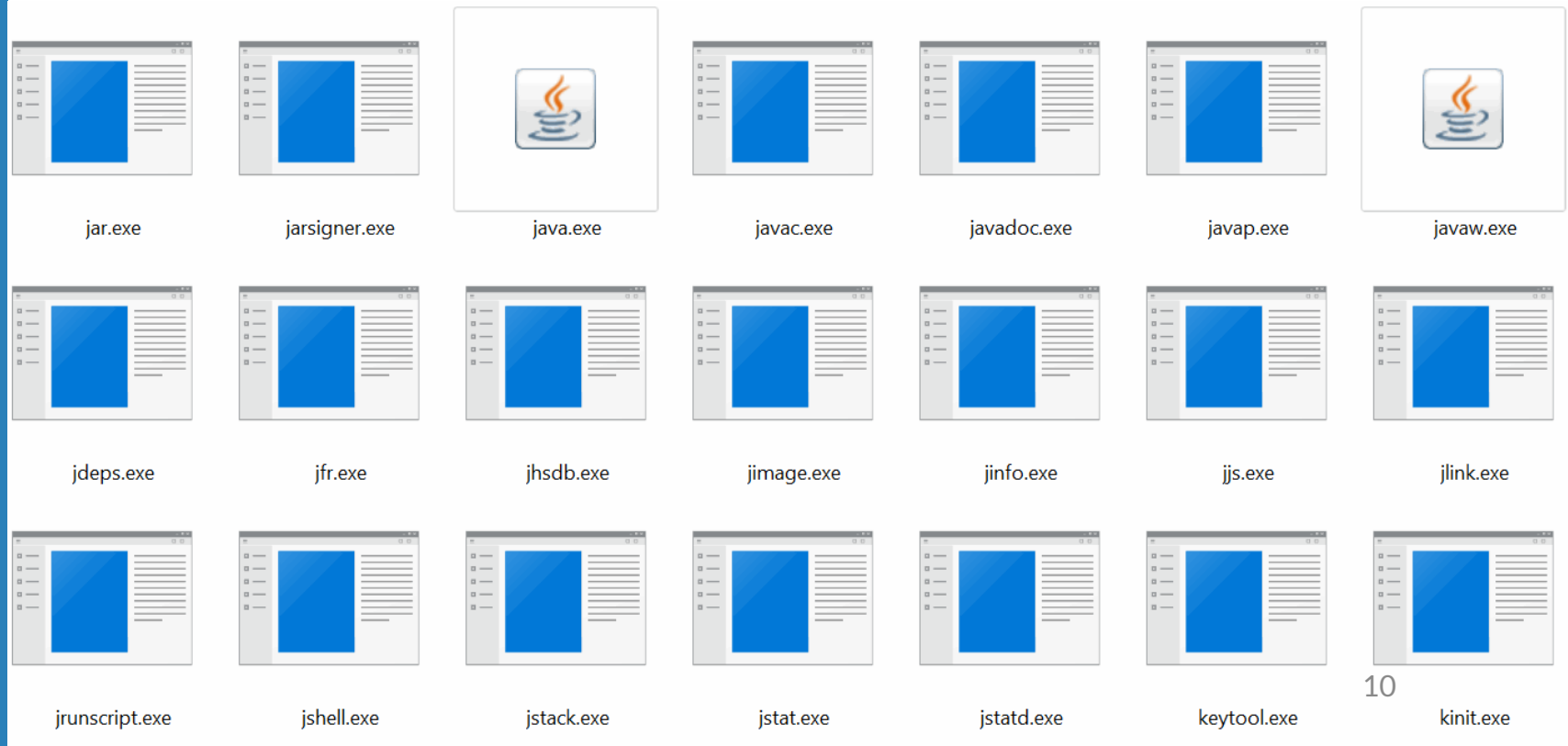
Súbor Úpravy Formát Zobraziť Pomocník

```
Manifest-Version: 1.0
Main-Class: ProgramJava
Created-By: 1.8.0_144 (Oracle Corporation)
```

Ln 3, Col 43 | 100 % | Windows (CRLF) | UTF-8

Vytvorenie spustiteľného súboru

1. *.jar - Jar archive (pre štandardné SE, EE aplikácie + slúži aj ako knižnice príp. frameworky)
2. *.war - Web archive (pre EE webové aplikácie, môže v sebe obsahovať viac * .jar archívov)
3. *.ear - Enterprise archive (pre EE projekty, môže v sebe obsahovať viac * .jar a * .war archívov)



WORA

stands for

Write Once Run Anywhere

Abbreviations.com

Čo je Java Virtual Machine (JVM)?

Sada programov a dátových štruktúr, ktorá využíva modul virtuálneho stroja

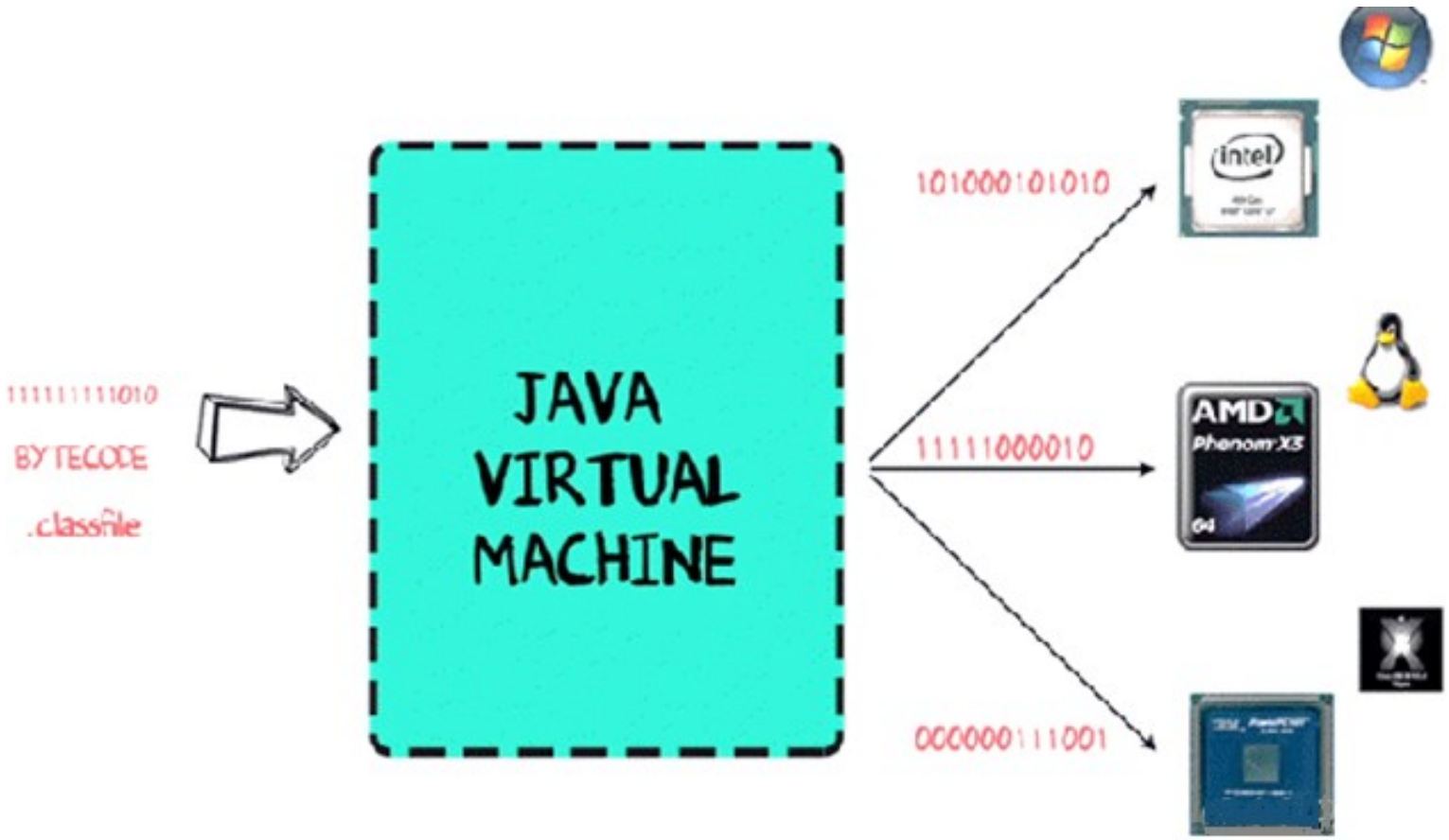
JVM

(Java Virtual Machine)

Úlohou je spracovať iba Java bytecode

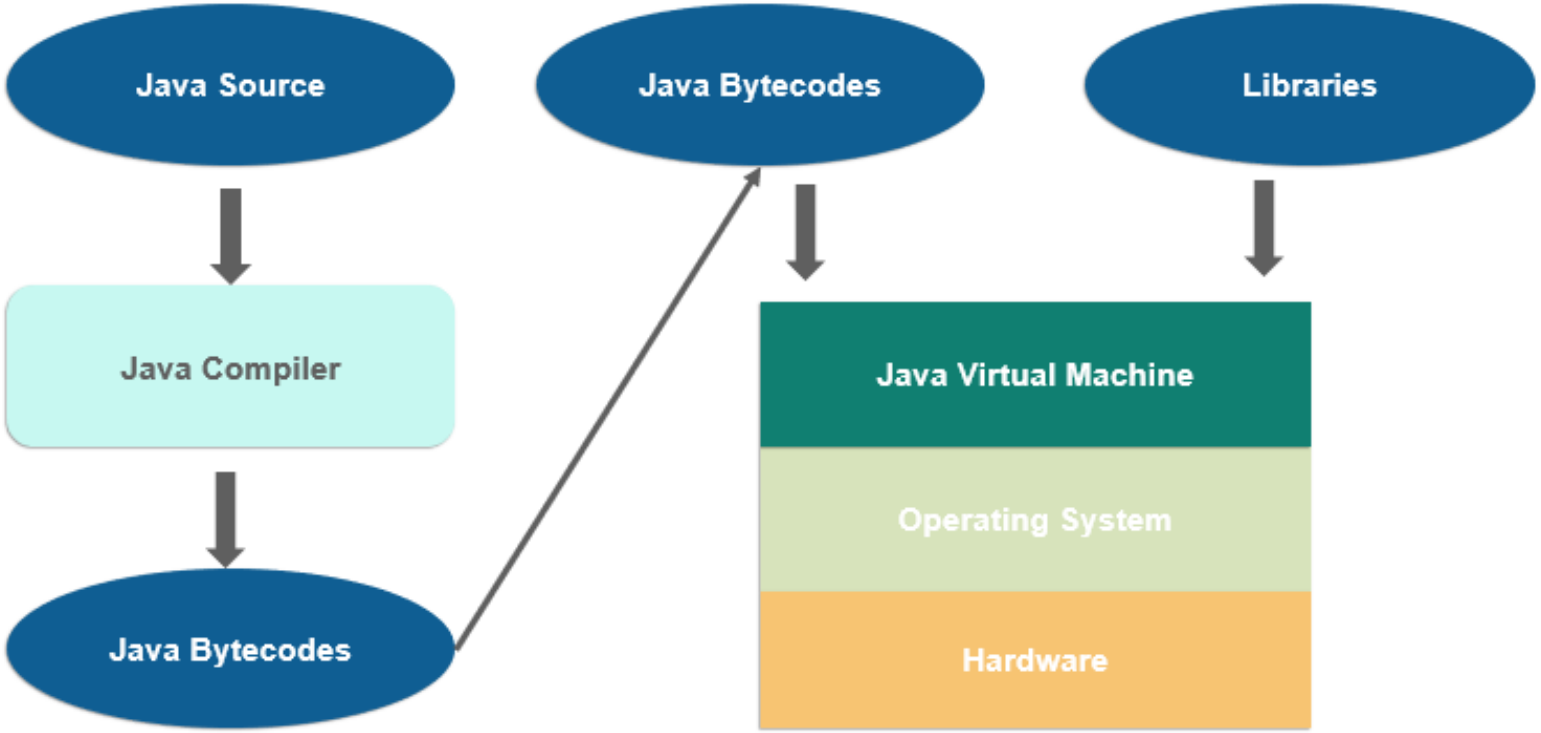
Čo je (JVM) podľa Oracle?

- Základným kameň platformy Java
- Súčasť technológie zodpovedná za nezávislosť od hardvéru a operačného systému
- Abstraktný výpočtový stroj
- Má inštrukčnú sadu a za behu manipuluje s rôznymi oblasťami pamäte
- Je bežné implementovať programovací jazyk pomocou



Prečo potrebujeme JVM?

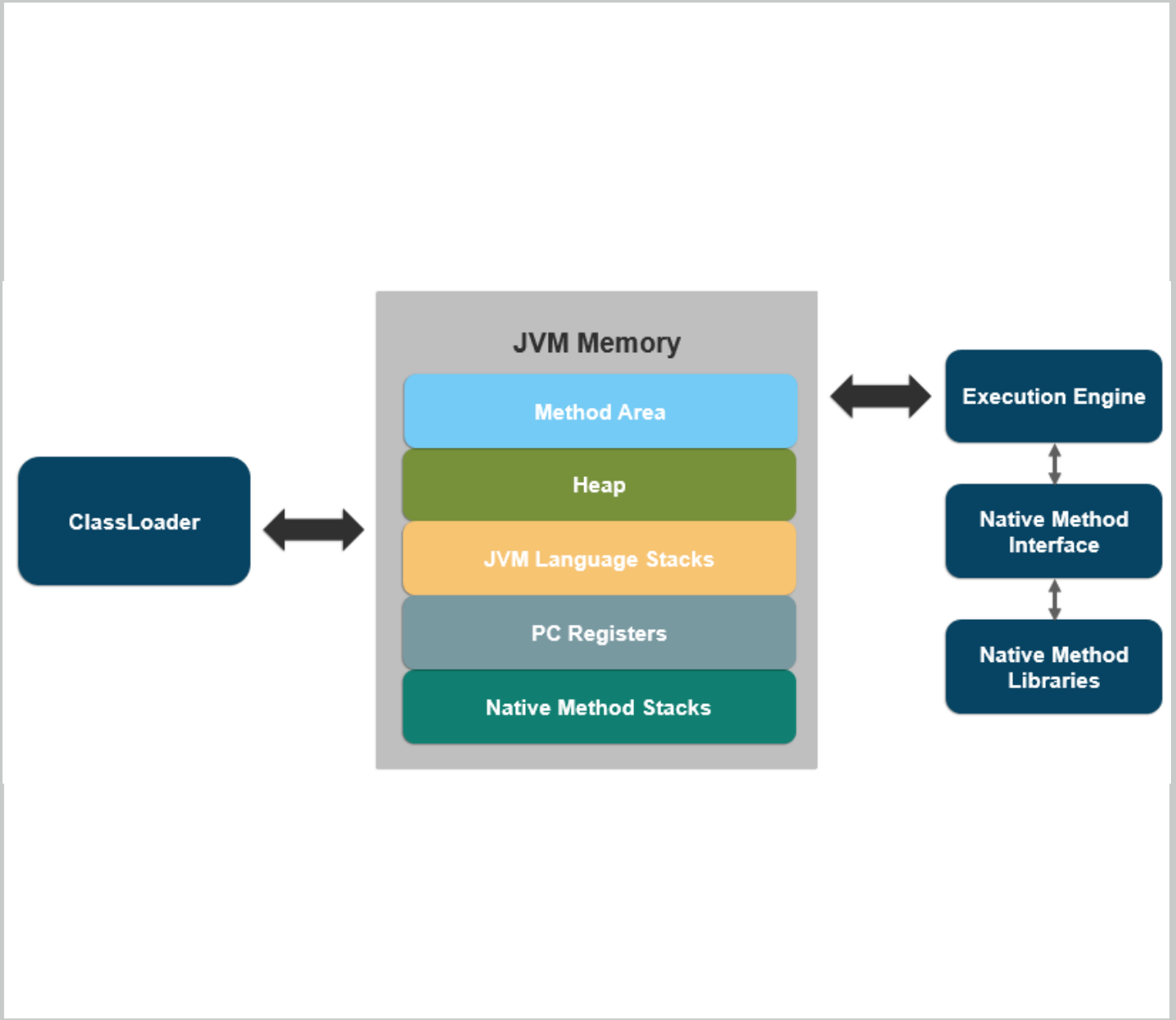
- Načíta kód
- Overenie kódu
- Vykonanie kódu
- Poskytuje run-time prostredie pre aplikácie
- Pamäťová oblasť
- Registre
- Garbage collection heap
- Hlásenie závažných chýb
- Poskytuje formát súboru triedy



