

Vývoj Aplikácií s Viacvrstvovou Architektúrou

9. Modularita a Modulárne Aplikácie



Čo nás čaká a neminie...

1. časť

Úvod do Javy

Štruktúra platformy

Vývojové technológie

Kolekcie

Logovanie

Lokalizácia

2. časť

NIO.2, IO, XML

Regulárne výrazy

Modularita

JDBC

Bezpečnosť

Prehľad JEE a .NET

Modulárny návrh (Modular design)

- Modulárny dizajn alebo modularita v dizajne je princíp dizajnu, ktorý **rozdeľuje systém na menšie časti** nazývané **moduly** (ako sú modulárne procesné lyžiny), ktoré **možno nezávisle vytvárať, upravovať, nahradzovať alebo vymieňať s inými modulmi alebo medzi rôznymi systémami.**

Kde všade je modulárny návrh?

- Autá, výťahy, nábytok, tkáčske stavy
- Solárne panely, veterné turbíny
- Železničné signalizačné systémy, telefónne ústredne
- Modulárne budovy, potrubné systémy, rozvody elekt. energie
- Počítače HW (USB), **informačné systémy**, procesné systémy

Kde všade je modulárny návrh v IT?

- Servisne orientovaná architektúra (SOA)
- Mikroslužby (MSA/MSOA)
- Rozšírenia a doplnky (Add-ons / Extensions / Widgets)
- Moduly a knižnice
- Trh mobilných aplikácií



Modulárne dokumentové systémy

EDM/CA - Electronic Document Management System/Contract Administration

EBIS - Electronic Buying Information System

EDCS - Electronic Document Control System

EDSS - Electronic Document Storage System

EDSS - Electronic Document Submission System

EDES - Electronic Document Encryption System

EDLS - Electronic Document Labeling System

EDIS - Electronic Document Information System

EDOCS - Electronic Document Management System

EDMS - Electronic Document Management System

EDMS - Electronic Document Marking System

MOLIS - Modular Open Laboratory Information System

DIS - Document Information System

EDDS - Electronic Document Delivery System

EDAS - Electronic Document Approval System

EDAS - Electronic Document Archive System

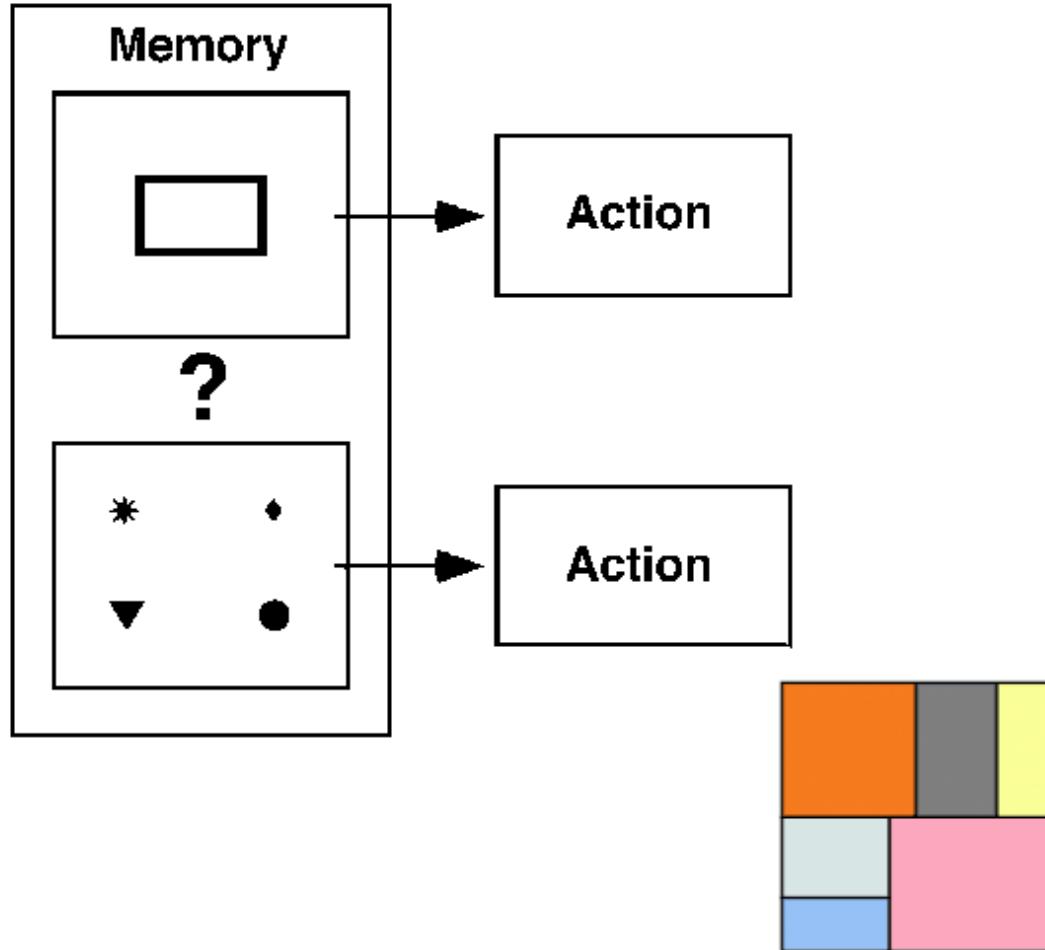
EDMS - Electronic Document Management System

**MITIS - Modular Integrated Transplant Information System
edds -**

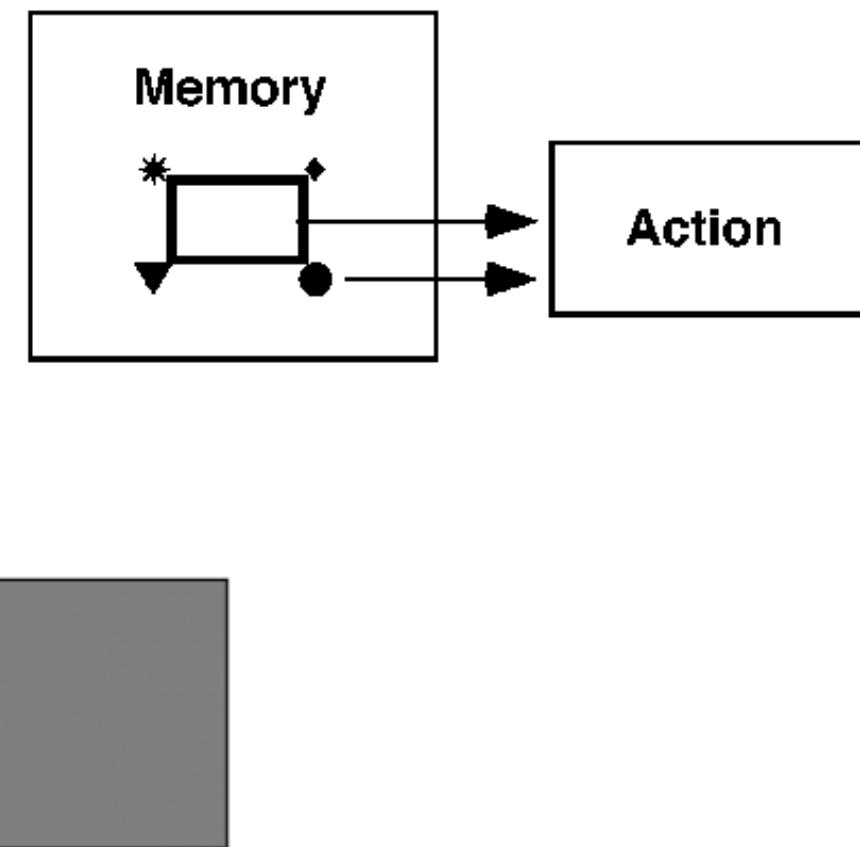
MEDIS - Modular Engineering Document Imaging System

Moduly vs Monolity

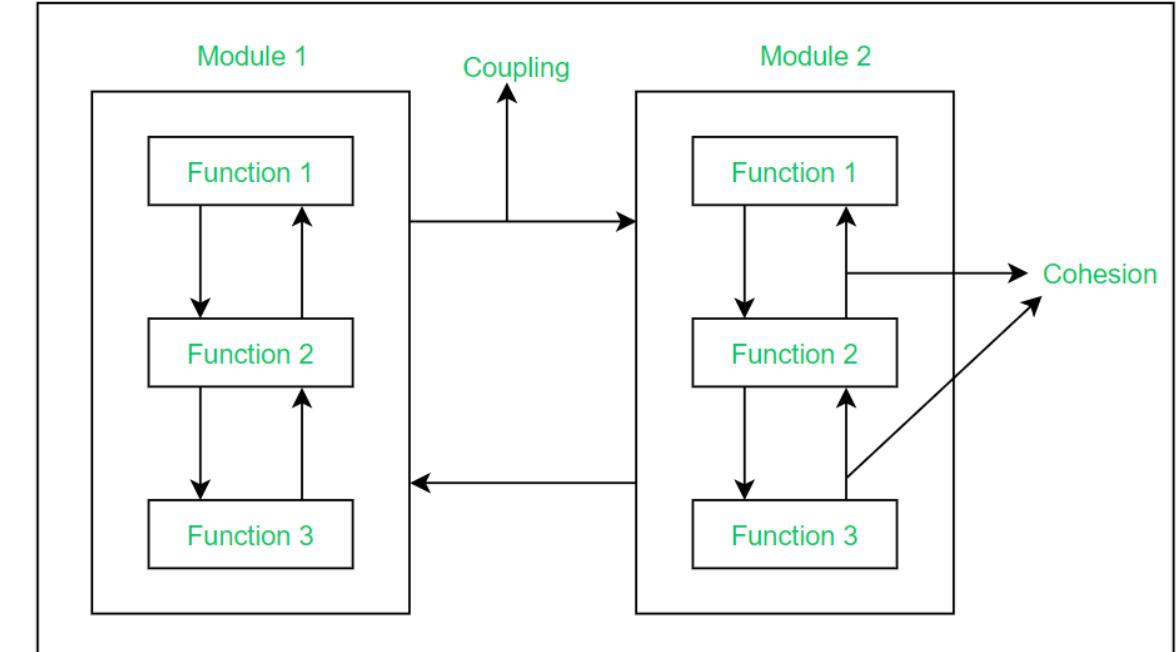
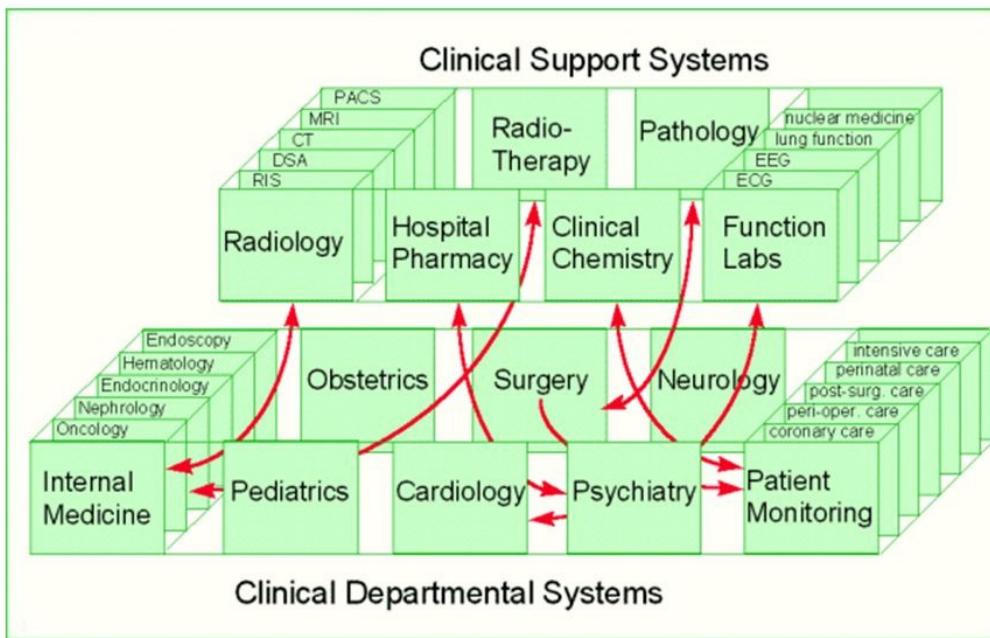
Modulárne systémy



Monolitické (unifikované) systémy



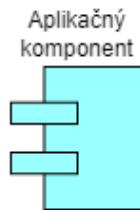
Príklady modularity



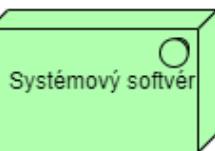
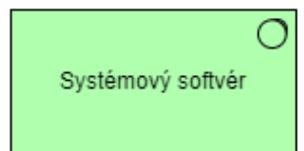
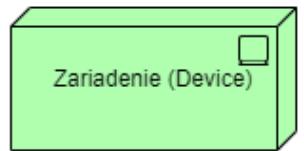
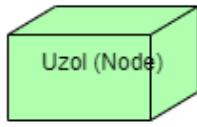
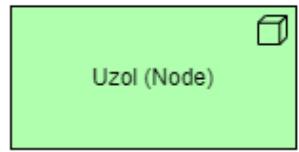
ArchiMate (Enterprise architektúra)

UML (2.0/2.5)

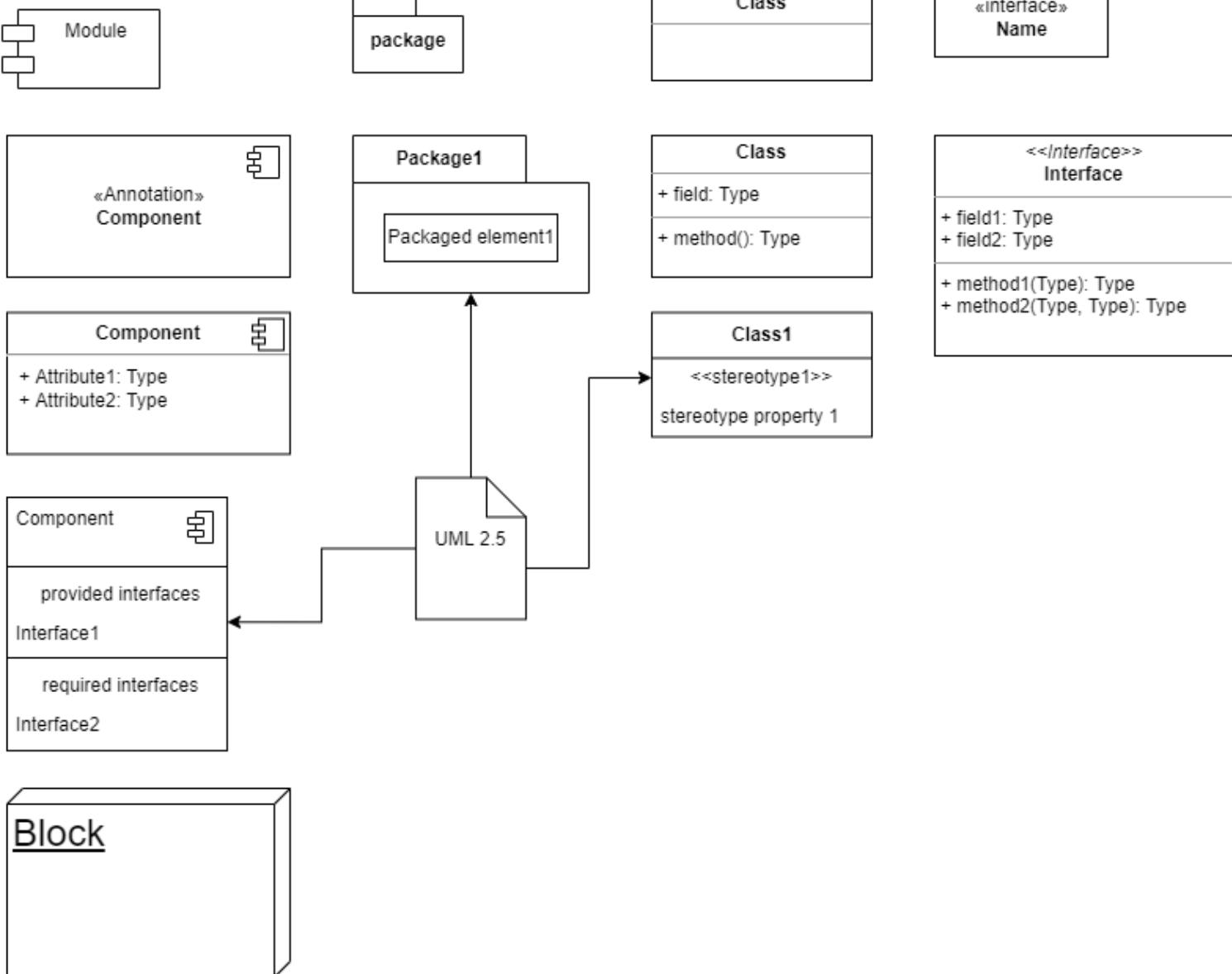
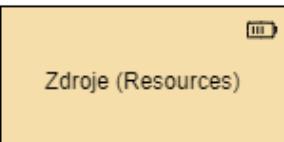
Applikačná vrstva



Technologická vrstva



Strategická vrstva



Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.

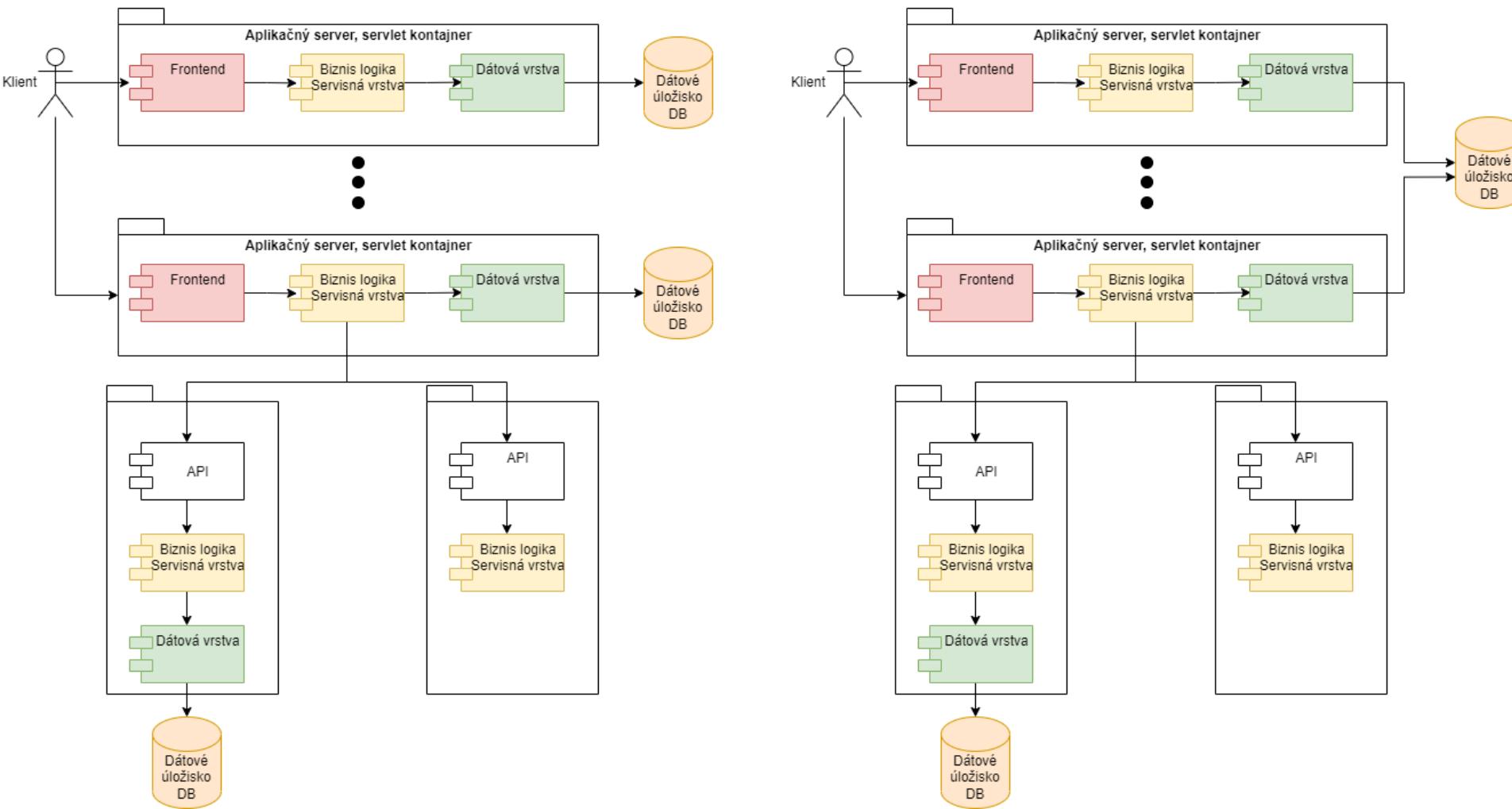
-- **Doug McIlroy** (This is the Unix philosophy)