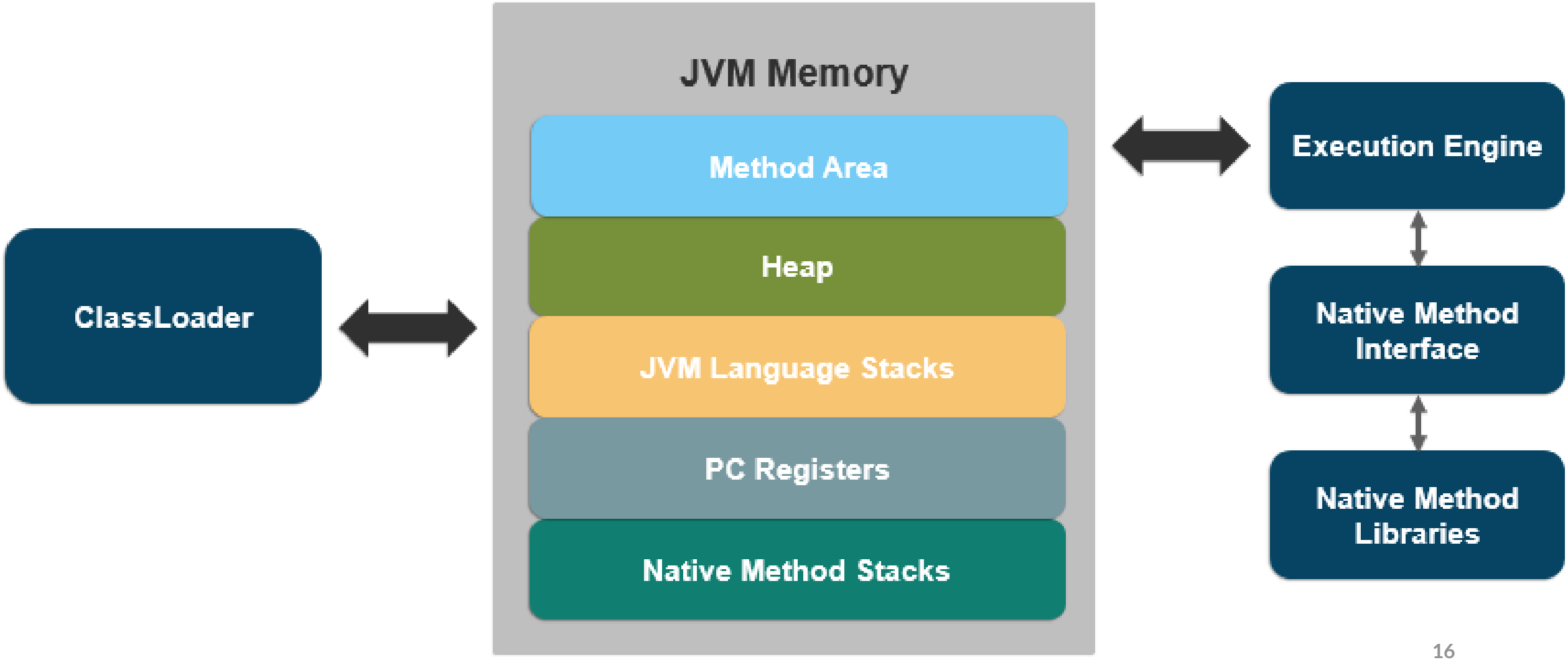
Architektúra JVM

Built-in classloaders

- Bootstrap ClassLoader
- Extension ClassLoader
- System / Application ClassLoader
- Loading Operation
- Linking Operation
- Initialization

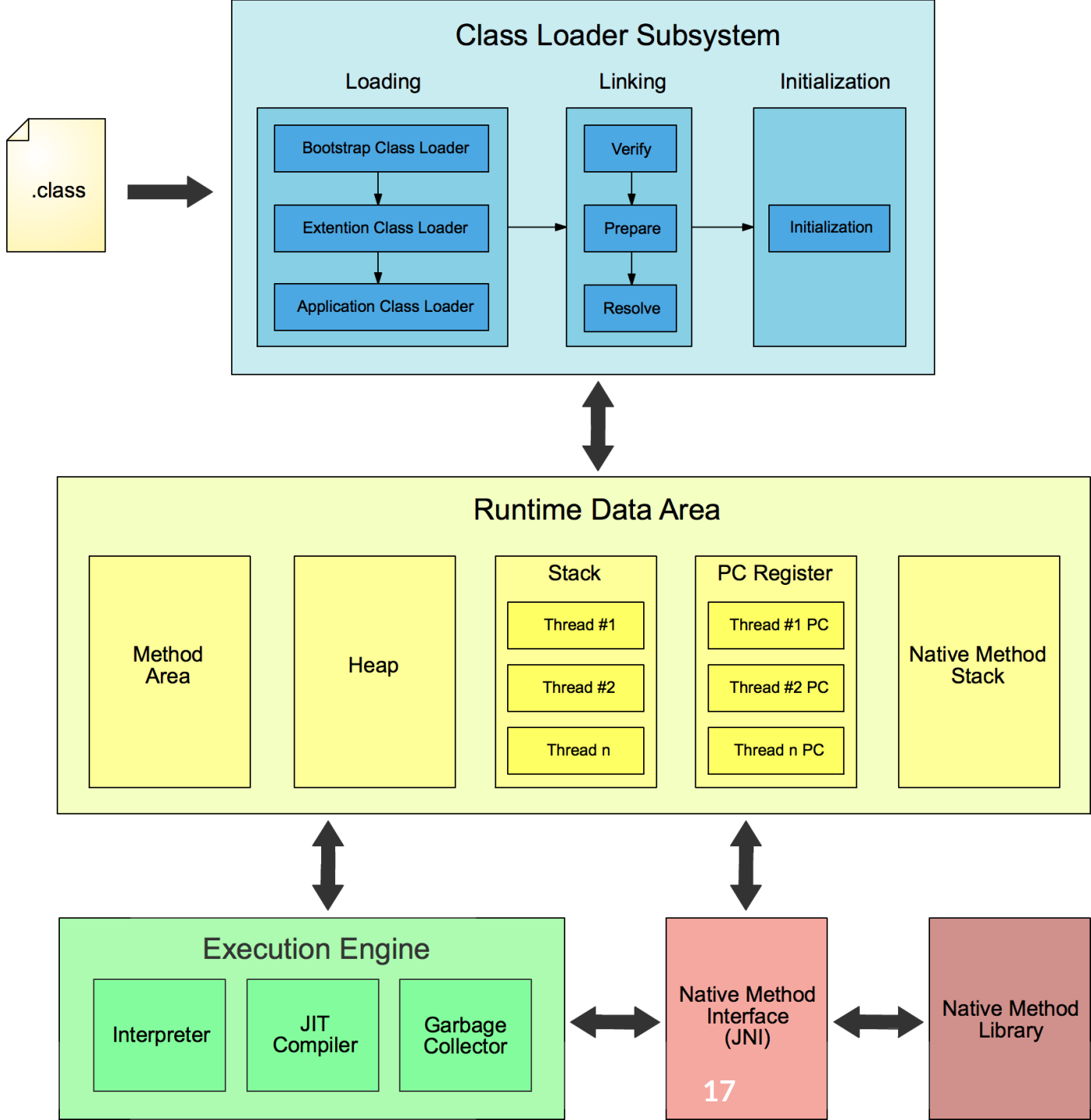


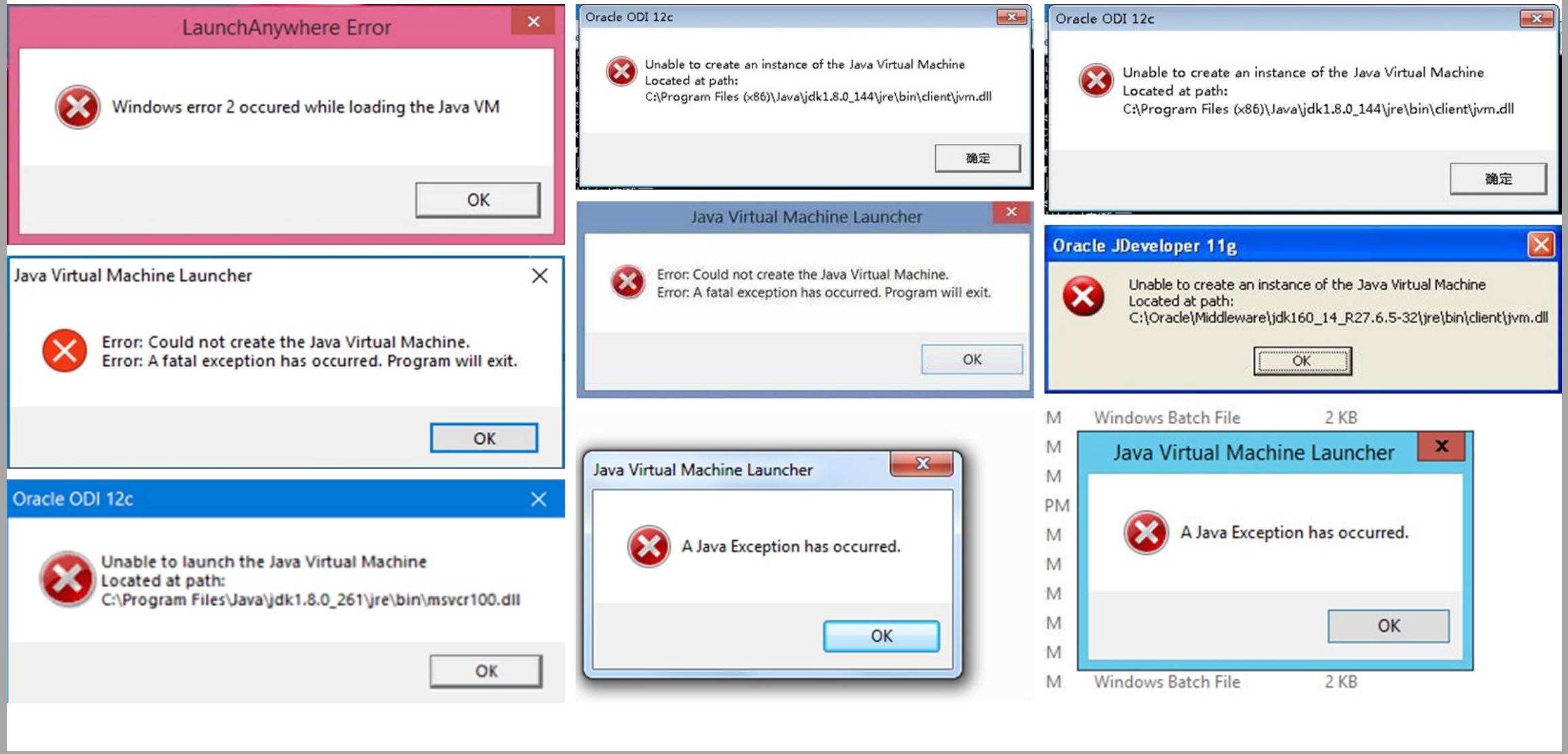
JVM Memory


ClassLoader
JVM Memory
Method Area
Heap
JVM Language Stacks
PC Registers
Native Method Stacks
Execution Engine
Native Method Interface
Native Method Libraries

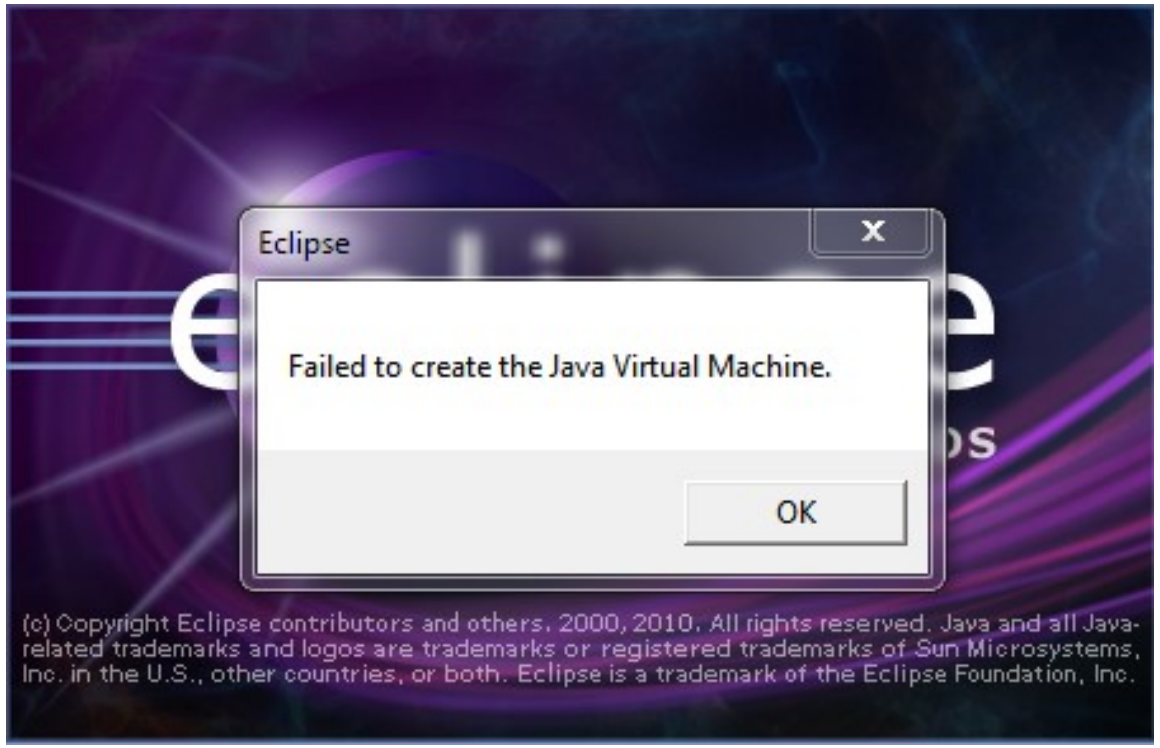
Execution Process

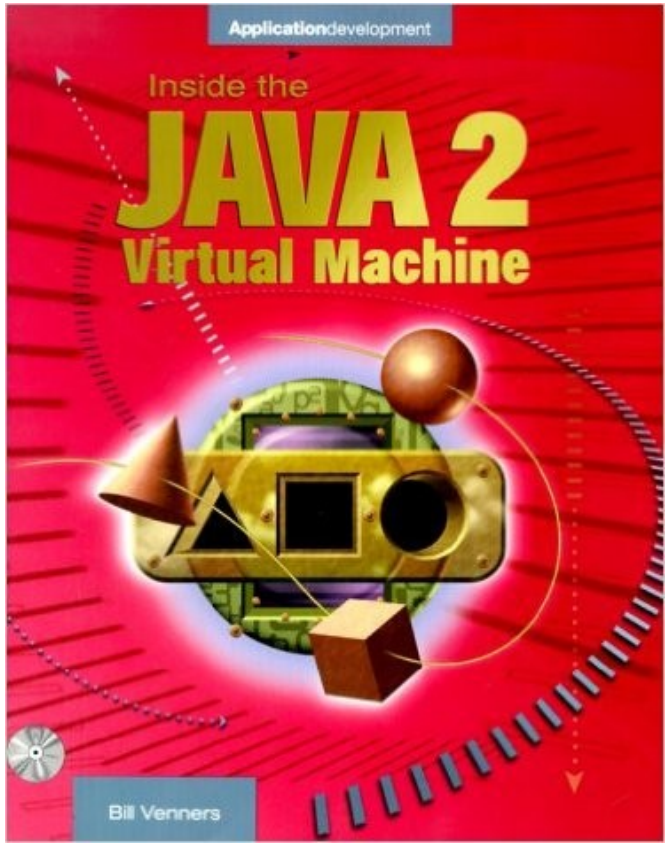
1. Hlavná metóda
2. Súbory .class
3. Overenie Classloaderu
4. Bajt kód na strojový kód

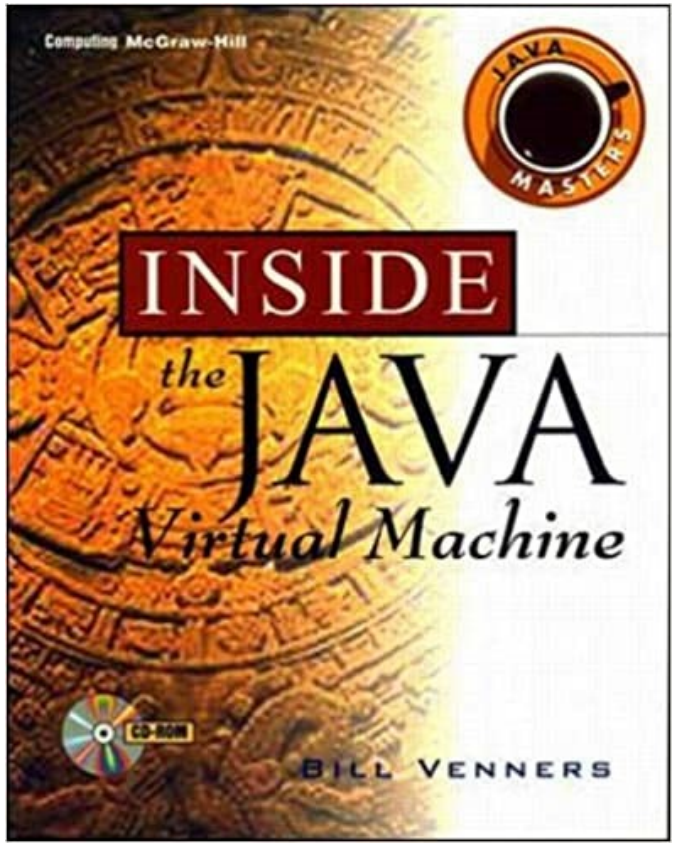


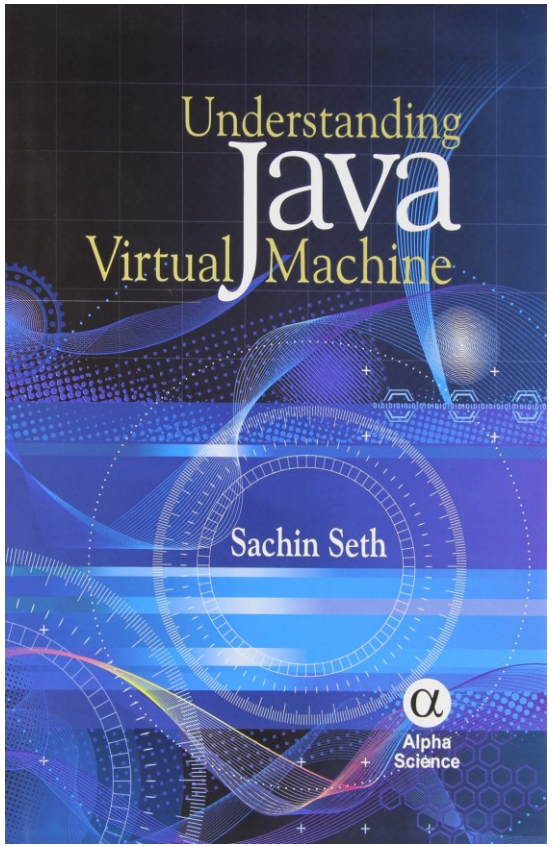

LaunchAnywhere Error
Windows error 2 occured while loading the Java VM
OK
Oracle ODI 12c
Unable to create an instance of the Java Virtual Machine
Located at path:
C:\Program Files (x86)\Java\jdk1.8.0_144\jre\bin\client\jvm.dll
確定
Oracle ODI 12c
Unable to create an instance of the Java Virtual Machine
Located at path:
C:\Program Files (x86)\Java\jdk1.8.0_144\jre\bin\client\jvm.dll
確定
Java Virtual Machine Launcher
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
OK
Java Virtual Machine Launcher
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
OK
Oracle JDeveloper 11g
Unable to create an instance of the Java Virtual Machine
Located at path:
C:\Oracle\Middleware\jdk160_14_R27.6.5-32\jre\bin\client\jvm.dll
OK
Oracle ODI 12c
Unable to launch the Java Virtual Machine
Located at path:
C:\Program Files\Java\jdk1.8.0_261\jre\bin\msvcr100.dll
OK
Java Virtual Machine Launcher
A Java Exception has occurred.
OK
M Windows Batch File 2 KB
M
M
PM
M
M
M
M
M
M
Java Virtual Machine Launcher
A Java Exception has occurred.
OK
Windows Batch File 2 KB

Aj priamo v IDE, nástrojoch, Mac OS X...

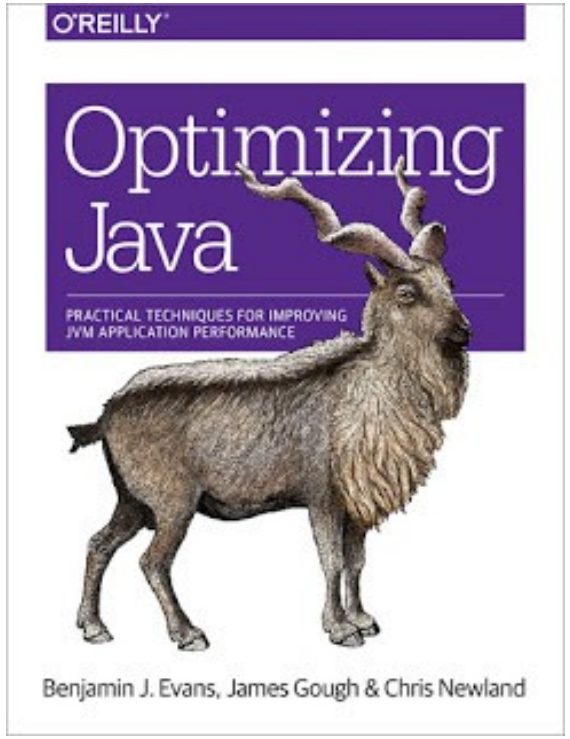


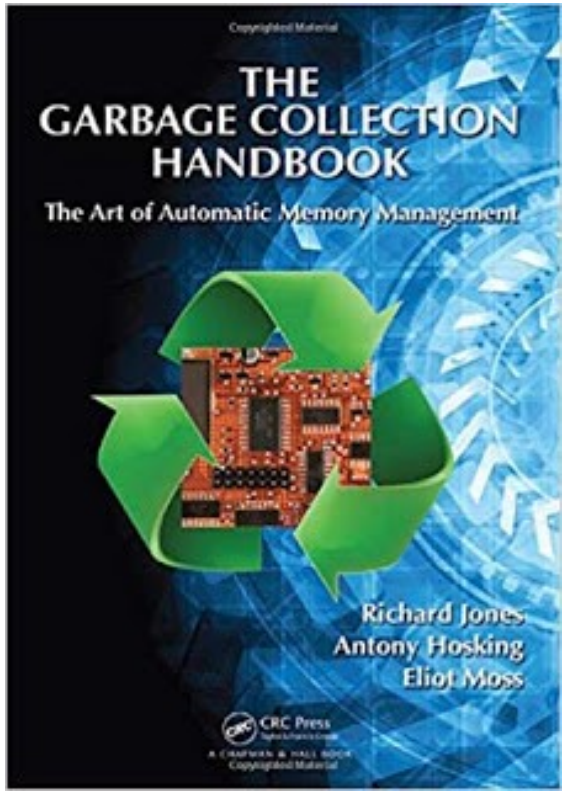


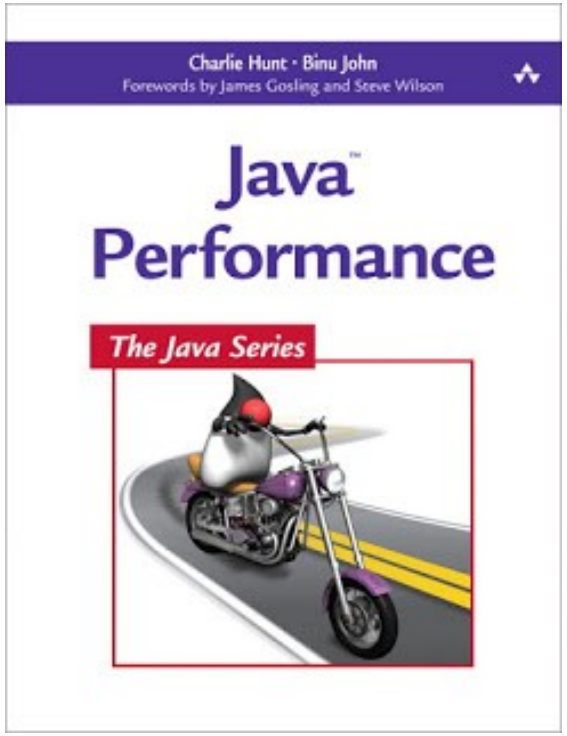


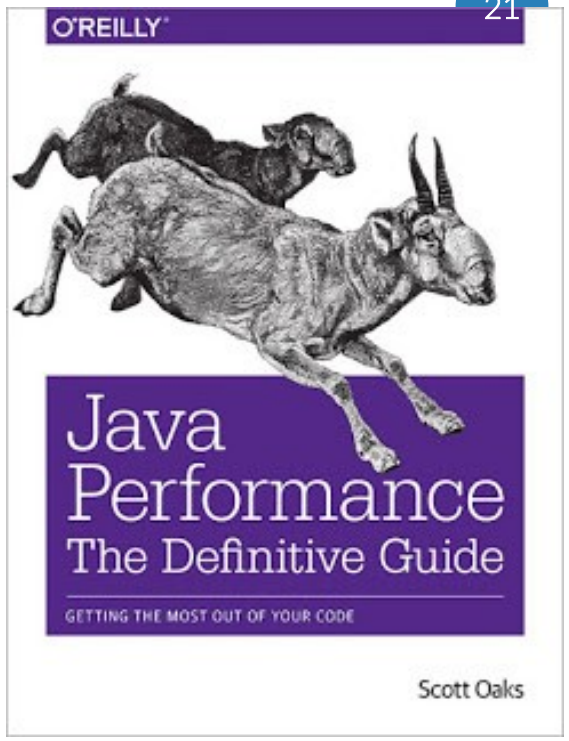


Knihy a literatúra





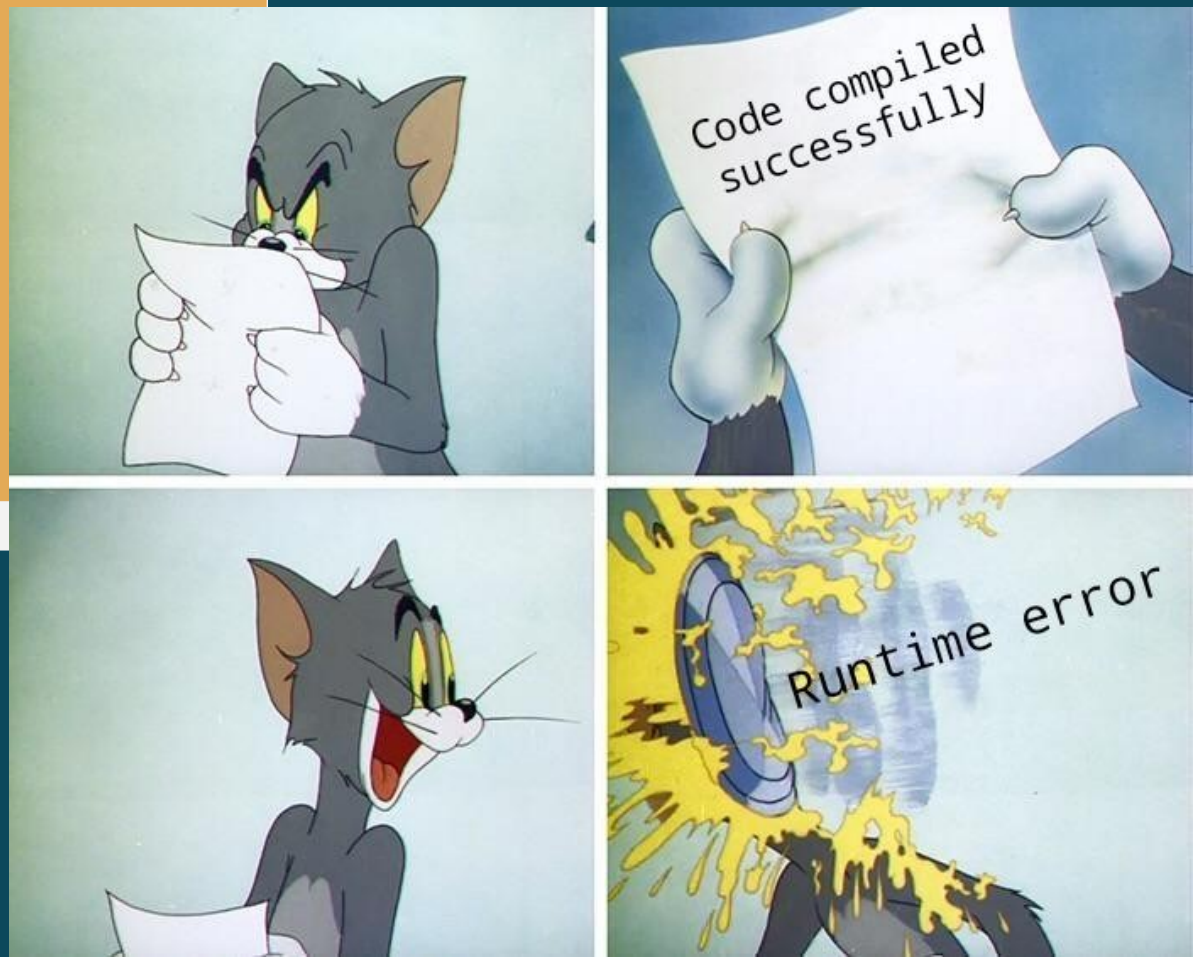




Čo aspoň odporúčam

Java magazine

A first look at Java 17: Encapsulating the runtime internals



Week of May 3, 2021

ORACLE

Java SE > Java SE Specifications

Java Language and Virtual Machine Specifications

Java SE 16

Released March 2021 as JSR 391

The Java Language Specification, Java SE 16 Edition

- HTML | PDF
- Preview feature: Sealed Classes

The Java Virtual Machine Specification, Java SE 16 Edition

- HTML | PDF
- Preview feature: Sealed Classes

Java SE 15

Released September 2020 as JSR 390

The Java Language Specification, Java SE 15 Edition

- HTML | PDF
- Preview features: Pattern matching for `instanceof`, Records, Sealed Classes

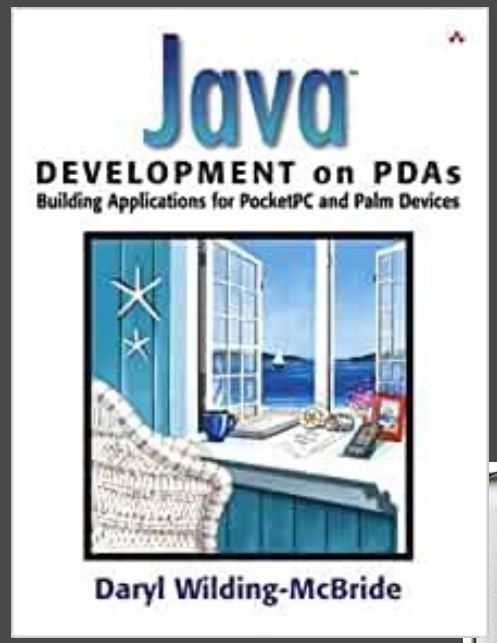
The Java Virtual Machine Specification, Java SE 15 Edition