Architektúra monolitickej aplikácie (trojvrstvová architektúra)



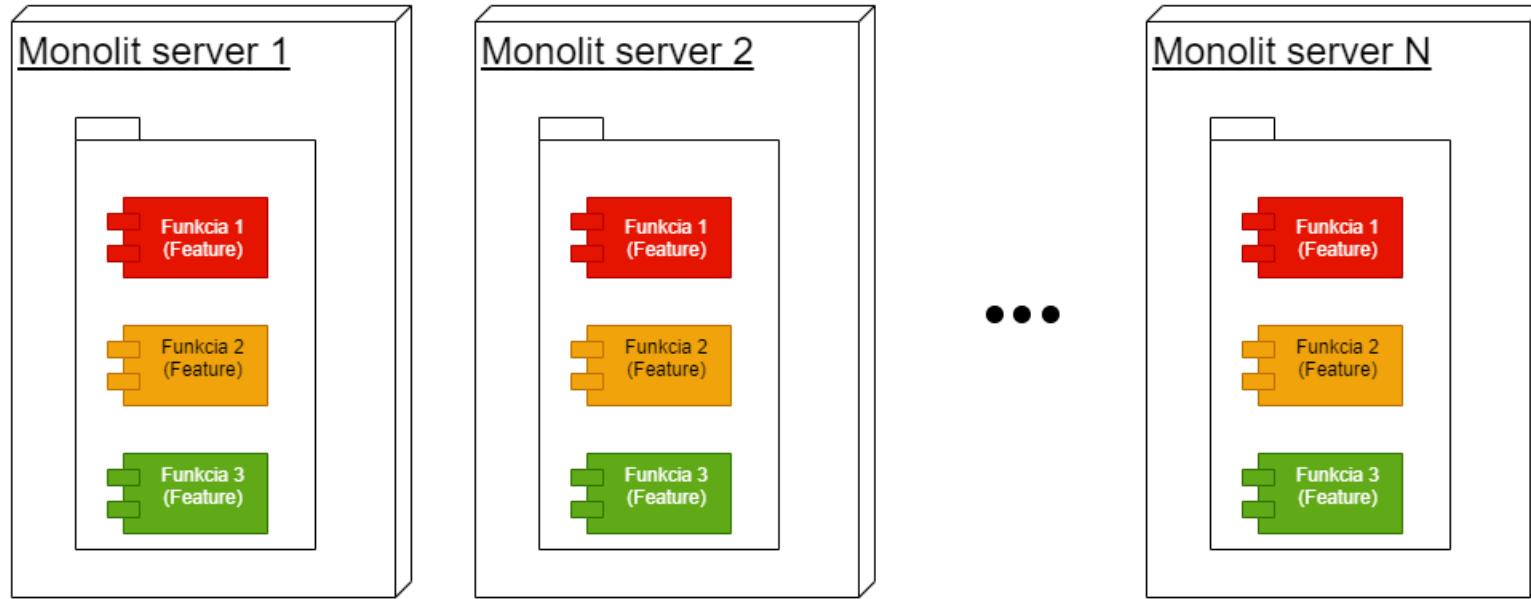
11

Architektúra aplikácie využívajúcej mikroslužby

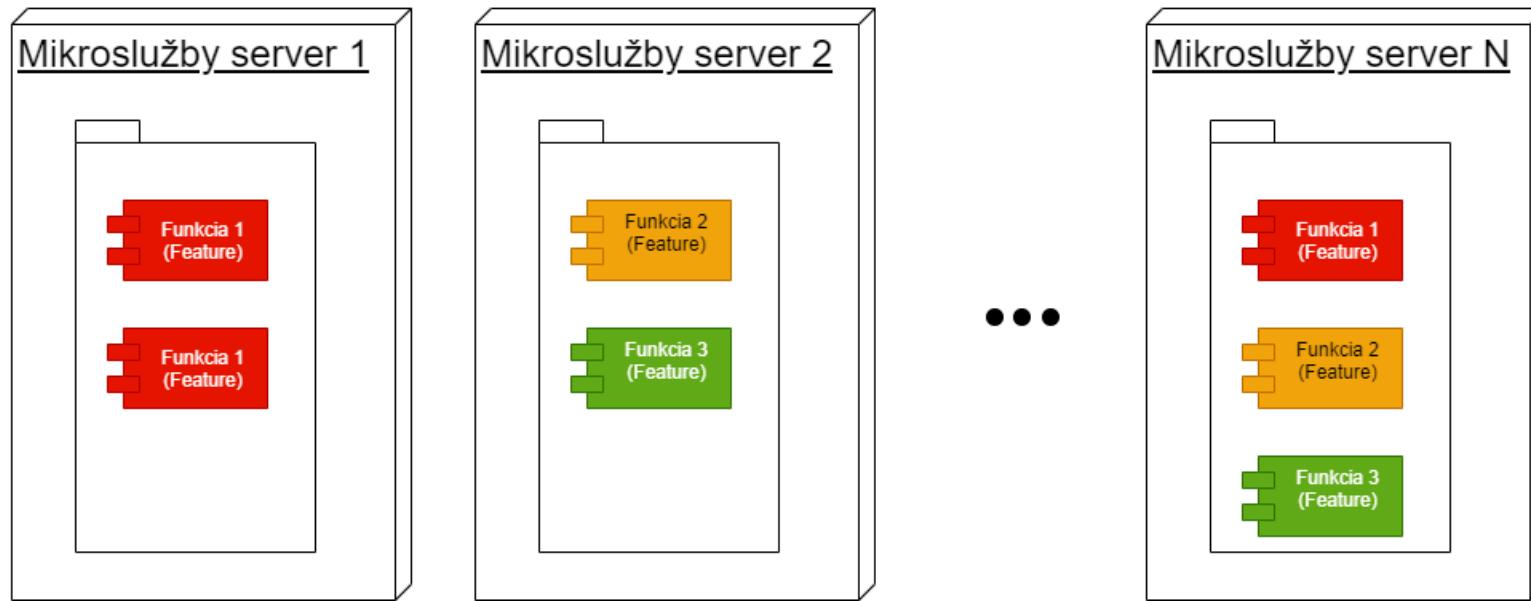


Škálovanie monolitickej aplikácie

12



Škálovanie aplikácie využívajúcej mikroslužby



Rozdiely v prístupe konceptov

SOA (Servisne orientovaná architektúra)

1. Skôr hrubšia granularita služieb („mikromonolity“)
2. Zameranie na štandardizáciu procesov, nástrojov a pod.
3. Použitie ESB (Enterprise Service Bus)
4. Podpora väčšieho množstva protokolov na prenos správ
5. Založené na jednom programovacom jazyku a sade knižníc
6. Beh vo viacerých vláknoch
7. Služby delené podľa business požiadaviek
8. Jediná databáza pre celú aplikáciu
9. Požiadavka na zmenu: úprava (mikro)monolitu

MSA/MSOA (Microservice Architecture)

1. Jemnejšia granularita služieb
2. Zameranie na spoluprácu ľudí a možnosť slobodného výberu technológií
3. Jednoduché systémy na posielanie správ
4. Zameranie na použitie jednoduchých protokolov (HTTP, STOMP, ...)
5. Voľnosť výberu jazyka i knižníc podľa potreby
6. Typicky beh v jednom vlákne s non-locking I/O, použitie zelených vlákien
7. Delenie skôr podľa kontextu
8. Každá mikroslužba používa vlastné dátové úložisko
9. Požiadavka na zmenu: vytvorenie novej mikroslužby