- HTML | PDF
- Preview features: Records, Sealed Classes

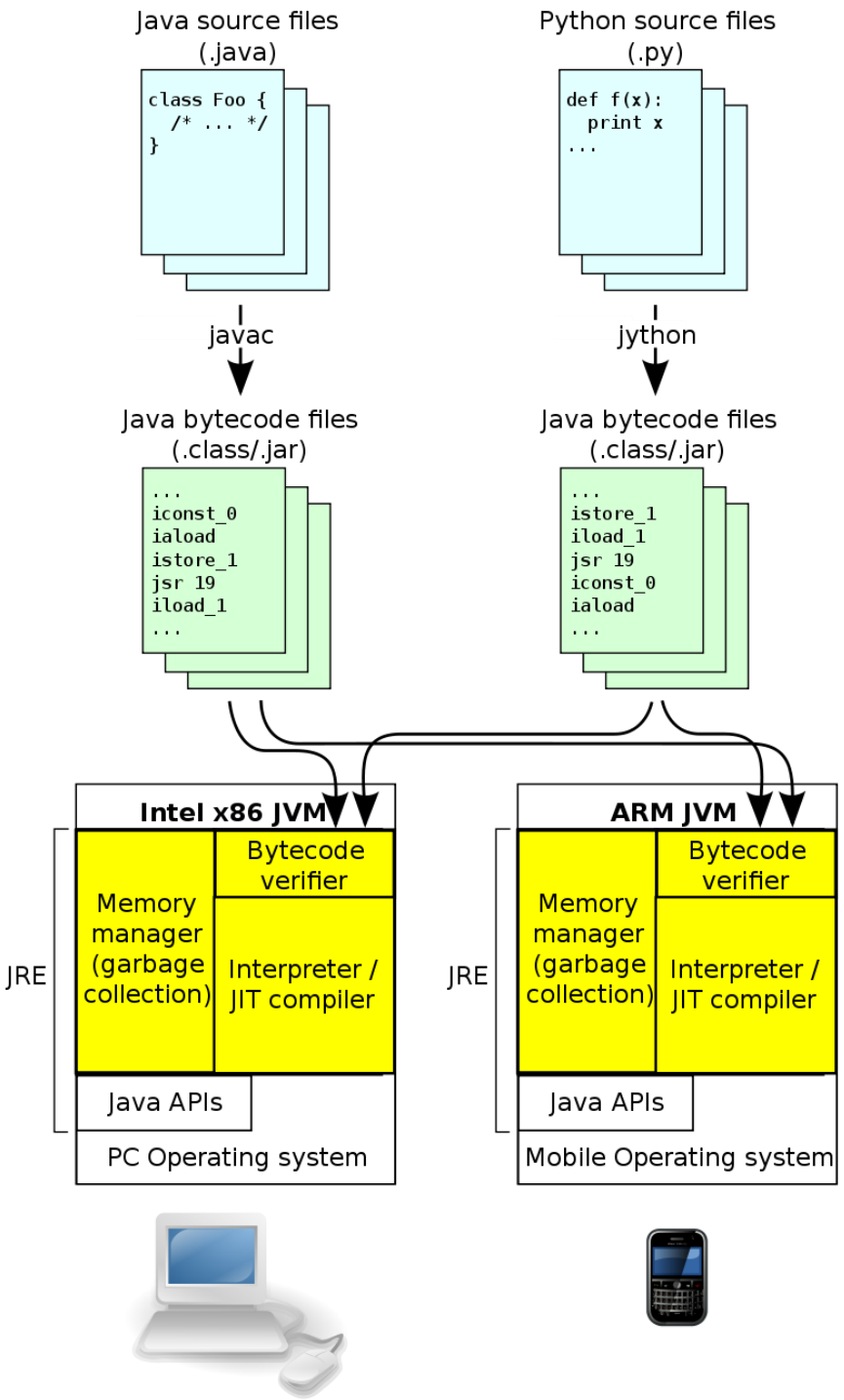
Java SE 14

https://docs.oracle.com/javase/specs/

Prvá implementácia Java Virtual Machine



Programmers can write a program once, and it will run on any machine supplying a Java run-time environment



Java Virtual Machine (JVM)

- **JVM nevie nič** o programovacom **jazyku Java**
- **Vie** iba **konkrétnom binárnom formáte**, **formáte súboru triedy**
- **Súbor triedy obsahuje pokyny Java Virtual Machine** (alebo bytecodes) a **tabuľku symbolov,** ako aj ďalšie pomocné informácie
- Kvôli bezpečnosti ukladá virtuálny počítač Java silné syntaktické a štrukturálne obmedzenia pre kód v súbore triedy
- **Akýkoľvek jazyk s funkčnosťou**, ktorý je **možné vyjadriť v podobe platného súboru triedy**, však môže byť hostený virtuálnym počítačom Java
- Implementátori iných jazykov, ktorých priťahuje všeobecne dostupná platforma nezávislá od stroja, sa môžu obrátiť na Java Virtual Machine

Podpora ďalších jazykov

- Ruby s JRuby
- JavaScript s Nashorn
- Python s Jython
- Groovy
- Scala
- Clojure
- Kotlin

Zoznam JVM (Free a Open source)

- Azul Zulu
- Codename One
- Eclipse OpenJ9
- GraalVM
- HotSpot
- JamVM
- leJOS
- Maxine

Microsoft Virtual Machine

The Microsoft Java Virtual Machine is no longer available from Microsoft directly due to legal wrangling with SUN, however it still can be downloaded...

Microsoft VM build 3810 for Windows Vista, Windows XP, Windows 2000, Windows 95/98, Windows Me, Windows NT 4.0. 5.07 MB

Microsoft Java Virtual Machine download links:

- http://www.saigoninfo.com/msjavx86.exe
- http://www.meetingworks.com/files/msjavx86.exe
- http://www.linktivity.com/home/java/msjavx86.exe
- http://www.zxll.com/files/msjavx86/
- http://visiteinteractive.free.fr/VM_java/msjavx86.exe
- http://www.laplink.com/download/pcsync/upgrade/msjavx86.exe
- https://www.alibre.com/alibrelibraries/ftp/JavaVM/9xNT4/msjavx86.exe
  or
- Search for msjavx86.exe!

Steps to install the Microsoft Virtual Machine:

1. Download the file using the links above and save it to your harddrive.
2. Once the file is on your harddrive, execute it and thus load the Virtual Machine.
3. As the file begins execution, answer Yes to the License Agreement question and then once complete, re-boot your PC.
4. Once the PC is re-booted you should be set to go.

One way to get Microsoft Java Virtual Machine

First go to this site and get the older version of the MS Java Virtual Machine:

Informácie o verziách

Name	Creator	First public release	Latest stable version	Latest release date	Cost, availability	License
Eclipse OpenJ9 (formerly IBM J9)	IBM	?	0.25.0[1]	16 March 2021[2]	Free	Apache License 2.0 Eclipse Public License 2.0
GCJ	GNU	6 September 1998	6.4 (Terminal)	4 July 2017	Free	GPL version 2 or later, with the "libgcc exception"[3]
HotSpot, OpenJDK edition	Sun Microsystems, Oracle	27 April 1999	jdk-16	16 March 2021	Free	GPL version 2 only
HotSpot, Oracle JDK edition	Sun Microsystems, Oracle	27 April 1999	jdk 16	16 March 2021	Free	Proprietary
HotSpot, Java SE embedded edition	Sun Microsystems, Oracle	27 April 1999	?	?	Commercial	Proprietary[4]
HotSpot, Zero port	Gary Benson[5]	?	?	?	Free	GPL version 2 only
IKVM.NET	Jeroen Frijters	28 June 2004	7.0.4335.0	5 December 2011	Free	zlib License[6]
JAmiga	Peter Werno, Joakim Nordström	19 May 2005[7]	1.2	6 January 2014	Free	GPL version 2 or later
JamVM	Robert Lougher	13 March 2003	2.0.0	30 July 2014	Free	GPL version 2 or later
Jato VM	Pekka Enberg and contributors[8]	?	0.3[9]	4 January 2012[9]	Free	GPL version 2 only[8]
JC virtual machine	Archie L. Cobbs	?	1.4.7	13 November 2005	Free	LGPL version 2.1 or later
Jikes RVM	IBM	14 October 2001	3.1.4[10]	18 February 2016[10]	Free	Eclipse Public License version 1.0[11]
Kaffe	Transvirtual Technologies	1996	1.1.9	22 February 2008	Free	GPL version 2 or later[12]
Mysaifu JVM	?	16 April 2005[13]	0.4.8[13]	5 March 2010[13]	Free	GPL version 2 only[13]
SableVM	Sable Research Group	?	1.13	30 March 2007[14]	Free	LGPL version 2.1 or later

Technické informácie

JVM	Status	Latest supported Java version	Supported class libraries			Performance		
			GNU Classpath	OpenJDK	Other	Interpretation	AOT	JIT
GCJ	No longer maintained or distributed by GNU as of GCC 7 [15]	?	Yes	No		Yes	Yes	No
HotSpot, OpenJDK edition	Reference implementation.	1.8	No	Yes		Yes	No	Yes
HotSpot, Oracle JDK edition	Reference implementation.	1.8	No	Yes		Yes	No	Yes
HotSpot, Java SE embedded edition		?	No	Yes		Yes	No	Yes
HotSpot, Zero port	Interpreter-only port of OpenJDK using almost no assembly language and designed to be very portable.	1.7	No	Yes		Yes	No	No
IKVM.NET		?	?	?		Whatever the .NET runtime uses		
JAmiga		1.4[16]	Yes[17]	No		Yes[18]	No	No
JamVM		1.8[19]	Yes[20]	Yes[20]		Yes[20]	No	Yes[20]
Jato VM		1.6[21]	Yes[9]	No[22]		No[22]	No[22]	Yes[9]
JC virtual machine	Translates Java to C and compiles it with a C compiler.	1.4[23]	Yes[24]	No		Yes[25]	Yes[25]	Yes[25]
Jikes RVM		1.6[26]	Yes[27]	Port[28][29]	Apache Harmony[27]	No[30]	?	Yes
Kaffe		1.4[12]	Yes[12]	No		Yes[12]	No[12]	Yes[12]
Mysaifu JVM		?	Yes[31]	No		Yes[32]	No[32]	No[32]
SableVM	Unmaintained	1.4	Yes	No		Yes	No	No

Podporované architektúry CPU

JVM	x86	x86-64	SPARC	MIPS	Itanium	Power ISA	ARM	Alpha	S/390	z/Architecture	m68k
GCJ	Yes[33]	Yes[33]	Yes[33]	Yes[33]	Yes[33]	Yes[33]	Yes[33]	Yes[33]	Yes[33]	No[33]	No[33]
HotSpot, OpenJDK edition	Yes[34]	Yes[34]	Solaris only[34]	Port[35][36]	No[34]	PowerPC/AIX port[37]	Yes[34]	No[34]	No[34]	No[34]	No[34]
HotSpot, Oracle JDK edition	Yes[34]	Yes[34]	Solaris only[34]	No[34]	Java 1.6[38]	No[34]	Yes[34]	No[34]	No[34]	No[34]	No[34]
HotSpot, Java SE embedded edition	Yes[4]	Yes[4]	Yes[4]	?	?	Yes[4]	Yes[4]	?	?	?	?
HotSpot, Zero port	Yes[39]	Yes[39]	No[39]	Yes[39]	Yes[39]	Yes[39]	Yes[39]	Yes[39]	No[39]	Yes[39]	No[39]
IKVM.NET	Any architecture with a .NET framework.										
JAmiga	?	?	No	No	No	Yes	No	No	No	No	Yes
JamVM	Yes[20]	Yes[20]	Yes[20]	Yes[20]	No[20]	Yes[20]	Yes[20]	No[20]	No[20]	No[20]	No[20]
Jato VM	Yes[9]	Under development[40]	No	No	No	Under development[40]	Preliminary ARMv5 support[22]	No	No	No	No
JC virtual machine	Yes	No	No	No	No	No	No	No	No	No	No
Jikes RVM	Yes[41]	No[41]	No[41]	No[41]	No[41]	Yes[41]	No[41]	No[41]	No[41]	No[41]	No[41]
Kaffe	Yes[12]	Yes[12]	Yes[12]	Yes[12]	Yes[12]	Yes[12]	Yes[12]	Yes[12]	Yes[12]	No[12]	Yes[12]
Mysaifu JVM	No[42]	No[42]	No[42]	No[42]	No[42]	No[42]	Yes[42]	No[42]	No[42]	No[42]	No[42]
SableVM	Yes[43]	Yes	Yes[43]	Yes	Yes[43]	Yes[43]	Yes[43]	Yes[43]	Yes[43]	No[43]	Yes[43]

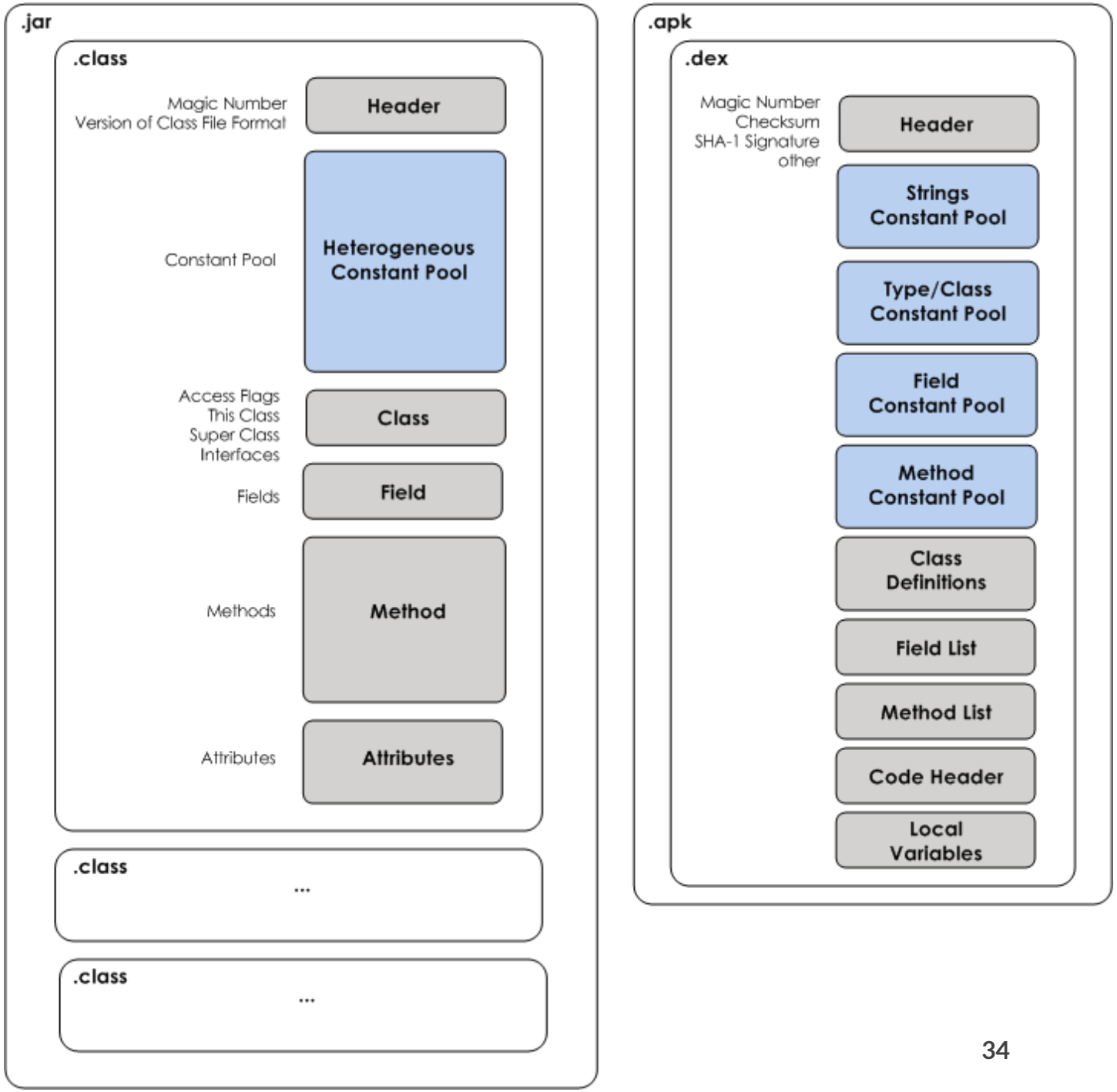
Podporované operačné systémy

JVM	Windows	Linux	FreeBSD	NetBSD	OpenBSD	Solaris	OpenSolaris	Darwin	macOS	iOS	BeOS	Haiku	AIX	IRIX	OS/2	Windows Mobile	AmigaOS
GCJ	Yes[33]	Yes[33]	Yes[33]	?	?	Yes[33]	?	Yes[33]	Yes[33]	?	?	?	?	Yes[33]	?	No[33]	No[33]
HotSpot, OpenJDK edition	Yes[44] (distributed as source code)	Yes	Port[45]	Port[45]	Port[45]	Yes	Yes[46]	?	Yes[47]	No	No	Port[48]	Port[37]	No	Port[49]	No	No
HotSpot, Oracle JDK edition	Yes[34]	Yes[34]	No	No	No	Yes[34]	?	?	Yes[34]	No	No	No	No	No	No	No	No
HotSpot, Java SE embedded edition	Yes[4]	Yes[4]	No[4]	No[4]	No[4]	Yes[4]	?[4]	No[4]	No[4]	No[4]	No[4]	No[4]	No[4]	No[4]	No[4]	No[4]	No[4]
HotSpot, Zero port	No[39][50]	Yes[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]	No[39][50]
IKVM.NET	Any operating system with a .NET framework																
JAmiga	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes
JamVM	No[20]	Yes[20]	Yes[20]	No[20]	Yes[20]	Yes[20]	Yes[20]	Yes[20]	Yes[20]	On Jailbroken iPhone[20]	No[20]	No[20]	No[20]	No[20]	No[20]	No[20]	No[20]

Porovnanie VM

Java VM	Dalvik VM
Stack Based	**Register Based**
Executes .class file	Executes .dex file
In the case of the .class file, each class has its own private, heterogeneous constant pool.	It uses shared, type-specific constant pools as it's primary mechanism for conserving memory. Rather than store these values throughout the class, they are always referred to by their index in the constant pool.