		Java Language											
		java	javac	javadoc	jar	javap	jdeps	Scripting					
JDK	Tools & Tool APIs	Security	Monitoring	JConsole	VisualVM	JMC	JFR						
		JPDA	JVM TI	IDL	RMI	Java DB	Deployment						
		Internationalization		Web Services		Troubleshooting							
JRE	Deployment	Java Web Start			Applet / Java Plug-in								
		JavaFX											
		Swing		Java 2D		AWT	Accessibility						
JRE	User Interface Toolkits	Drag and Drop		Input Methods		Image I/O	Print Service	Sound					
		IDL	JDBC	JNDI	RMI	RMI-IIOP		Scripting					
		Beans	Security		Serialization		Extension Mechanism						
JRE	Integration Libraries	JMX	XML JAXP		Networking		Override Mechanism						
		JNI	Date and Time		Input/Output		Internationalization						
		lang and util											
JRE	Base Libraries	Math		Collections		Ref Objects		Regular Expressions					
		Logging		Management		Instrumentation		Concurrency Utilities					
		Reflection	Versioning		Preferences API		JAR	Zip					
Java Virtual Machine		Java HotSpot Client and Server VM											

Java SE

API

Compact Profiles

System (C:) > Program Files > Java > jdk-11.0.9 >

- 📁 bin
- 📁 conf
- 📁 include
- 📁 jmods
- 📁 legal
- 📁 lib
- 📄 COPYRIGHT
- ⭕ README.html
- 📄 release

System (C:) > Program Files > Java > jdk-11.0.9 >

- 📁 bin
- 📁 conf
- 📁 include
- 📁 jmods
- 📁 legal
- 📁 lib
- 📄 COPYRIGHT
- ⭕ README.html
- 📄 release

System (C:) > Program Files > Java > jdk-14.0.2 >

Názov

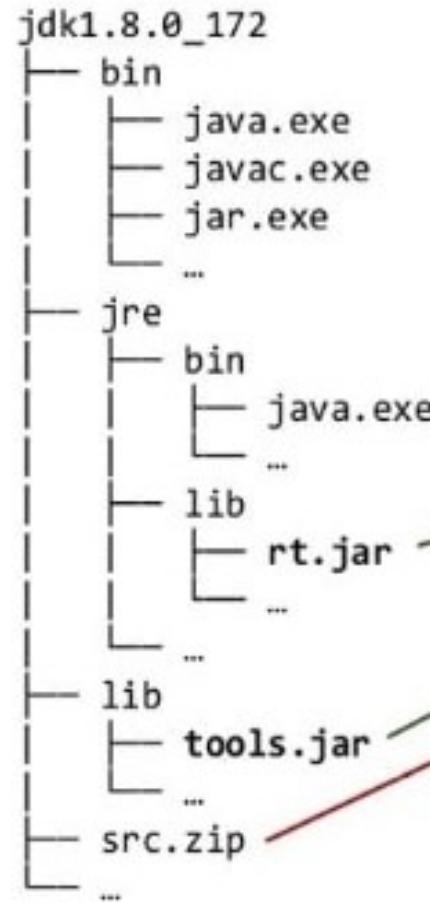
- 📁 bin
- 📁 conf
- 📁 include
- 📁 jmods
- 📁 legal
- 📁 lib
- 📄 COPYRIGHT
- 📄 release

Správne nastavenie premennej prostredia JAVA_HOME

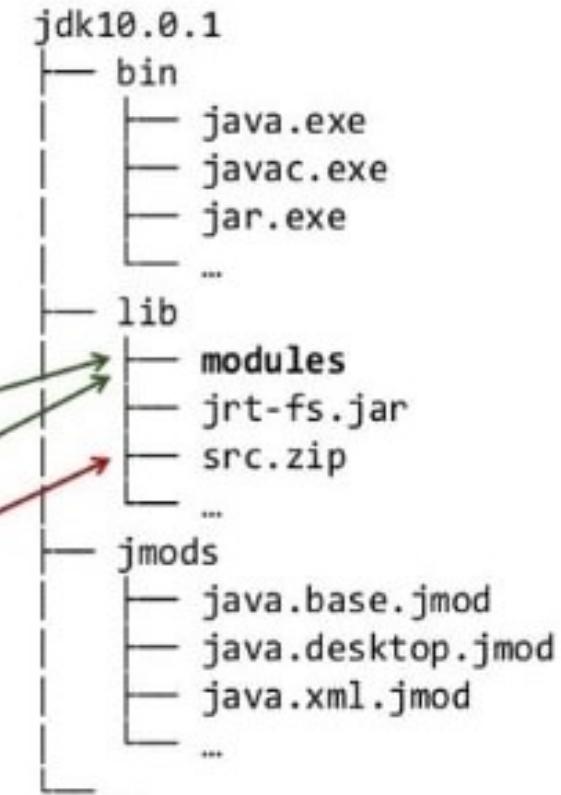
Najdite 1 dôležitý rozdiel (Nie Readme)

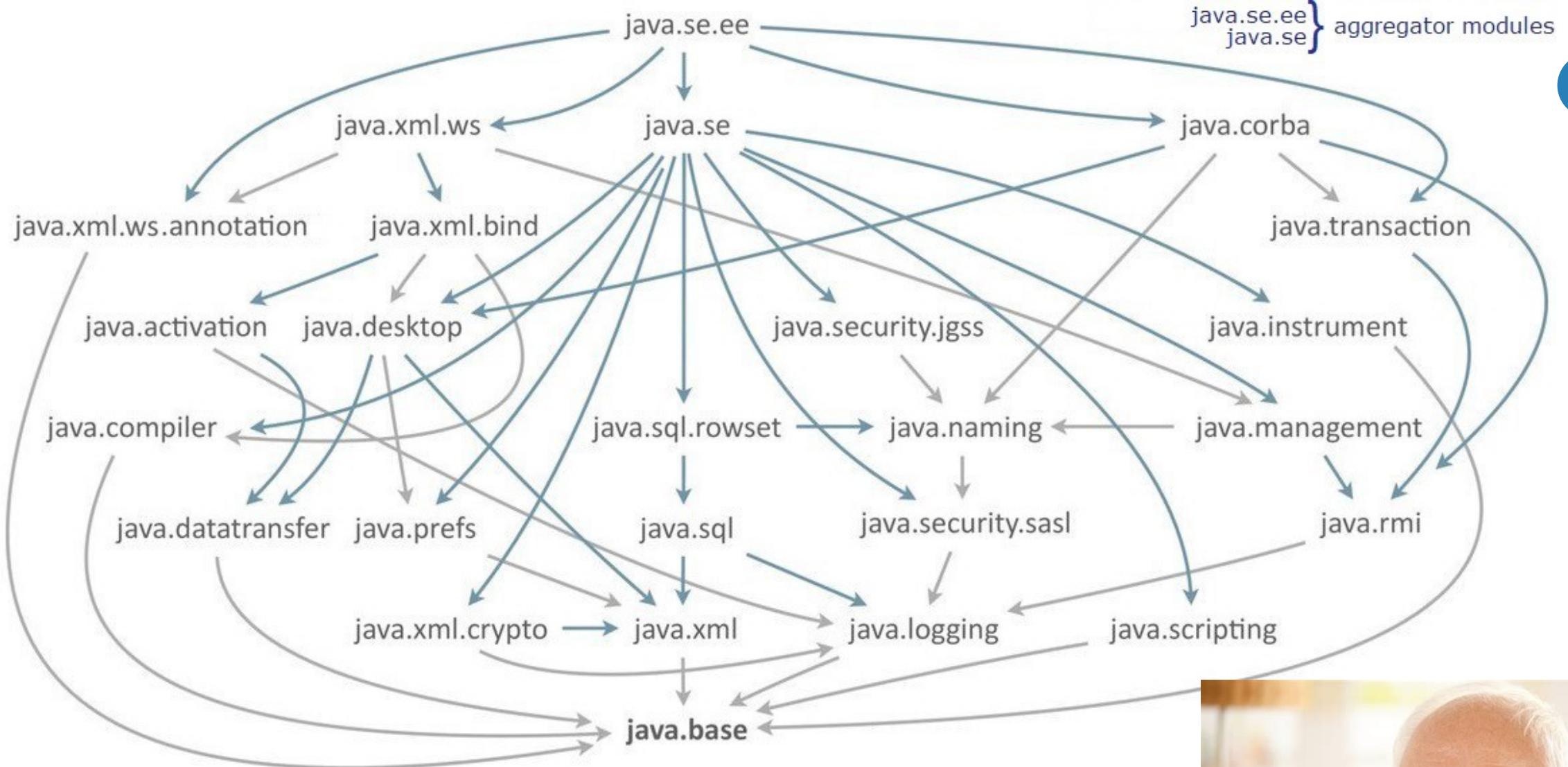
Zavedenie modulov

Java 8



Java 9+





Používateľské premenné pre používateľa Administrator

Premenná	Hodnota
OneDrive	D:\OneDrive
OneDriveConsumer	D:\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\Administ...
PGDATABASE	postgres
PyCharm Community Editio...	C:\Program Files\JetBrains\PyCharm Community Edition with Ana...
TEMP	C:\Users\Administrator\AppData\Local\Temp
TMP	C:\Users\Administrator\AppData\Local\Temp

Nové...Upravit...Odstrániť

Systémové premenné

Premenná	Hodnota
IWBPath	C:\Program Files (x86)\SAP\FrontEnd\iwb
JAVA_HOME	C:\Program Files\Java\jre1.8.0_181
MIC_LD_LIBRARY_PATH	%INTEL_DEV_REDIST%compiler\lib\mic
MSMPI_BIN	C:\Program Files\Microsoft MPI\Bin\
NUMBER_OF_PROCESSORS	8
OS	Windows_NT
Path	C:\Program Files (x86)\Intel\Intel(R) Management Engine Compo...
PATH_ORIGINAL	C:\Users\Administrator\AppData\Roaming\ActiveState\bin;C:\Pvt...