Porovnanie jar a apk



Zásobník (Stack) vs. Registre

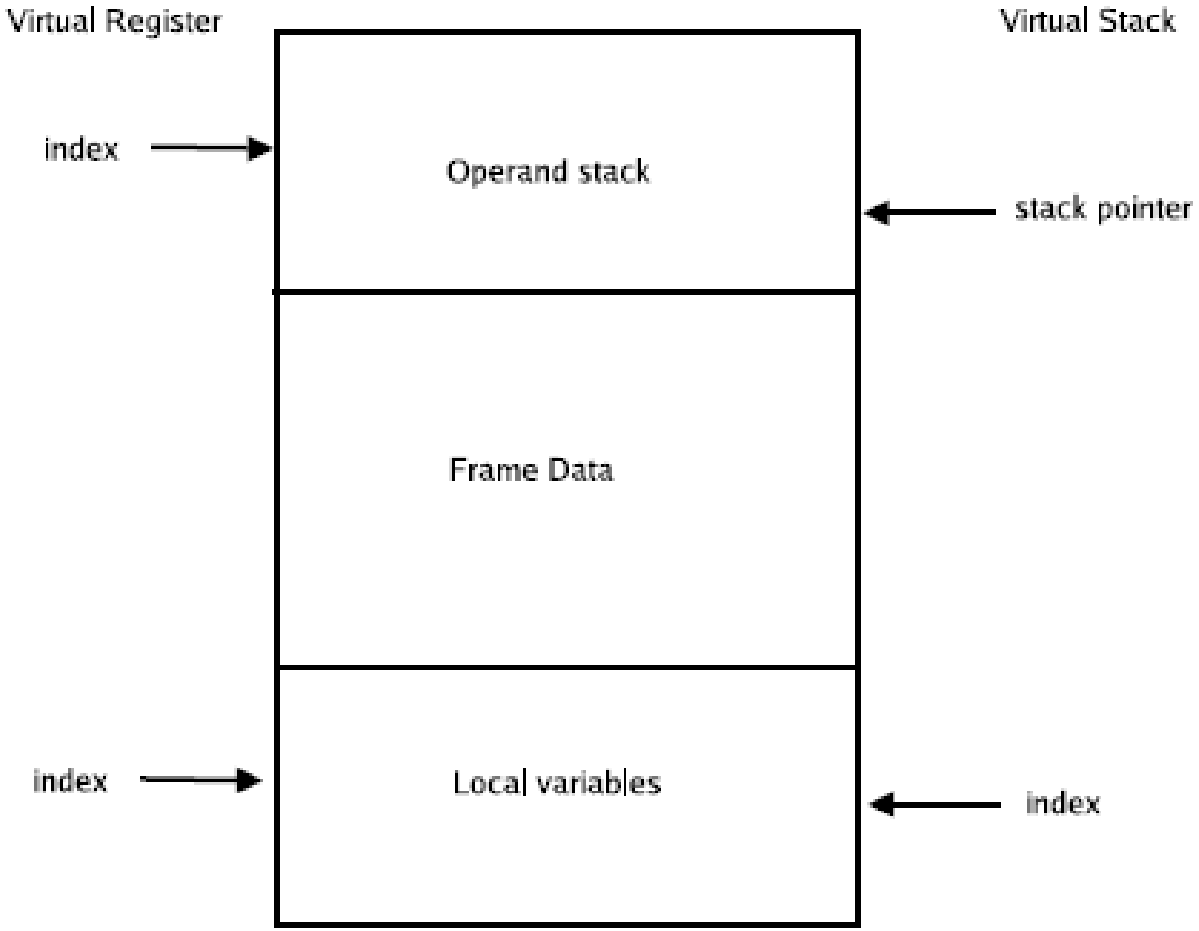
- Réžia na vykonanie inštrukcie interpretera VM pozostáva z 3 komponentov:

1. Odoslanie/dispečing inštrukcie
2. Prístup k operandom
3. Prevedenie výpočtu

- `Príklad:  A = B + C`

- **`ILOAD`** `c,` **`ILOAD`** `b,` **`IADD`**, **`ISTORE`** `a` **`(Stack JVM)`**
- **`IADD`** `a, b, c` **`(Virtual Register Machine)`**

Preklad zásobníka do registra



Stack-based bytecode	Register-based bytecode
iload_1	move r10, r1
iload_2	move r11, r2
iadd	iadd r10, r10, r11
istore_3	move r3, r10

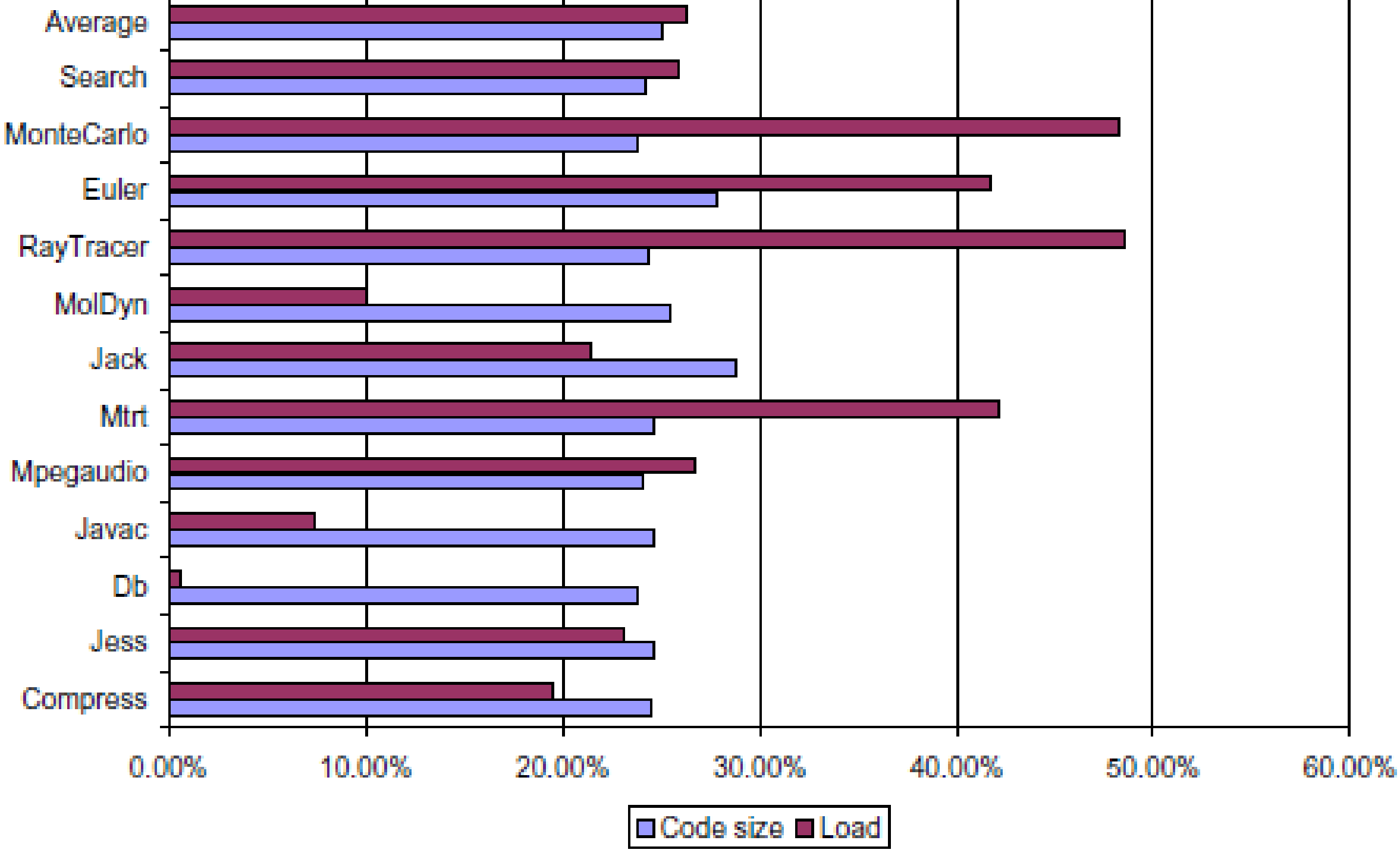
Príklad copy propagation

Inštrukcia po preklade pre JVM

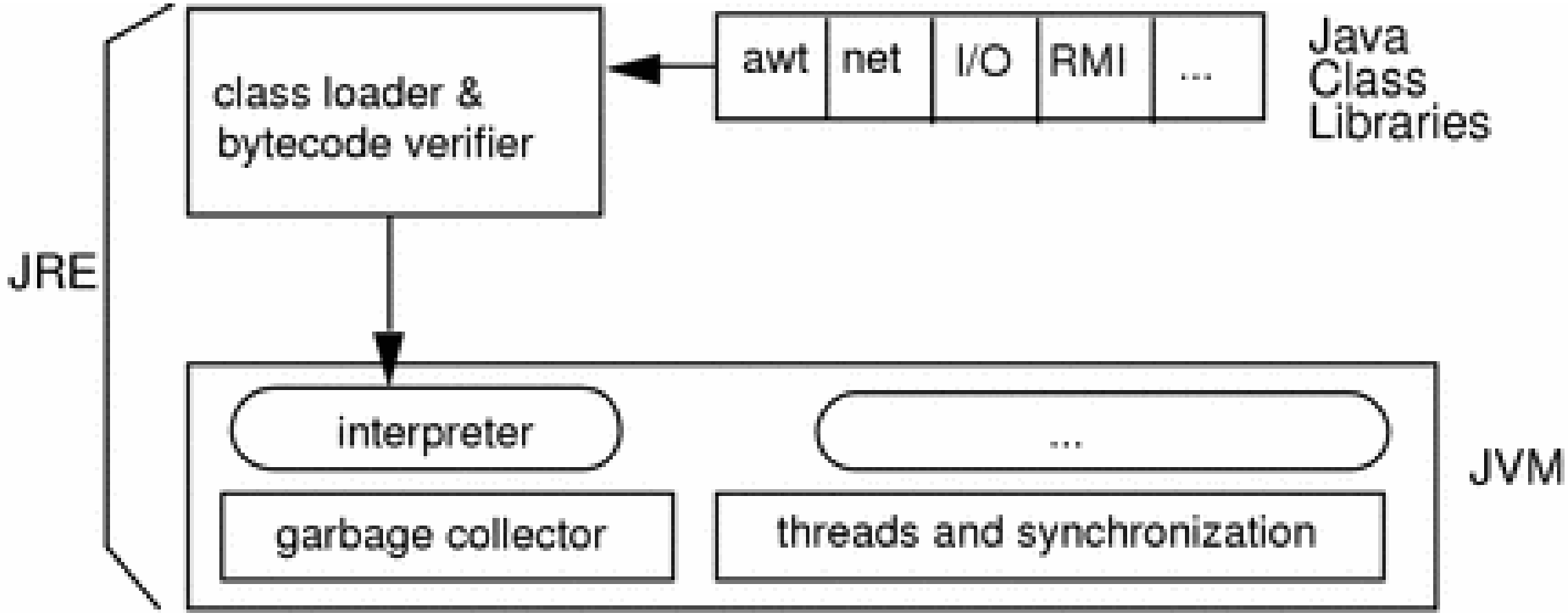
```
1. move r10, r1          //iload_1
2. move r11, r2          //iload_2
3. iadd r10, r10, r11    //iadd
4. move r3, r10          //istore_3
```

Po šírení kópie dopredu a dozadu

```
3. iadd r3, r1, r2
```

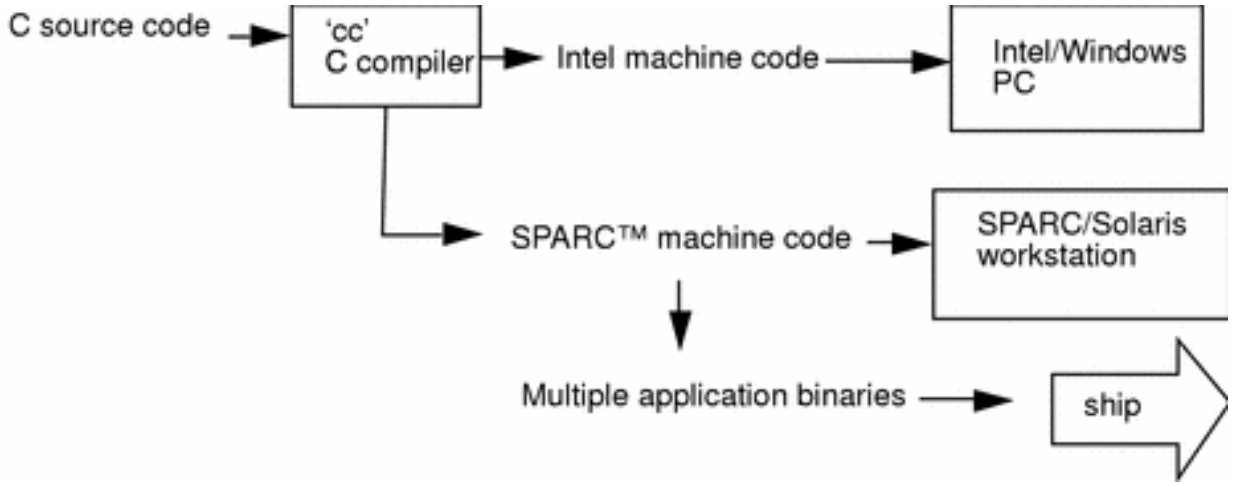

Average
Search
MonteCarlo
Euler
RayTracer
MolDyn
Jack
Mtrt
Mpegaudio
Javac
Db
Jess
Compress
0.00%
10.00%
20.00%
30.00%
40.00%
50.00%
60.00%
Code size
Load