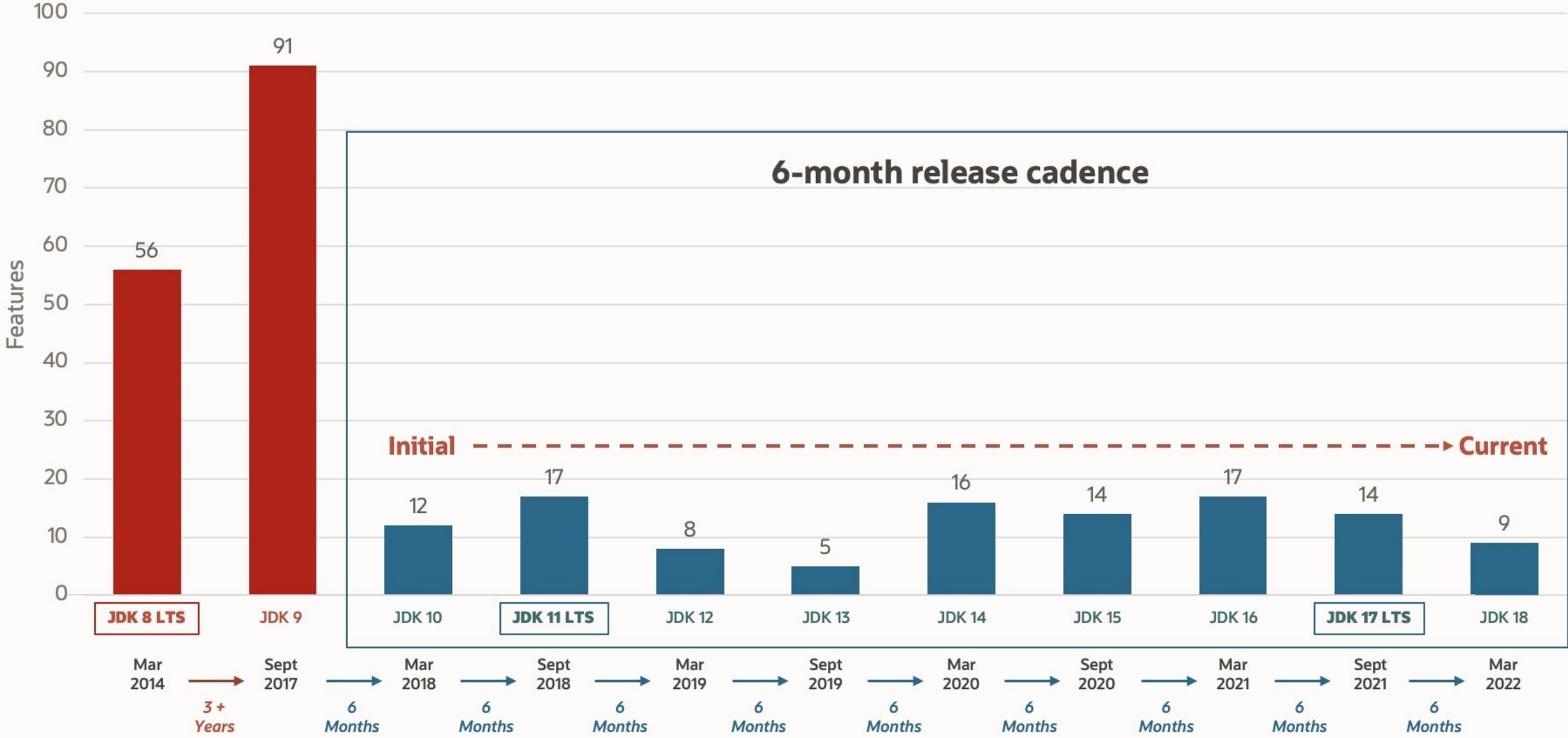
Nové...Upravit...OdstrániťOKZrušiť

Java LTS

- Java 7 až do roku 2022
- Java 8 minimálne do roku 2030
- Java 11 až 2026
- Java 17 minimálne do roku 2029
- **Najnovšia Java 18 (Nie LTS/non LTS)**
- Budúca Java 21 (Bude LTS, 09-2023)



The 6-month feature-release model



Issues fixed in JDK 11-JDK 18 per organization

■ Alibaba ■ Amazon ■ Ampere Computing ■ ARM ■ Azul ■ BellSoft ■ DataDog
■ Fujitsu ■ Google ■ Huawei ■ IBM ■ Independent ■ Intel ■ JetBrains
■ Linaro ■ Loongson ■ Microdoc ■ Microsoft ■ NTT Data ■ Oracle



System Dashboard

Introduction



Welcome to the JDK Bug System

The JDK Bug System (JBS) is a JIRA instance which provides bug tracking for Projects in the [OpenJDK Community](#).

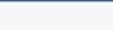
Everyone with OpenJDK Author status or above has a JBS account which may be used to create and edit bugs. Those without accounts can view bugs anonymously.

See the [OpenJDK wiki](#) for information on using the system. For help in resolving access problems send email to help@openjdk.java.net.

Projects



Open JDK Projects



Code Tools (CODETOOLS)



Lead J. Duke

Open Issues (by priority)

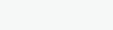
Device I/O (DIO)



Lead Olga Nazarkina

Open Issues (by priority)

Description The Device I/O Project provides a Java-level API for accessing generic device peripherals on embedded devices.



JDK (JDK)



Lead J. Duke

Open Issues (by priority)

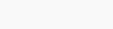
Java Mission Control (JMC)



Lead Marcus Hirt

Open Issues (by priority)

Description The production time profiling and diagnostics tools-suite for the HotSpot JDK.



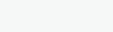
Kona (KONA)



Lead Riaz Aimandi

Open Issues (by priority)

Description Welcome to your project



Everything you need to know about how your project is running is tracked on this page. As your project evolves, the information will be updated. Use the tabs on the left to navigate within your project.

Describe your project here

Change the project description to include details about your project.



JDK

Open issues [Switch filter ▾](#)[View all issues and filters](#)

Order by Priority

- [JDK-8284855](#) Update needed to Cleaners added t...
- [JDK-8284108](#) RunThese hang with Xcomp in loom...
- [JDK-8284220](#) TypeMirror#toString omits enclosing...
- [JDK-8281297](#) TestStressG1Humongous fails with ...
- [JDK-8278390](#) [jworld] Scalarization of nullable inlin...
- [JDK-8269820](#) C2 PhaseldealLoop::do_unroll get ...
- [JDK-8283899](#) G1PageBasedVirtualSpace::uncom...
- [JDK-8282080](#) Lambda deserialization fails for Obj...
- [JDK-8087163](#) Re-examine caller-sensitivity of j...
- [JDK-8134507](#) Improve the tiered compilation com...
- [JDK-8282555](#) Missing memory edge when spilling ...
- [JDK-8283716](#) java/time/test/java/time/TestZoneOff...
- [JDK-8283590](#) Application window does not get act...
- [JDK-8276797](#)

JDK / [JDK-8284855](#)

Update needed to Cleaners added to jdk.crypto.cryptoki

1 of 20537

[Export ▾](#)

Details

Type:	Bug	Status:	OPEN
Priority:	P2	Resolution:	Unresolved
Affects Version/s:	19	Fix Version/s:	19
Component/s:	security-libs		
Labels:	tencent-interest		
Subcomponent:	javax.crypto:pkcs11		

People

Assignee:	Xueli Fan
Reporter:	Brent Christian
Votes:	0 Vote for this issue
Watchers:	2 Start watching this issue

Description

The fix for [JDK-8284368](#) replaced the finalizers in jdk.crypto.cryptoki with Cleaners.

However, there is a problem with the code changes. The Runnables registered with Cleaner refer to the object being registered ('this'). Meaning, the Cleaner mechanism will keep the objects reachable, preventing them from being cleaned and collected.

The [JDK-8284368](#) change needs to be reworked to not reference 'this'.

Issue Links

relates to

[JDK-8253568](#) Replace and mitigate Object.finalize() uses in JDK libraries OPEN

[JDK-8284368](#) Remove finalizer method in jdk.crypto.cryptoki RESOLVED

Activity

All Comments Work Log History Activity

Brent Christian added a comment - 15 hours ago

For PKCS11, some native refactoring may be needed to allow it to disconnect() without keeping the PKCS11 object itself around.

```
C:\Program Files\Java\jdk-11.0.9\bin>java -version
java version "11.0.9" 2020-10-20 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.9+7-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.9+7-LTS, mixed mode)
```

```
C:\Program Files\Java\jdk-11.0.9\bin>java --version
java 11.0.9 2020-10-20 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.9+7-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.9+7-LTS, mixed mode)
```

```
C:\Program Files\Java\jdk-11.0.9\bin>cd C:\Program Files\Java\jdk1.8.0_221\bin
```

```
C:\Program Files\Java\jdk1.8.0_221\bin>java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
```

```
C:\Program Files\Java\jdk1.8.0_221\bin>java --version
Unrecognized option: --version
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
```

```
C:\Program Files\Java\jdk1.8.0_221\bin>cd C:\Program Files\Java\jdk-14.0.2\bin
```

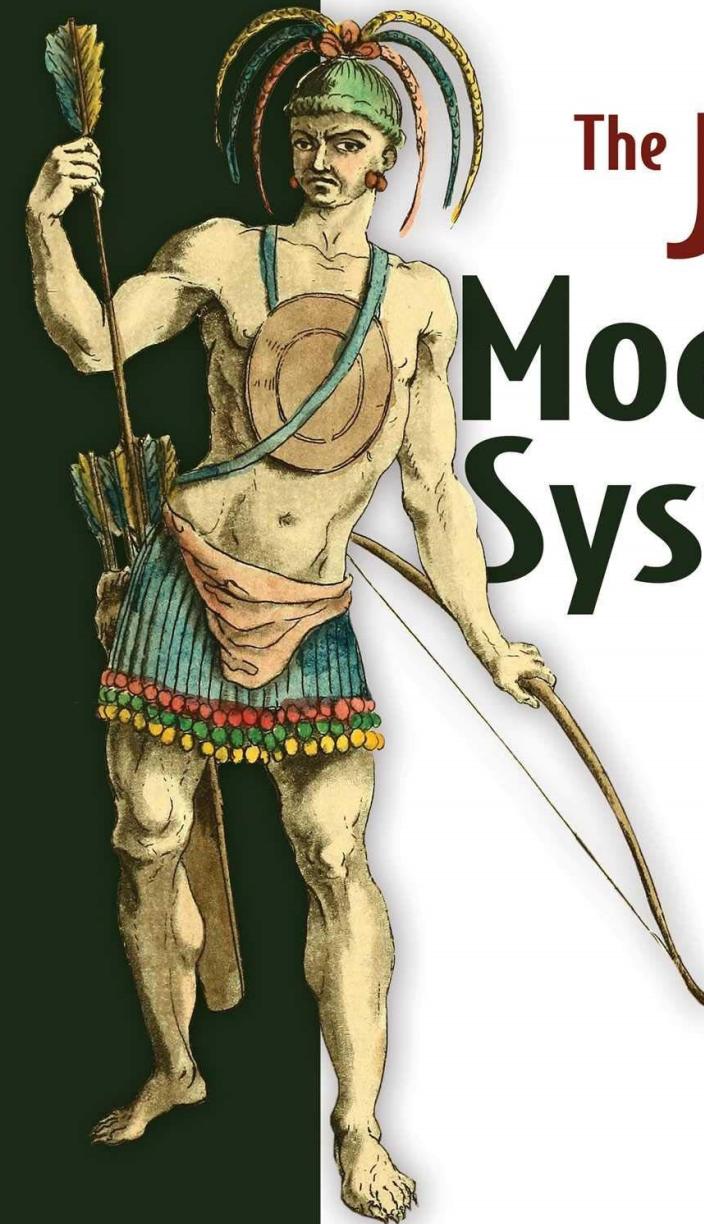
```
C:\Program Files\Java\jdk-14.0.2\bin>java -version
java version "14.0.2" 2020-07-14
Java(TM) SE Runtime Environment (build 14.0.2+12-46)
Java HotSpot(TM) 64-Bit Server VM (build 14.0.2+12-46, mixed mode, sharing)
```

```
C:\Program Files\Java\jdk-14.0.2\bin>java --version
java 14.0.2 2020-07-14
Java(TM) SE Runtime Environment (build 14.0.2+12-46)
Java HotSpot(TM) 64-Bit Server VM (build 14.0.2+12-46, mixed mode, sharing)
```

```
C:\Program Files\Java\jdk-14.0.2\bin>
```

Updated for Java 11

The Java Module System



Nicolai Parlog
Foreword by Kevlin Henney

```
C:\Program Files\Java\jdk-11.0.9\bin>java
Usage: java [options] <mainclass> [args...]
      (to execute a class)
or  java [options] -jar <jarfile> [args...]
      (to execute a jar file)
or  java [options] -m <module>[/<mainclass>] [args...]
    java [options] --module <module>[/<mainclass>] [args...]
      (to execute the main class in a module)
or  java [options] <sourcefile> [args]
      (to execute a single source-file program)
```

Arguments following the main class, source file, -jar <jarfile>, -m or --module <module>/<mainclass> are passed as the arguments to main class.

where options include:

```
-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
--class-path <class search path of directories and zip/jar files>
      A ; separated list of directories, JAR archives,
      and ZIP archives to search for class files.
-p <module path>
--module-path <module path>...
      A ; separated list of directories, each directory
      is a directory of modules.
--upgrade-module-path <module path>...
      A ; separated list of directories, each directory
      is a directory of modules that replace upgradeable
      modules in the runtime image
--add-modules <module name>[,<module name>...]
      root modules to resolve in addition to the initial module.
      <module name> can also be ALL-DEFAULT, ALL-SYSTEM,
      ALL-MODULE-PATH.
--list-modules
      list observable modules and exit
-d <module name>
--describe-module <module name>
      describe a module and exit
--dry-run
      create VM and load main class but do not execute main method.
      The --dry-run option may be useful for validating the
      command-line options such as the module system configuration.
--validate-modules
      validate all modules and exit
```

Programátori Vývojáři Architekti DevOps

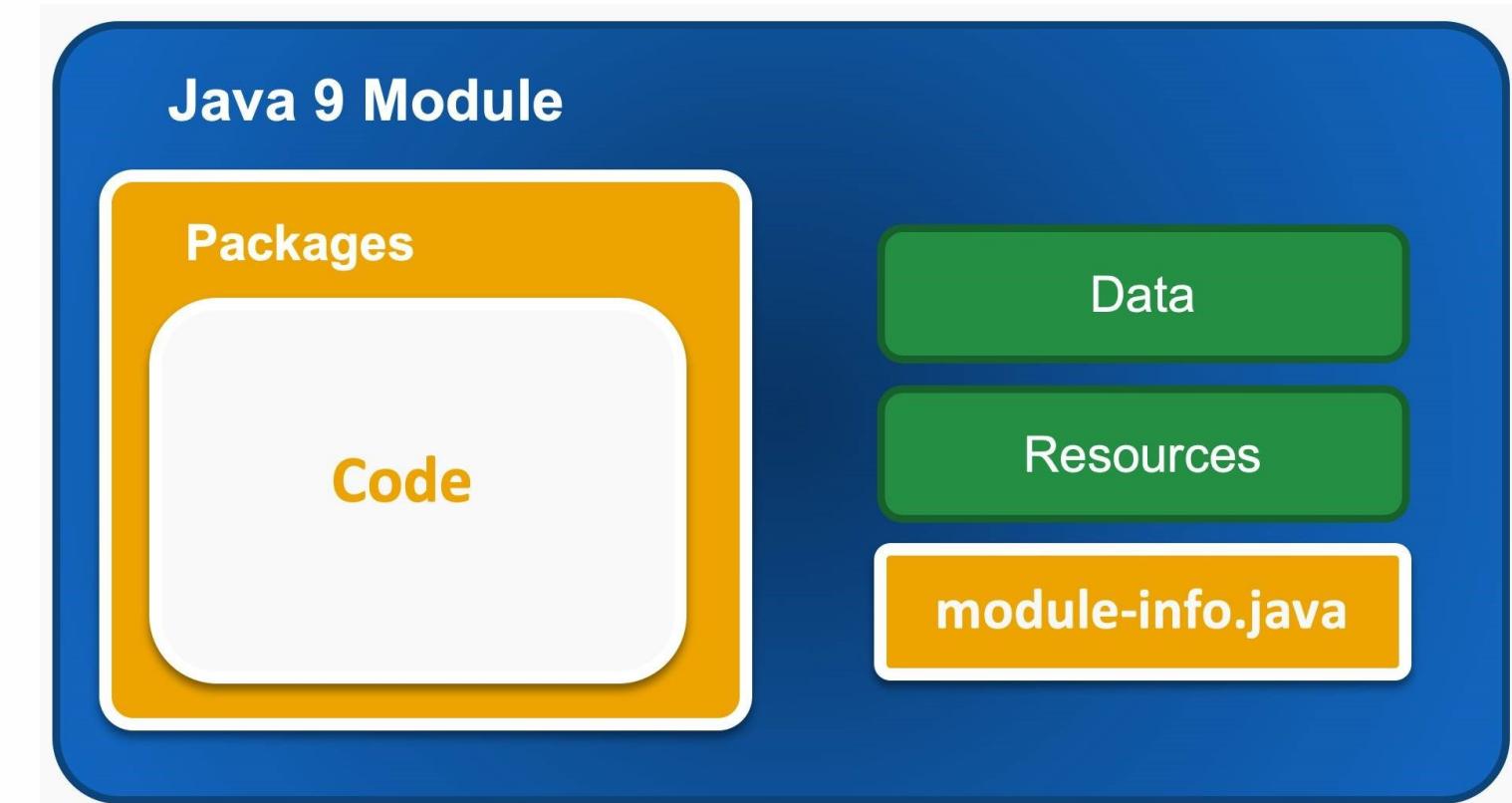
```
-p <module path>
--module-path <module path>...
    A ; separated list of directories, each directory
    is a directory of modules.
--upgrade-module-path <module path>...
    A ; separated list of directories, each directory
    is a directory of modules that replace upgradeable
    modules in the runtime image
--add-modules <module name>[,<module name>...]
    root modules to resolve in addition to the initial module.
    <module name> can also be ALL-DEFAULT, ALL-SYSTEM,
    ALL-MODULE-PATH.
--list-modules
    list observable modules and exit
-d <module name>
--describe-module <module name>
    describe a module and exit
--dry-run
    create VM and load main class but do not execute main method.
    The --dry-run option may be useful for validating the
    command-line options such as the module system configuration.
--validate-modules
    validate all modules and exit
    The --validate-modules option may be useful for finding
    conflicts and other errors with modules on the module path.
-D<name>=<value>
    set a system property
-verbose:[class|module|gc|jni]
    enable verbose output
-version
    print product version to the error stream and exit
--version
    print product version to the output stream and exit
-showversion
    print product version to the error stream and continue
--show-version
    print product version to the output stream and continue
--show-module-resolution
    show module resolution output during startup
-? -h -help
    print this help message to the error stream
--help
    print this help message to the output stream
-X
    print help on extra options to the error stream
--help-extra
    print help on extra options to the output stream
-ea[:<packagename>...|:<classname>]
-enableassertions[:<packagename>...|:<classname>]
    enable assertions with specified granularity
-da[:<packagename>...|:<classname>]
```