Typická implementácia JVM

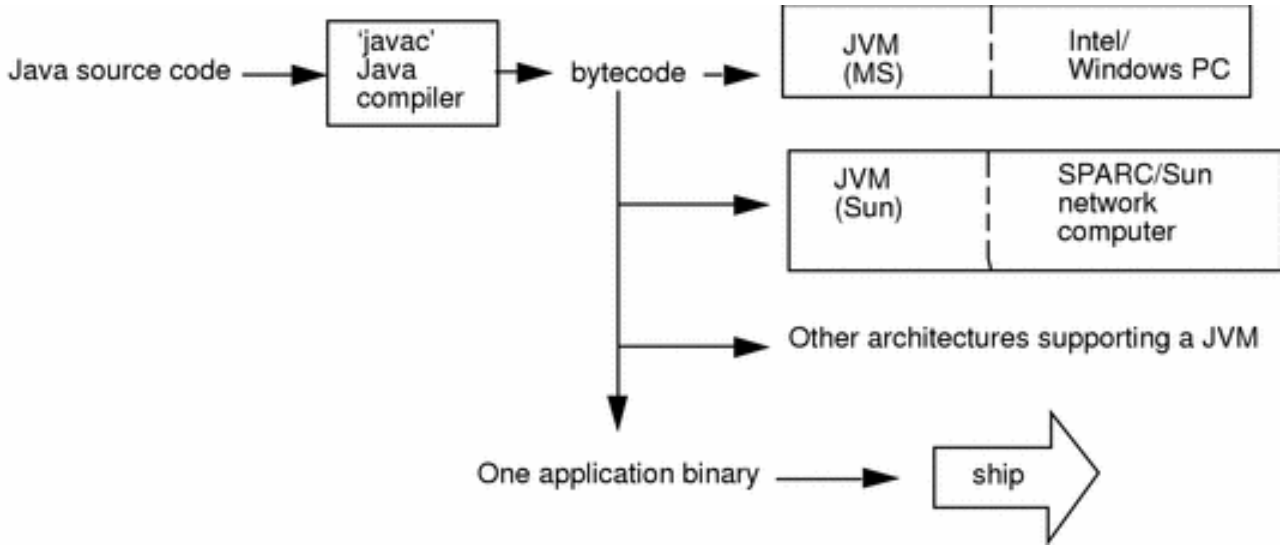


JVM a kompilovanie

Traditional Compile-Time Environment

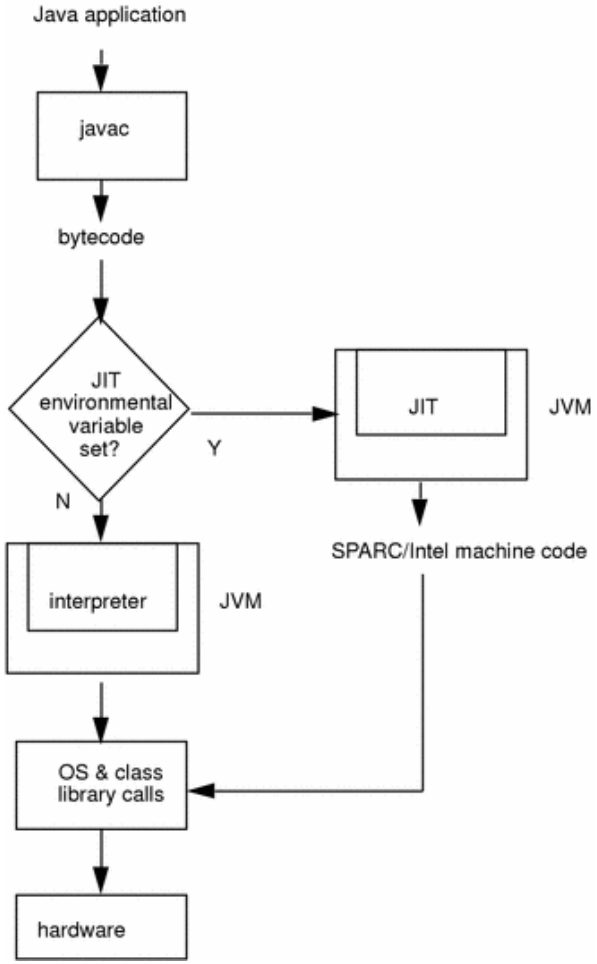


Java Compile-Time Environment

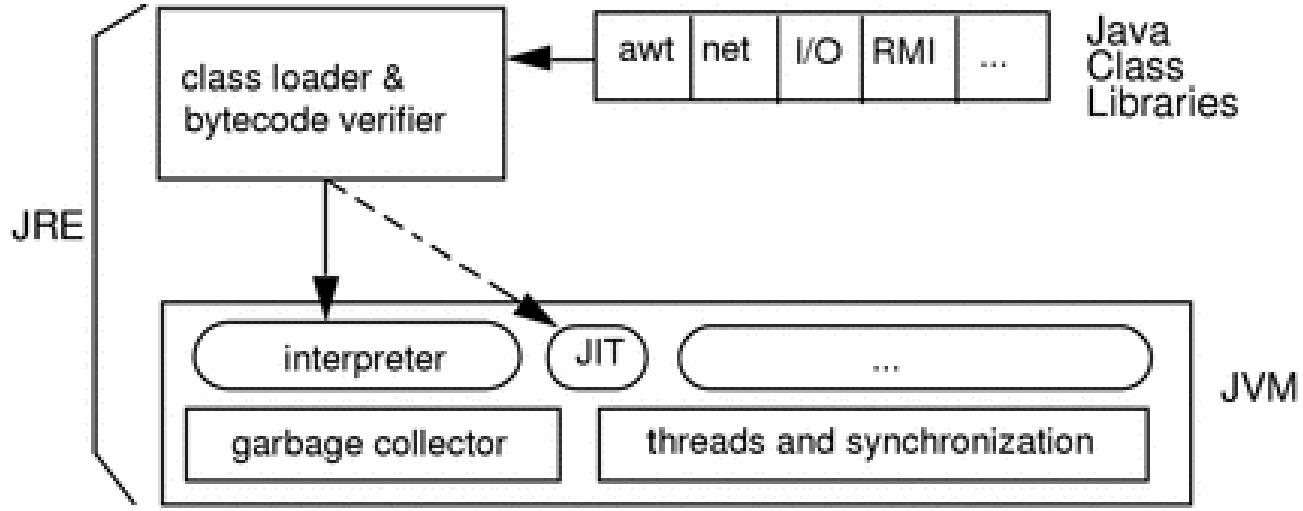


Just-In-Time (JIT) Compiler

JIT Compile Process



JVM Relationship to JIT Compiler

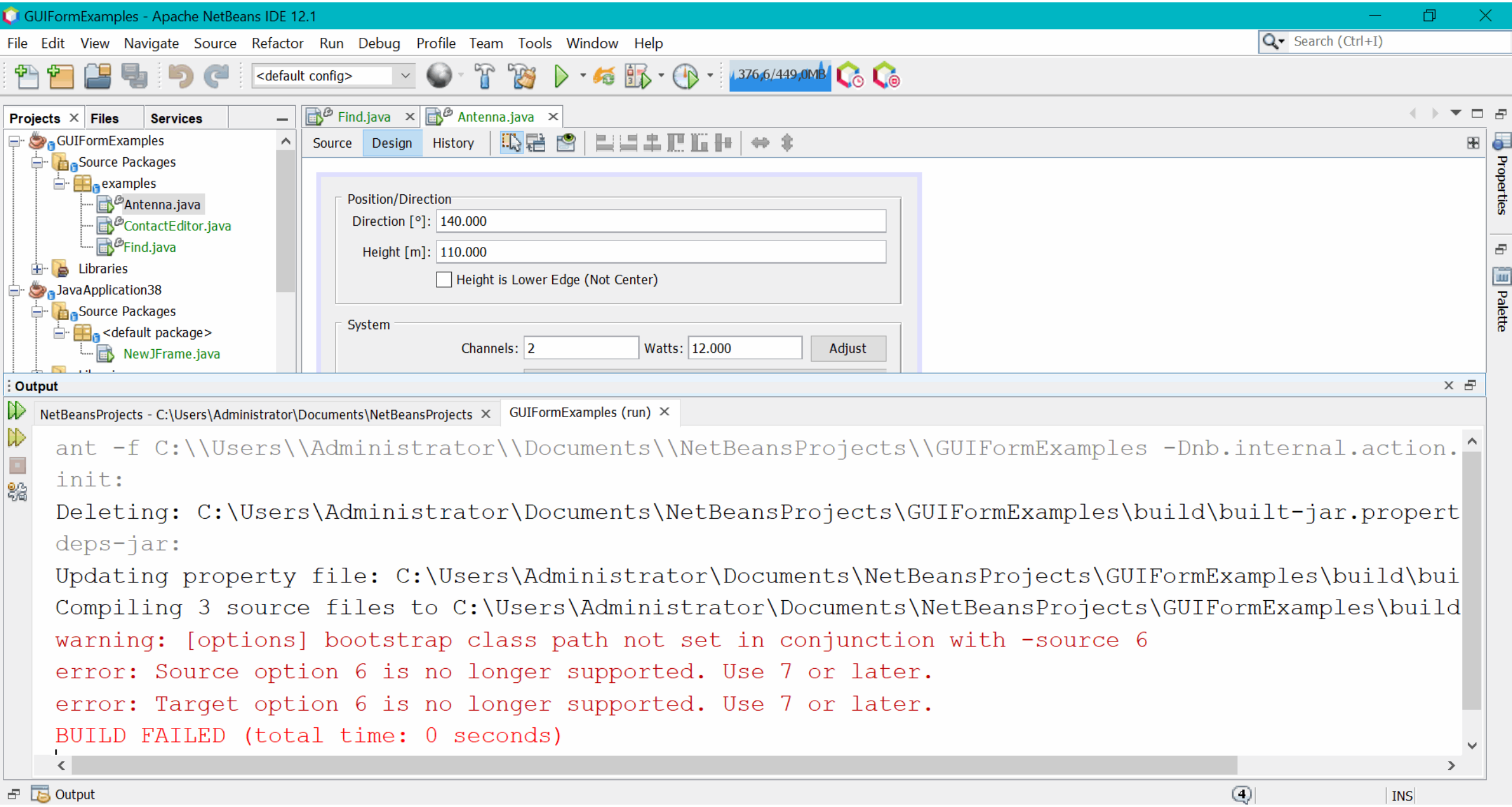


Administrator: C:\Windows\System32\cmd.exe

```
C:\Windows\system32>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                  Read options and filenames from file
  -Akey[=value]                Options to pass to annotation processors
  --add-modules <module>(,<module>)*
        Root modules to resolve in addition to the initial modules, or all modules
        on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
        Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
        Specify where to find user class files and annotation processors
  -d <directory>               Specify where to place generated class files
  -deprecation
        Output source locations where deprecated APIs are used
  --enable-preview
        Enable preview language features. To be used in conjunction with either -source or --release.
  -encoding <encoding>         Specify character encoding used by source files
  -endorseddirs <dirs>         Override location of endorsed standards path
  -extdirs <dirs>              Override location of installed extensions
  -g                           Generate all debugging info
  -g:{lines,vars,source}       Generate only some debugging info
  -g:none                      Generate no debugging info
  -h <directory>
        Specify where to place generated native header files
  --help, -help, -?            Print this help message
  --help-extra, -X             Print help on extra options
  -implicit:{none,class}
        Specify whether or not to generate class files for implicitly referenced files
  -J<flag>                     Pass <flag> directly to the runtime system
  --limit-modules <module>(,<module>)*
        Limit the universe of observable modules
  --module <module>(,<module>)*, -m <module>(,<module>)*
```

Administrator: C:\Windows\System32\cmd.exe

```
      Specify where to find application modules
--module-source-path <module-source-path>
      Specify where to find input source files for multiple modules
--module-version <version>
      Specify version of modules that are being compiled
-nowarn                     Generate no warnings
-parameters
      Generate metadata for reflection on method parameters
-proc:{none,only}
      Control whether annotation processing and/or compilation is done.
-processor <class1>[,<class2>,<class3>...]
      Names of the annotation processors to run; bypasses default discovery process
--processor-module-path <path>
      Specify a module path where to find annotation processors
--processor-path <path>, -processorpath <path>
      Specify where to find annotation processors
-profile <profile>
      Check that API used is available in the specified profile
--release <release>
      Compile for the specified Java SE release. Supported releases: 7, 8, 9, 10, 11, 12, 13, 14, 15
-s <directory>              Specify where to place generated source files
--source <release>, -source <release>
      Provide source compatibility with the specified Java SE release. Supported releases: 7, 8, 9, 10, 11, 12, 13, 14, 15
--source-path <path>, -sourcepath <path>
      Specify where to find input source files
--system <jdk>|none         Override location of system modules
--target <release>, -target <release>
      Generate class files suitable for the specified Java SE release. Supported releases: 7, 8, 9, 10, 11, 12, 13, 14, 15
--upgrade-module-path <path>
      Override location of upgradeable modules
-verbose                    Output messages about what the compiler is doing
--version, -version         Version information
-Werror                     Terminate compilation if warnings occur
```


GUIFormExamples - Apache NetBeans IDE 12.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
376,6/449,0MB
Projects
Files
Services
GUIFormExamples
Source Packages
examples
Antenna.java
ContactEditor.java
Find.java
Libraries
JavaApplication38
Source Packages
<default package>
NewJFrame.java
Find.java
Antenna.java
Source
Design
History
Position/Direction
Direction [°]: 140.000
Height [m]: 110.000
Height is Lower Edge (Not Center)
System
Channels: 2
Watts: 12.000
Adjust
Properties
Palette
Output
NetBeansProjects - C:\Users\Administrator\Documents\NetBeansProjects
GUIFormExamples (run)
ant -f C:\\Users\\Administrator\\Documents\\NetBeansProjects\\GUIFormExamples -Dnb.internal.action.
init:
Deleting: C:\Users\Administrator\Documents\NetBeansProjects\GUIFormExamples\build\built-jar.propert
deps-jar:
Updating property file: C:\Users\Administrator\Documents\NetBeansProjects\GUIFormExamples\build\bui
Compiling 3 source files to C:\Users\Administrator\Documents\NetBeansProjects\GUIFormExamples\build
warning: [options] bootstrap class path not set in conjunction with -source 6
error: Source option 6 is no longer supported. Use 7 or later.
error: Target option 6 is no longer supported. Use 7 or later.
BUILD FAILED (total time: 0 seconds)
Output
INS