**Programátori
Vývojáři
Architekti
DevOps**

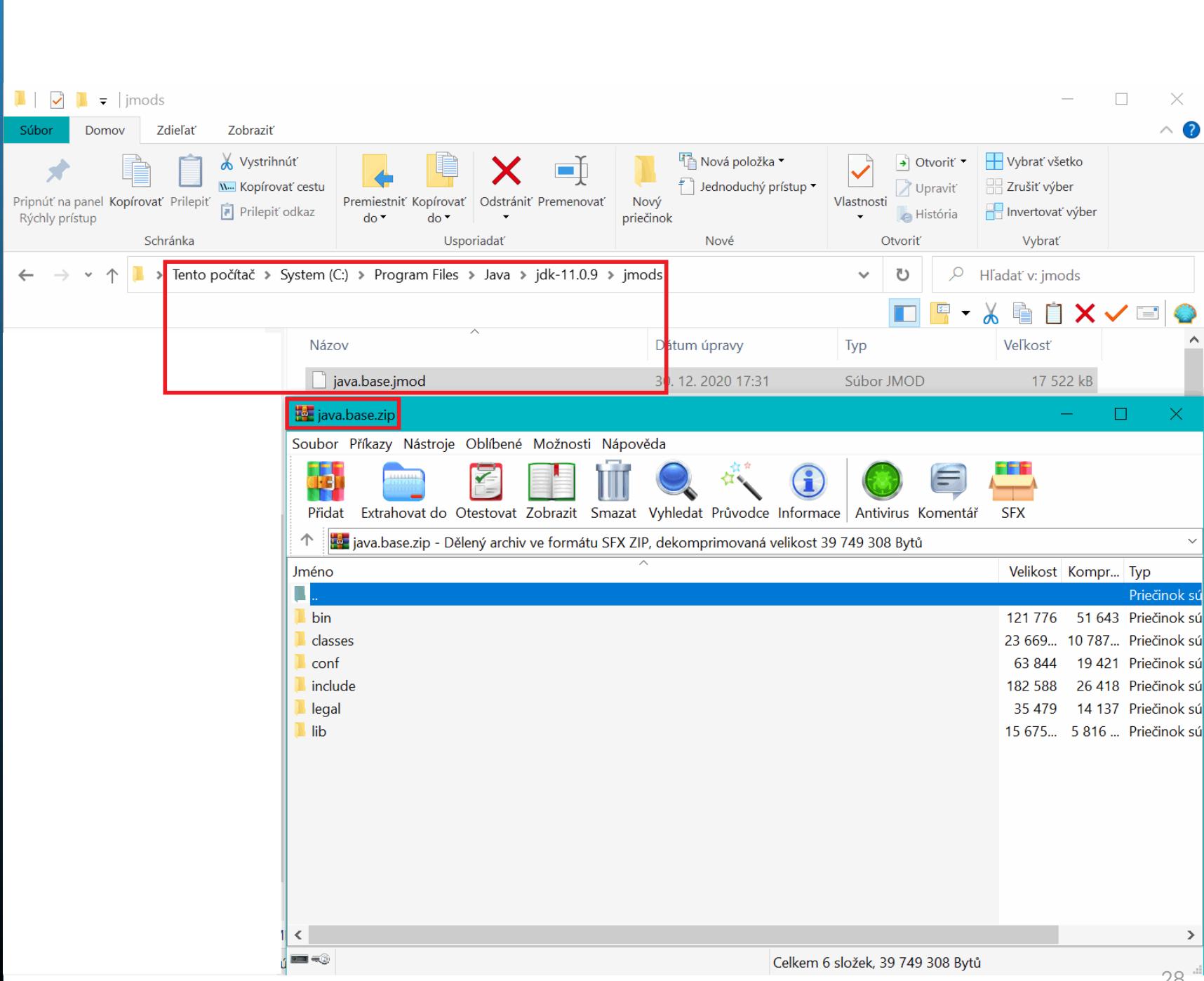
**Testeri
Test analytici**

```
C:\Program Files\Java\jdk-11.0.9\bin>java --list-modules
```

```
java.base@11.0.9
java.compiler@11.0.9
java.datatransfer@11.0.9
java.desktop@11.0.9
java.instrument@11.0.9
java.logging@11.0.9
java.management@11.0.9
java.management.rmi@11.0.9
java.naming@11.0.9
java.net.http@11.0.9
java.prefs@11.0.9
java.rmi@11.0.9
java.scripting@11.0.9
java.se@11.0.9
java.security.jgss@11.0.9
java.security.sasl@11.0.9
java.smartcardio@11.0.9
java.sql@11.0.9
java.sql.rowset@11.0.9
java.transaction.xa@11.0.9
java.xml@11.0.9
java.xml.crypto@11.0.9
jdk.accessibility@11.0.9
jdk.aot@11.0.9
jdk.attach@11.0.9
jdk.charsets@11.0.9
jdk.compiler@11.0.9
jdk.crypto.cryptoki@11.0.9
jdk.crypto.ec@11.0.9
jdk.crypto.mscapi@11.0.9
jdk.dynalink@11.0.9
jdk.editpad@11.0.9
jdk.hotspot.agent@11.0.9
jdk.httpserver@11.0.9
jdk.internal.ed@11.0.9
jdk.internal.jvmsstat@11.0.9
jdk.internal.le@11.0.9
jdk.internal.opt@11.0.9
jdk.internal.vm.ci@11.0.9
jdk.internal.vm.compiler@11.0.9
jdk.internal.vm.compiler.management@11.0.9
jdk.jartool@11.0.9
jdk.javadoc@11.0.9
```



Jmods (Zip súbory) podobne ako Jar



```
C:\Program Files\Java\jdk-11.0.9\bin>java -d java.base
java.base@11.0.9
exports java.io
exports java.lang
exports java.lang.annotation
exports java.lang.invoke
exports java.lang.module
exports java.lang.ref
exports java.lang.reflect
exports java.math
exports java.net
exports java.net.spi
exports java.nio
exports java.nio.channels
exports java.nio.channels.spi
exports java.nio.charset
exports java.nio.charset.spi
exports java.nio.file
exports java.nio.file.attribute
exports java.nio.file.spi
exports java.security
exports java.security.acl
exports java.security.cert
exports java.security.interfaces
exports java.security.spec
exports java.text
exports java.text.spi
exports java.time
exports java.time.chrono
exports java.time.format
exports java.time.temporal
exports java.time.zone
exports java.util
exports java.util.concurrent
exports java.util.concurrent.atomic
exports java.util.concurrent.locks
exports java.util.function
exports java.util.jar
exports java.util.regex
exports java.util.spi
exports java.util.stream
exports java.util.zip
exports javax.crypto
exports javax.crypto.interfaces
```

java --describe-module <module name>
java --describe-module java.base

Prehľad modulov

Typy modulov

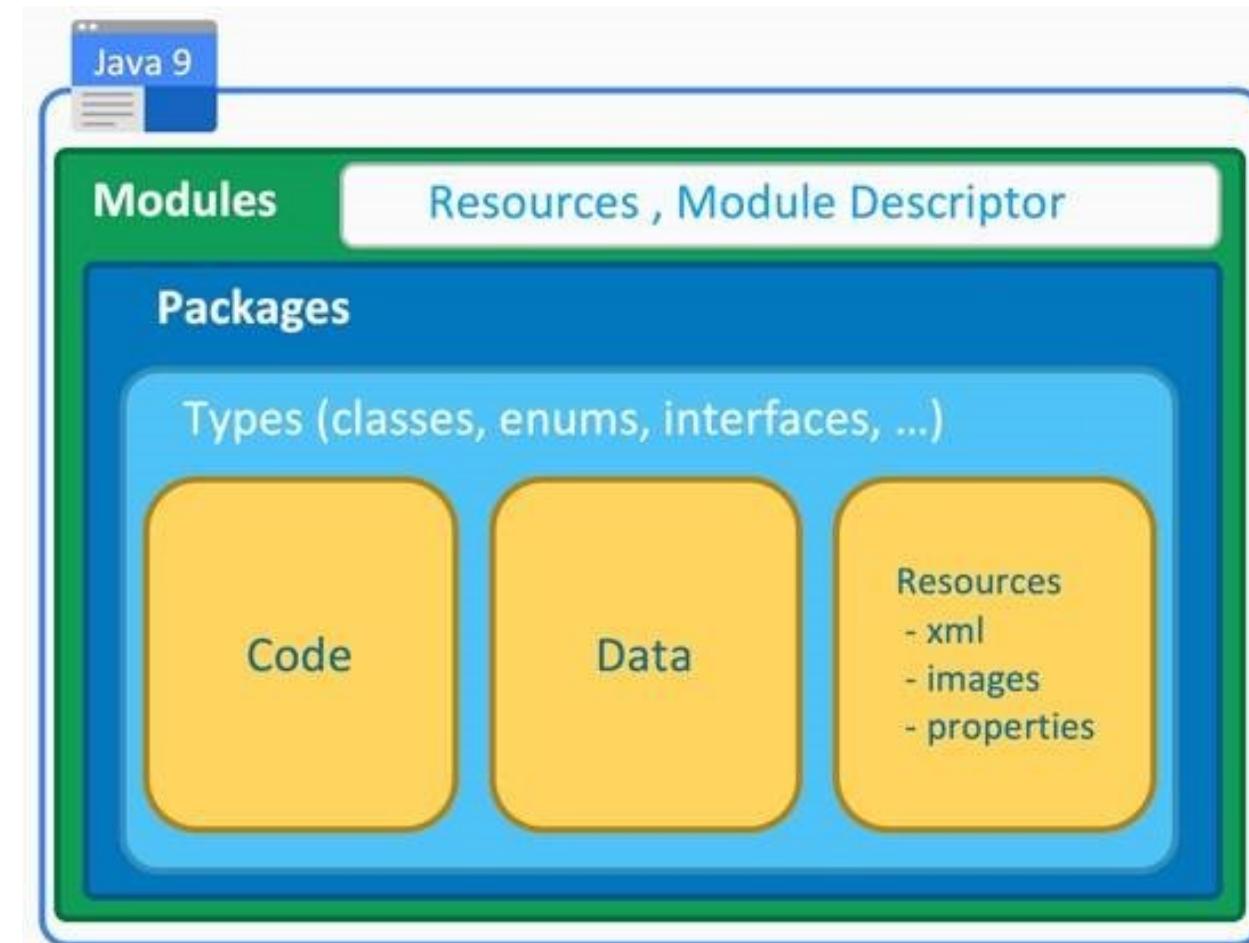
1. **Systemové moduly - Java SE a JDK moduly.** Tu máme umiestnené všetky už default poskytované moduly.
2. **Aplikačné moduly** - sú **moduly, ktoré chceme vytvárať**, keď sa rozhodneme, že chceme používať moduly.
3. **Automatické moduly** - vytvoria sa, keď pridávame **JAR súbory na path modulu**. Názov modulu sa preberie z názvu jar súboru. Tieto automatické moduly **majú plný prístup na čítanie do všetkých ostatných modulov načítaných na path**.
4. **Nepomenovaný modul** - ak sa na path (nie na module path) **načítajú nejaké triedy alebo jar súbory**, tak tieto triedy a jar súbory sú automaticky pridané do tohto nepomenovaného modulu. Slúži na **spätnú kompatibilitu s predchádzajúcim starším javovským kódom**.