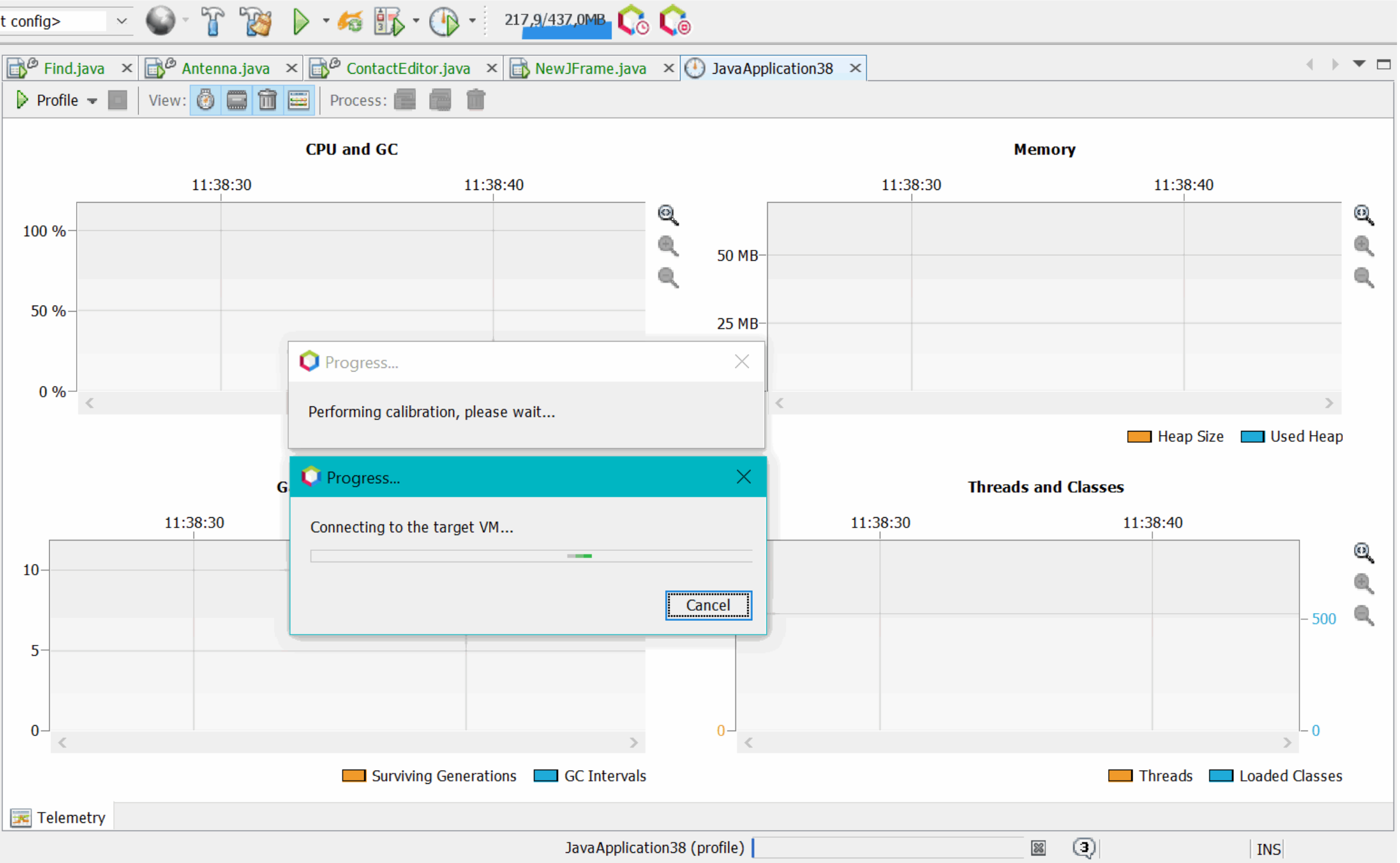
Administrator: C:\Windows\System32\cmd.exe

```
C:\Windows\system32>java
Usage: java [-options] class [args...]
           (to execute a class)
   or  java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
    -d32          use a 32-bit data model if available
    -d64          use a 64-bit data model if available
    -server       to select the "server" VM
                  The default VM is server.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  A ; separated list of directories, JAR archives,
                  and ZIP archives to search for class files.
    -D<name>=<value>
                  set a system property
    -verbose:[class|gc|jni]
                  enable verbose output
    -version      print product version and exit
    -version:<value>
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  require the specified version to run
    -showversion  print product version and continue
    -jre-restrict-search | -no-jre-restrict-search
                  Warning: this feature is deprecated and will be removed
                  in a future release.
                  include/exclude user private JREs in the version search
    -? -help      print this help message
    -X            print help on non-standard options
    -ea[:<packagename>...|:<classname>]
    -enableassertions[:<packagename>...|:<classname>]
```

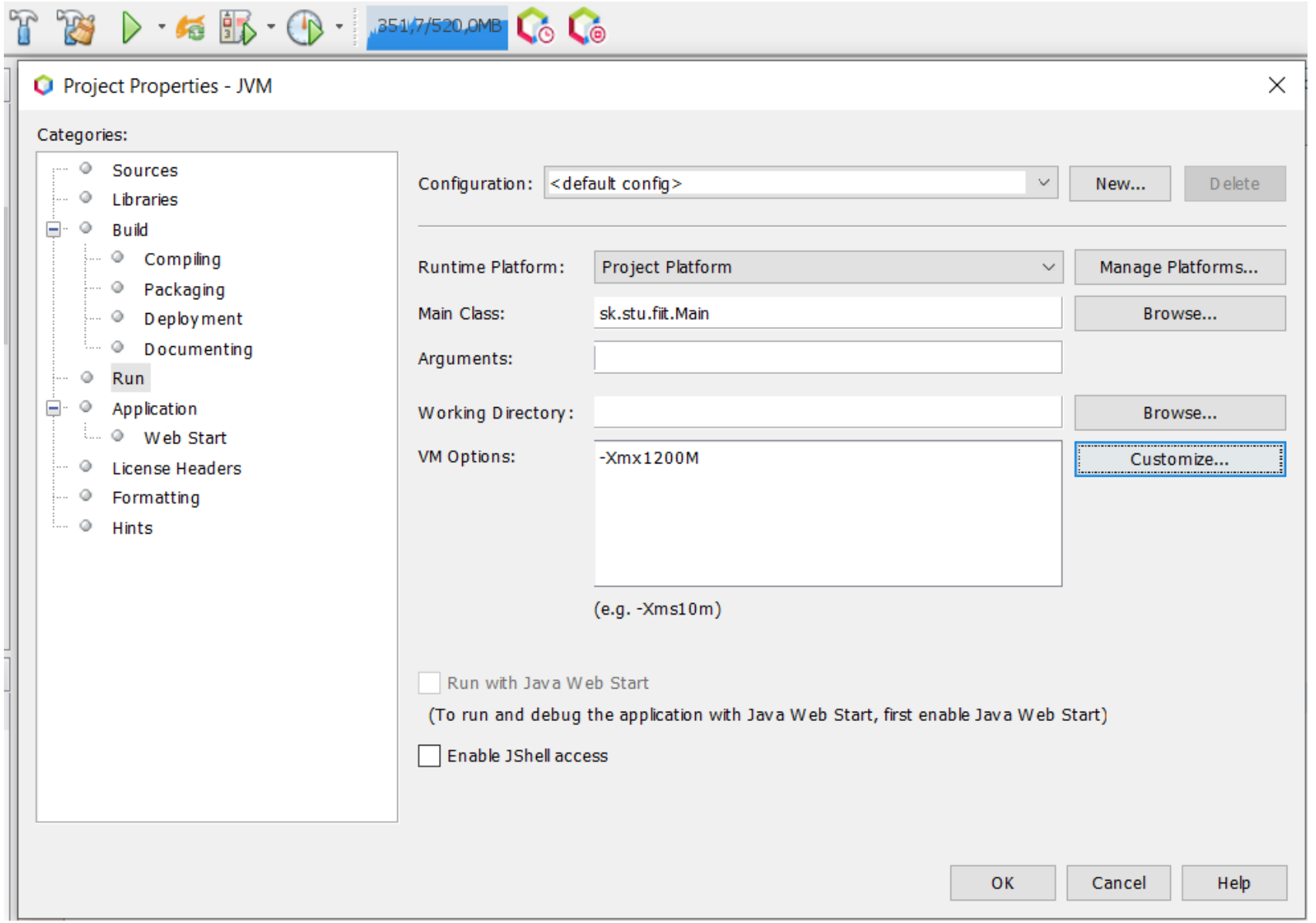
Output

NetBeansProjects - C:\Users\Administrator\Documents\NetBeansProjects | JavaApplication38 (profile)

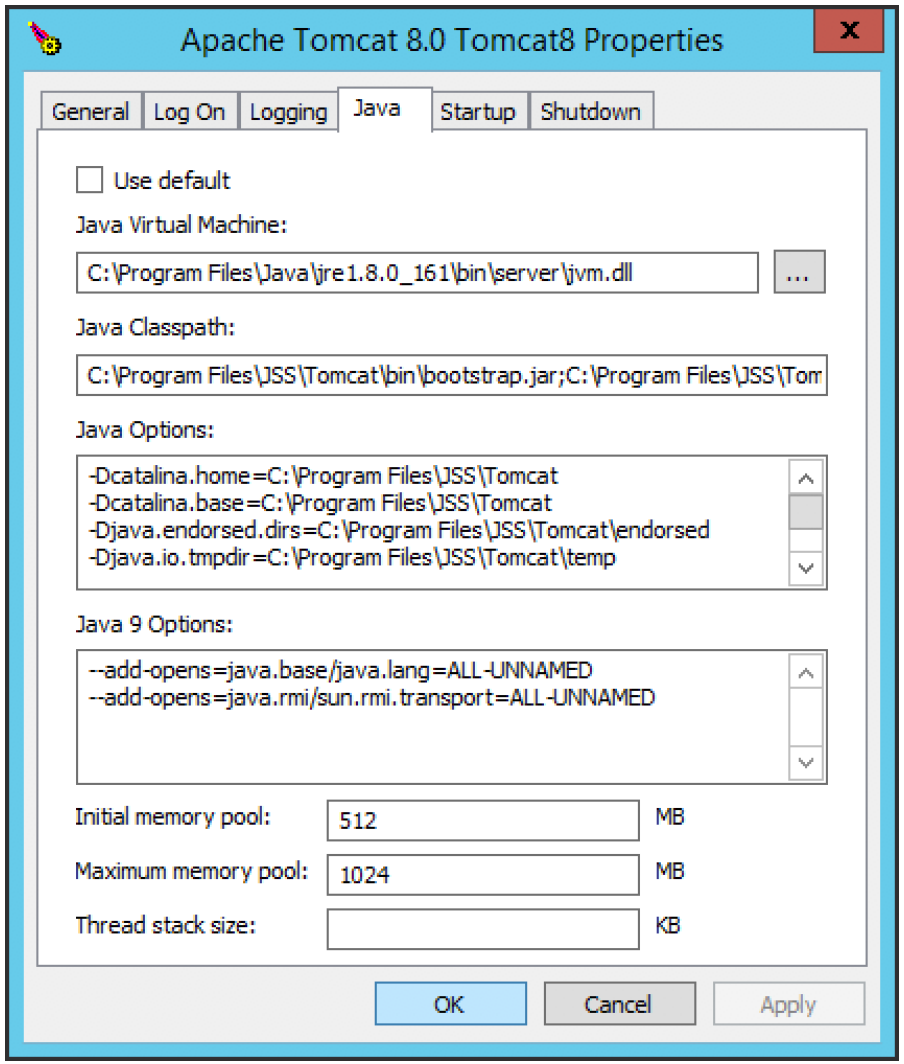
```
ant -f C:\\Users\\Administrator\\Documents\\NetBeansProjects\\JavaApplication38 -Dnb.internal.action.name=profile -Drun.class=NewJFrame "-Drun.jvmargs.ide= -agentpath:\"C:/Program Files/NetBeans-12.1/netbeans/profiler/lib/deployed/jdk16/windows-amd64/profilerinterface.dll\"=\"C:\\Program Files\\NetBeans-12.1\\netbeans\\profiler\\lib\",5140,10 -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=C:\\Users\\Administrator\\Documents\\NetBeansProjects\\JavaApplication38\\nbproject\\private\\profiler " profile
init:
profile-init:
Deleting: C:\Users\Administrator\Documents\NetBeansProjects\JavaApplication38\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\Administrator\Documents\NetBeansProjects\JavaApplication38\build\built-jar.properties
compile:
profile:
*** Profiler message (Sat May 08 11:39:08 CEST 2021): Starting target application...
C:\Program Files\Java\jdk-14.0.2\bin\java.exe -agentpath:C:/Program Files/NetBeans-12.1/netbeans/profiler/lib/deployed/jdk16/windows-amd64/profilerinterface.dll -Xbootclasspath/a:C:\Program Files\NetBeans-12.1\netbeans\profiler\lib\jfluid-server.jar;C:\Program Files\NetBeans-12.1\netbeans\profiler\lib\jfluid-server-15.jar org.netbeans.lib.profiler.server.ProfilerServer C:/Program Files/NetBeans-12.1/netbeans/profiler/lib/deployed/jdk16/windows-amd64 5141 10 ____Profiler+Calibration+Run____
*** Profiler error (Sat May 08 11:41:20 CEST 2021): connection with server not open
C:\Users\Administrator\Documents\NetBeansProjects\JavaApplication38\nbproject\build-impl.xml:1459    : User abor
```

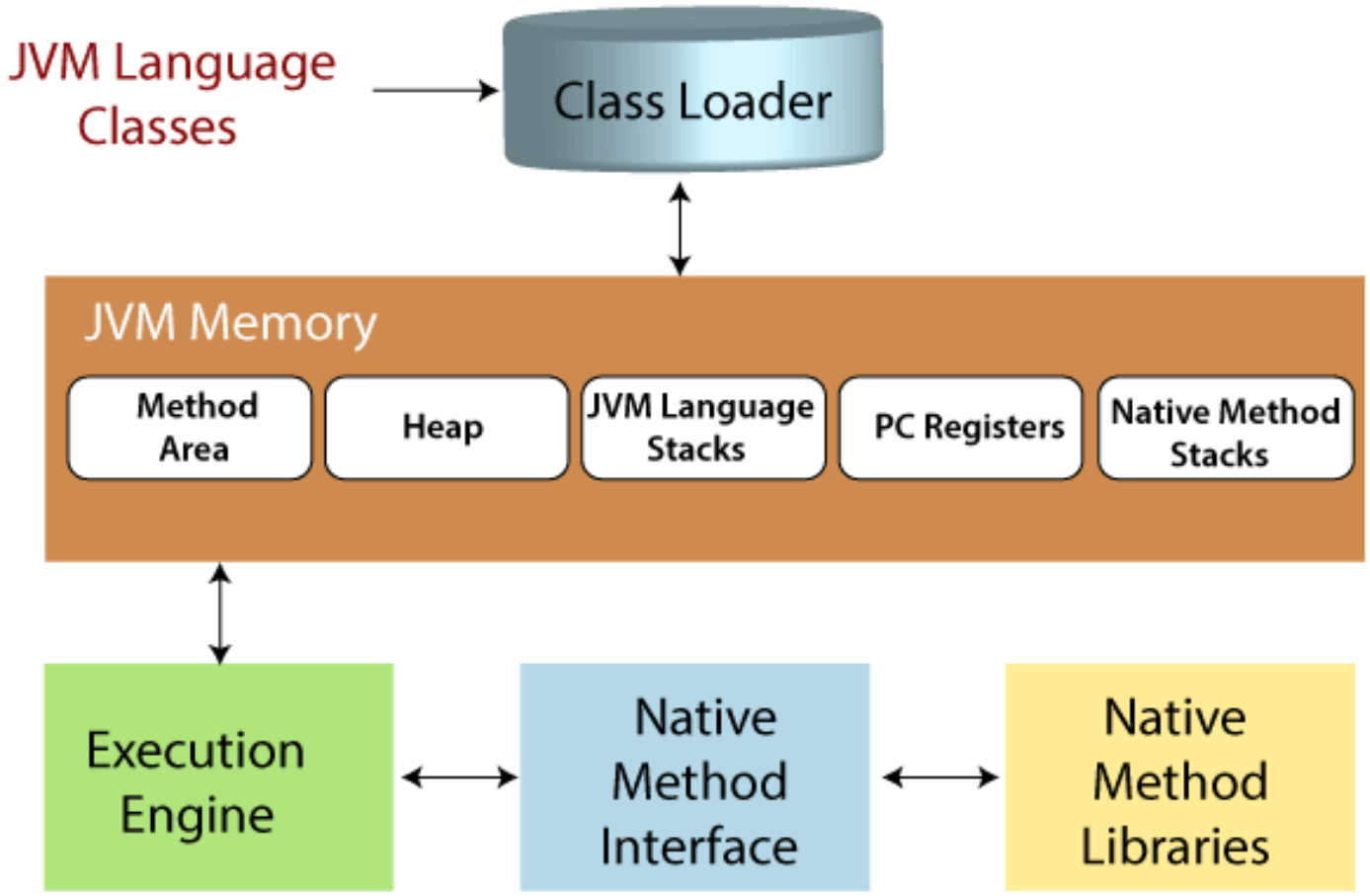

t config>
217,9/437,0MB
Find.java
Antenna.java
ContactEditor.java
NewJFrame.java
JavaApplication38
Profile
View:
Process:
CPU and GC
11:38:30
11:38:40
100 %
50 %
0 %
Memory
50 MB
25 MB
Heap Size
Used Heap
Progress...
Performing calibration, please wait...
Progress...
Connecting to the target VM...
Cancel
Threads and Classes
10
5
0
500
Surviving Generations
GC Intervals
Threads
Loaded Classes
Telemetry
JavaApplication38 (profile)
INS

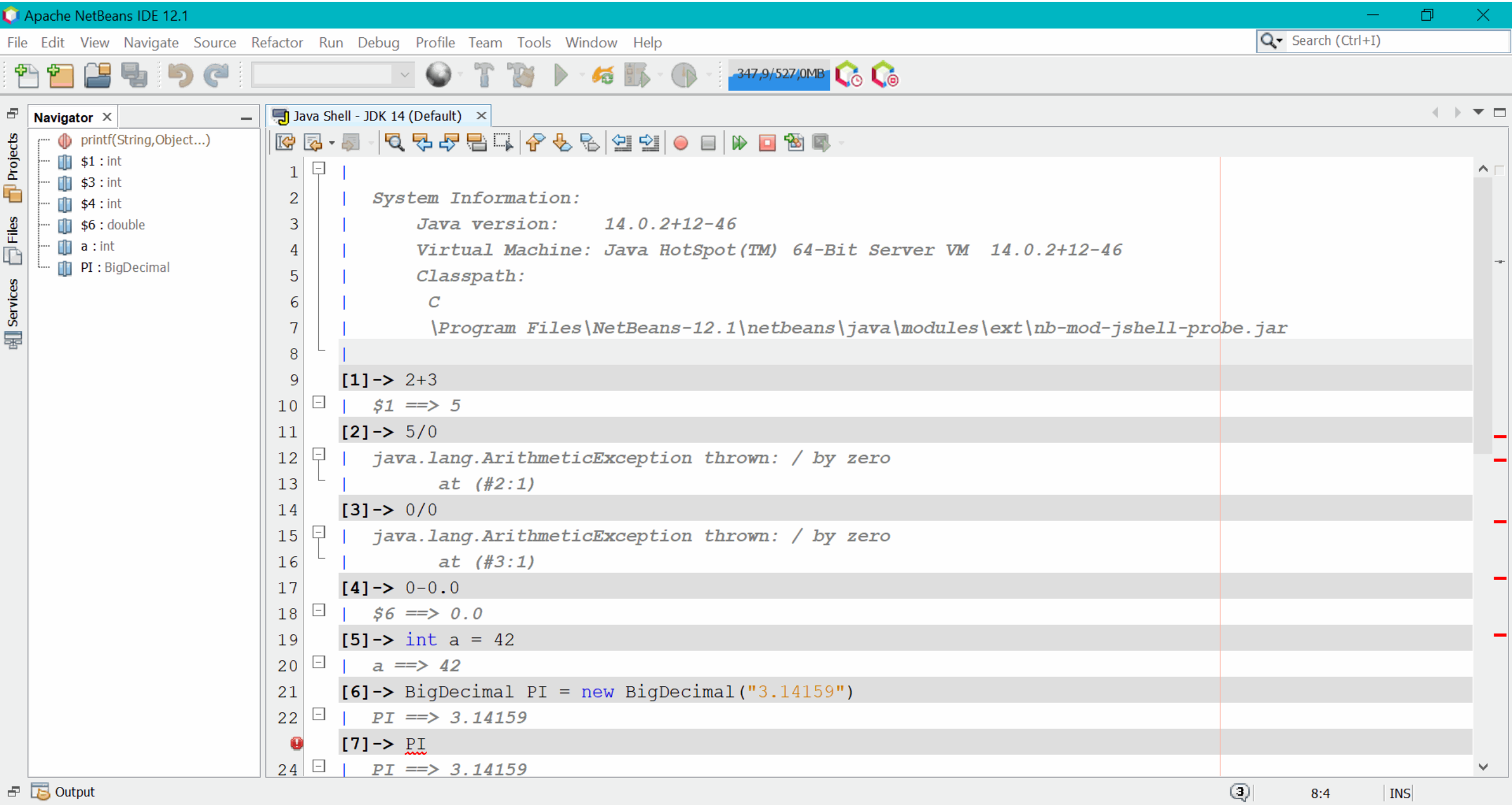
Vytvorenie vlastných configov



Nastavenie JVM pre Webové servery






Apache NetBeans IDE 12.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
347,9/527,0MB
Navigator
printf(String,Object...)
$1 : int
$3 : int
$4 : int
$6 : double
a : int
PI : BigDecimal
Java Shell - JDK 14 (Default)
| System Information:
| Java version: 14.0.2+12-46
| Virtual Machine: Java HotSpot(TM) 64-Bit Server VM 14.0.2+12-46
| Classpath:
| C
| \Program Files\NetBeans-12.1\netbeans\java\modules\ext\nb-mod-jshell-probe.jar
[1]-> 2+3
| $1 ==> 5
[2]-> 5/0
| java.lang.ArithmeticException thrown: / by zero
| at (#2:1)
[3]-> 0/0
| java.lang.ArithmeticException thrown: / by zero
| at (#3:1)
[4]-> 0-0.0
| $6 ==> 0.0
[5]-> int a = 42
| a ==> 42
[6]-> BigDecimal PI = new BigDecimal("3.14159")
| PI ==> 3.14159
[7]-> PI
| PI ==> 3.14159
Output
8:4
INS

JVM argumenty (parametre)

Standard Options for Java

- Voľby garantované a podporované všetkými implementáciami JVM
- Nastavenie modulov (Java9 a vyššie)
- Testovanie verzie JVM
- Nastavenie classpath
- Pomocné výstupy
- Systémové premenné JVM

Extra Options for Java

- Začínajú -X
- Nemusia byť podporované všetkými implementáciami JVM, alebo sa môžu meniť
- Pokročilé možnosti pre vývojárov používané pre vyladenie špecifických oblastí behu JVM začínajú -XX

Ďalšie JVM argumenty (parametre)

Pokročilé možnosti

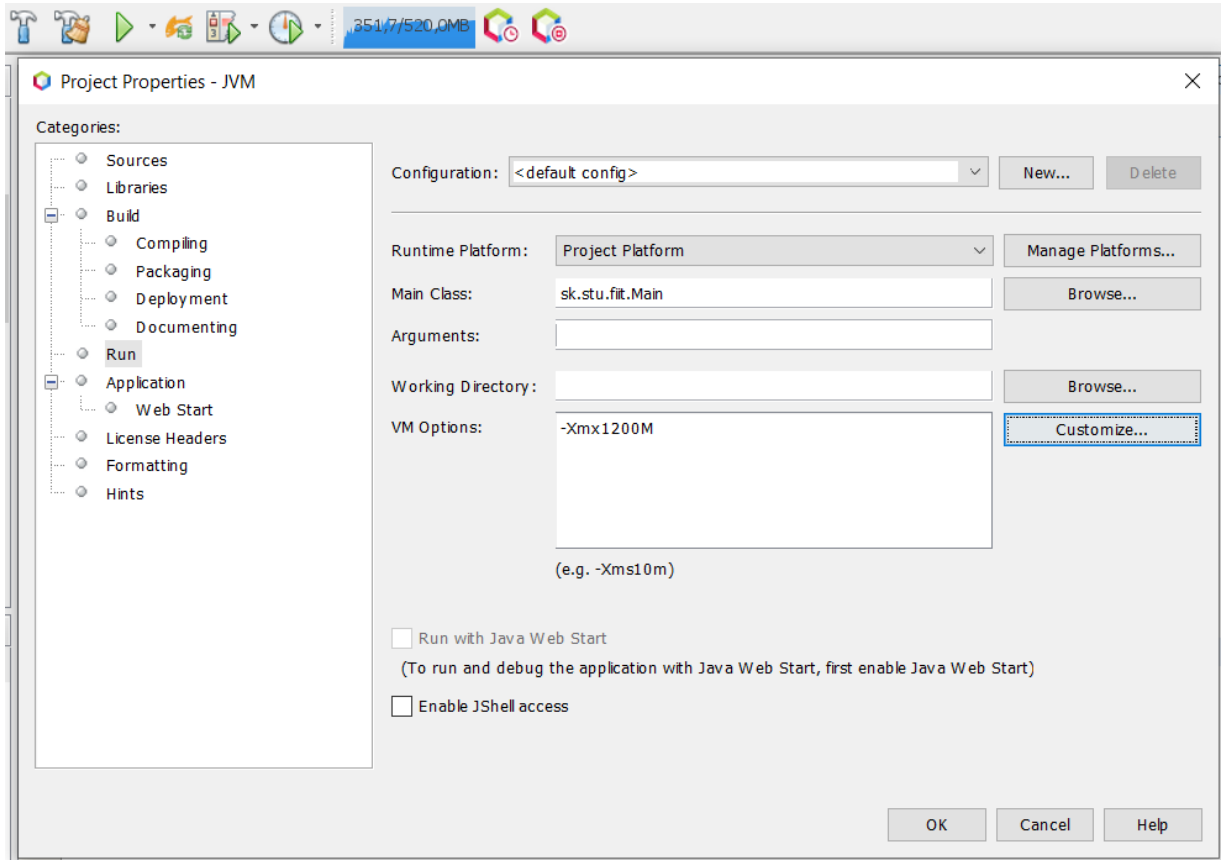
- **Advanced Runtime Options for Java** - Riadi behové správanie JVM
- **Advanced JIT Compiler Options for Java** - Umožňuje pokročilé nastavenie kompilátora
- **Advanced Serviceability Options for Java** - Zhromažďuje systémové informácie a vykonáva rozsiahle ladenie
- **Advanced Garbage Collection Options for Java** - Umožňuje nastaviť správanie garbage collector, pravidlá pre čistenie haldy

Staršie nastavenia

- **Deprecated Java Options** - Funkčné, ale neodporúčané, takže sa vykonajú s Warning
- **Obsolete Java Options** - Funkčné, ale ignorované, nevykonajú sa, vyvolajú warning
- **Removed Java Options** - Odstránené, nevykonajú sa a spôsobí Error

Veľkosť haldy

- **Maximálna veľkosť haldy**
  - `java -Xmx1200M -jar Program.jar`
- **Predvolená veľkosť haldy**
  - `java -Xms512M -jar Program.jar`



New JDK 7 Feature: Support for Dynamically Typed Languages in the Java Virtual Machine

By Ed Ort, July 2009

DaVinci Helicopter - This article describes a new feature provided in JDK 7: support for dynamically typed languages in the Java virtual machine (JVM). This feature, which implements JSR 292: Supporting Dynamically Typed Languages on the Java Platform, is the logical follow-on to JSR 223: Scripting for the Java Platform. Support for JSR 223 was provided as part of Java SE 6 and implemented in JDK 6.

With the addition of support for JSR 292 in JDK 7, dynamically typed languages should run faster in the JVM than they do today. A key part of this support is the addition of a new Java bytecode, `invokedynamic`, for method invocation, and an accompanying linkage mechanism that involves a new construct called a method handle. These features enable implementers of compilers for dynamically typed languages, that is, the people who develop compilers for languages such as JRuby and Jython, to generate bytecode that runs extremely fast in the JVM.

Support for JSR 292 in JDK 7 will enable developers of compilers for dynamically typed languages to generate bytecode that runs extremely fast in the JVM.

This should increase the variety and quality of dynamically typed languages that run in the JVM. Application developers should see more of their favorite dynamically typed languages available in the Java ecosystem. Also, these features should boost the performance of code generated by dynamically typed language compilers that already run in the JVM. For example, the JRuby compiler generates bytecode that performs well in the JVM, but the JRuby bytecode will run even faster when the JRuby compiler is modified to use the `invokedynamic` bytecode and method handles.

The JSR 292 Expert Group is also investigating support for interface injection, the ability to modify classes at runtime so that they can implement new interfaces -- this is a feature that is c

Administrator: Príkazový riadok - jshell

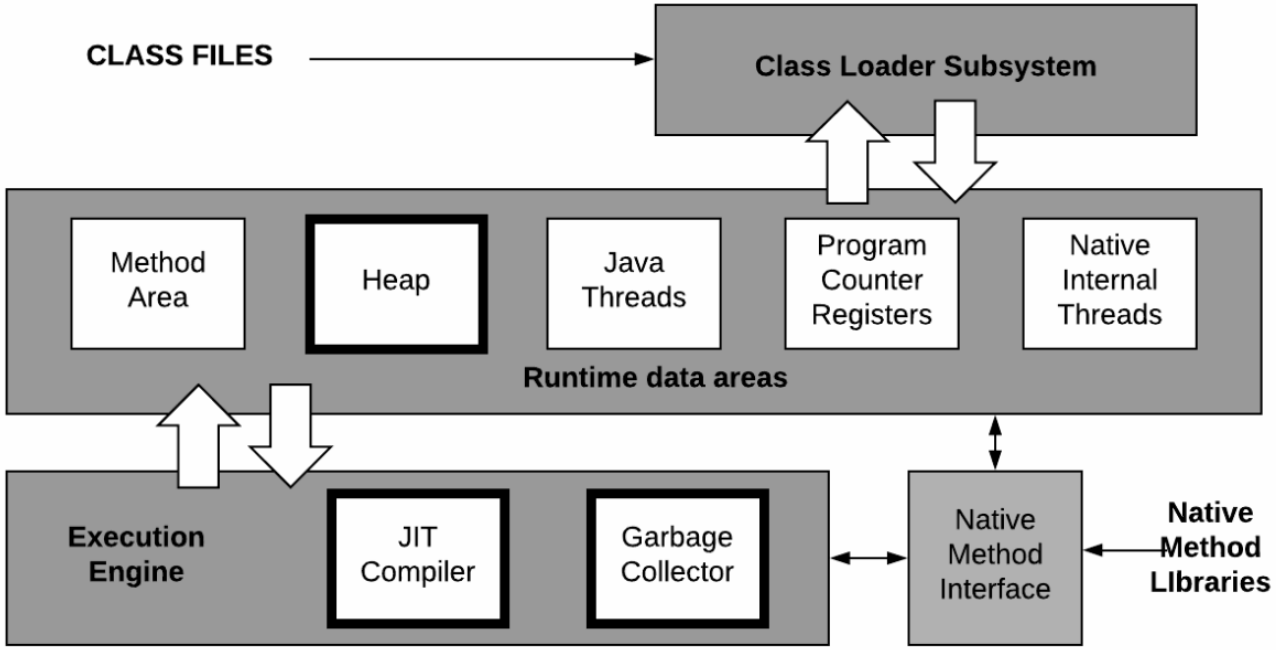
```
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. Všetky práva vyhradené.

C:\WINDOWS\system32>jshell
|  Welcome to JShell -- Version 14.0.2
|  For an introduction type: /help intro

jshell> /help intro
|
|                                intro
|                                =====
|
|  The jshell tool allows you to execute Java code, getting immediate results.
|  You can enter a Java definition (variable, method, class, etc), like:  int x = 8
|  or a Java expression, like:  x + x
|  or a Java statement or import.
|  These little chunks of Java code are called 'snippets'.
|
|  There are also the jshell tool commands that allow you to understand and
|  control what you are doing, like:  /list
|
|  For a list of commands: /help

jshell> /vars

jshell>
```

Java Developers never RIP,
they just get Garbage Collected.

- ilike.nitty-witty.com

Garbage collectors in JDK

- Serial Garbage Collector
- Parallel Garbage Collectors in several versions
- Concurrent Mark Sweep (CMS) Collector
- Garbage First (G1) Garbage Collector



```
java -Xms1G -Xmx1G -jar dacapo-9.12-bach.jar
h2 -n 5 --no-pre-iteration-gc
```

The example of the command that would execute 5 iterations of *h2* application with limited memory heap to 1 GB, and no system GC between iterations.

Garbage Collector

Getting Started with the G1 Garbage Collector

- Overview
- Java Technology and the JVM
- The G1 Garbage Collector
- Reviewing GC with the CMS
- The G1 Garbage Collector Step by Step
- Command Line Options and Best Practices
- Logging GC with G1
- Summary

Čo ti odporúčam si pozrieť?

1. https://docs.oracle.com/javase/specs/
2. https://blogs.oracle.com/javamagazine/java-runtime-encapsulation-internals?source=:em:nw:mt::::RC_WWMK200429P00043:NSL400155430
3. https://www.javatpoint.com/jvm-java-virtual-machine
4. https://www.oracle.com/technetwork/tutorials/tutorials-1876574.html
5. https://www.researchgate.net/publication/326701102_Comparison_of_garbage_collectors_in_Java_programming_language/figures?lo=1
6. https://docs.oracle.com/javase/10/jshell/snippets.htm#JSHEL-GUID-E24D9E84-C9E3-43DE-B14D-C9544A35E5FC
7. https://www.oracle.com/technical-resources/articles/javase/dyntypelang.html

Šup do záložiek