Distribúcia

- **Modul má byť zbalený ako jar súbor – 1 jar súbor má obsahovať max 1 modul.**
- Keď robíme build projektu, **musíme si dať pozor, aby sme zabalili každý modul** v našom projekte ako **samostatný jar**.
- **Každý modul**, ktorý vytvoríme, bude **používať implicitne java.base modul**. Použitie iných modulov bude dostupné po konfigurácii.
- Modul **vznikne**, ak **zadefinujeme v roote zdrojového kódu modulu súbor module-info.java**.
- **Moduly sa navzájom nevidia – nemajú prístup jeden do druhého a do tretieho** a tak ďalej.

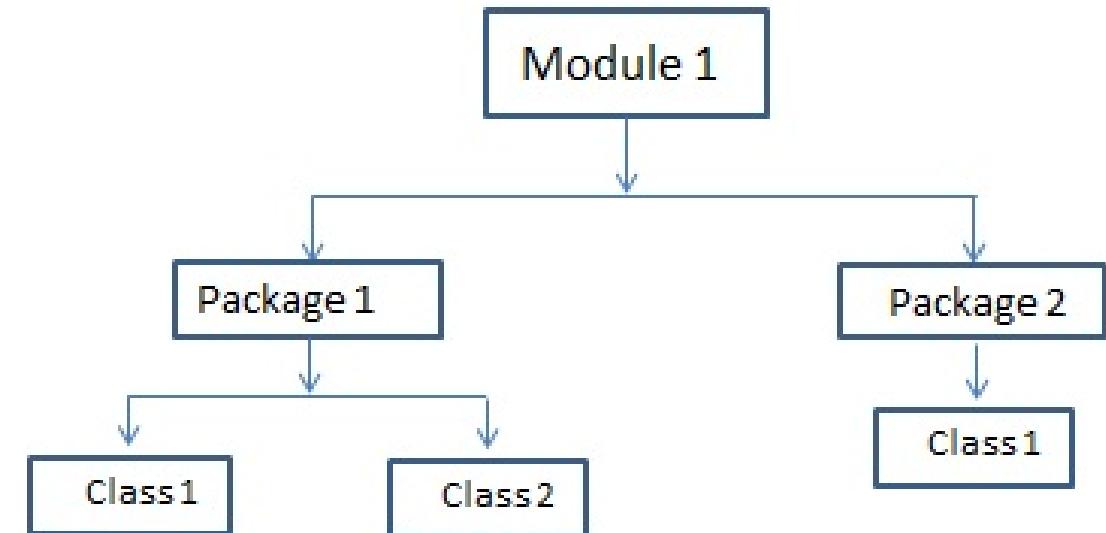
Výhoda modulov

1. Silné zapuzdrenie je hlavnou výhodou modulového systému. Keďže „public“ už nie je prístupný každému. S modulovým systémom môžeme povoliť, aby bola obmedzená množina balíkov prístupná pre vonkajšie aplikácie.
2. Vďaka tomu je vaša aplikacia "odliahčená", takže ju možno spustiť na väčšom počte zariadení.
3. Keďže je odhľahčená, zlepšuje výkon aplikácie.
4. Architektúra, ktorá vám umožňuje rozdeliť vašu aplikáciu do externých balíkov a skrytých balíkov, a teda ľahko dodržiavať princíp oddelenia záujmu (separation of concern principle).
5. Niektoré interné triedy v balíkoch ako sun.security.* , com.sun.crypto.* už nie sú prístupné, pretože tieto balíky sú teraz skryté, čím sa zvyšuje bezpečnosť.



Príklad modulu

```
1. //src/java.sql/module-info.java  
2. module java.sql {  
    1. requires transitive java.logging;  
    2. requires transitive java.xml;  
    3. exports java.sql;  
    4. exports javax.sql;  
    5. exports javax.transaction.xa;  
    6. uses java.sql.Driver;  
3. }
```



Ako vytvoriť a používať moduly

```
1. \---module-demo  
2.   \---src  
3.     +---sk.stu.kalkulacka  
4.       |   |   module-info.java  
5.       |   |  
6.       |   \---sk  
7.         \---stu  
8.           \---kalkulacka  
9.               Main.java  
10.  
11. \---sk.stu.matematicka_analyza  
12.   |   module-info.java  
13.   |  
14.   \---sk  
15.     \---stu  
16.       \--- matematicka_analyza  
17.           Mocnina.java
```

```
module sk.stu.matematicka_analyza {  
    exports sk.stu.matematicka_analyza;  
}  
  
1. package sk.stu.matematicka_analyza;  
2. public class Mocnina {  
3.     public static double power(double a, double b) {  
4.         double vysledok = Math.pow(a, b);  
5.         System.out.println("Mocnina: " + vysledok);  
6.         return vysledok;  
7.     }  
8. }
```



```
module sk.stu.kalkulacka {  
    requires sk.stu.matematicka_analyza;  
}
```

Ako vytvoriť a používať moduly

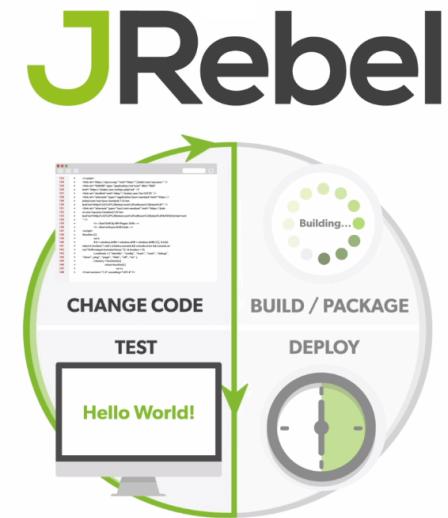
```
1. package sk.stu.kalkulacka;  
  
2. import  
sk.stu.matematicka_analyza.Mocnina;  
  
3. public class Main {  
4. public static void main(String[] args)  
{  
5.     Mocnina.umocni(Integer.parseInt(args  
[0]), Integer.parseInt(args[1]));  
6. }  
7. }
```

```
>javac -d mods/sk.stu.matematicka_analyza  
src/ sk.stu.matematicka_analyza/module-  
info.java  
src/sk.stu.matematicka_analyza/sk/stu/  
matematicka_analyza/Mocnina.java
```

```
>javac --module-path mods -d mods/  
sk.stu.kalkulacka  
src/sk.stu.kalkulacka/module-info.java src/  
sk.stu.kalkulacka  
/sk/stu/kalkulacka/Main.java
```

```
>java --module-path mods -m  
sk.stu.kalkulacka / sk.stu.kalkulacka.Main  
2 5
```

Mocnina: 32



module-info.java file contents

module module.name - declares module.name

requires module.name - this module depends on module.module.name

requires transitive module.name - this means that any module that reads your module implicitly also reads the transitive module or module specifically referenced.

exports pkg.name - this module exports public members in package pkg.name

exports pkg.name to module.name - this module allows the target module to access public members in package pkg.name

uses class.name - this module declares itself as a consumer for service class.name

provides class.name with class.name.impl - provides an implementation of a service for others to consume

opens pkg.name - allows reflective access to the private members of package pkg.name

opens pkg.name to module.name - opens private members of package pkg.name to the given module

Manifest attributes

Automatic-Module-Name: module.name - declares stable module name for non-modularized jar

Add-Exports: <module>/<package> - exports the package to all unnamed modules

Add-Opens: <module>/<package> - opens the package to all unnamed modules

Java command line options

--module-path or (-p) is the module path; its value is one or more directories that contain modules.

--add-reads src.module=target.module - a command-line form of a requires clause in a module declaration.

--add-exports src.module/pkg.name=target.module - a command line form of an exports clause.

--add-opens src.module/pkg.name=target.module - a command line form of the open clause in a module description.

--add-modules - adds the indicated modules to the default set of root modules.

--list-modules - displays the names and version strings of the observable modules.

--patch-module - adds or overrides classes in a module. Replaces -Xbootclasspath/p.

--illegal-access=permit|warn|deny - relaxes strong encapsulation of the module system; Java 9 default is permit.

Mechanism	Compile Access	Reflection Access
Export	all code → public	all code → public
Qualified Export	specified modules → public	specified modules → public
Open Package	none ↘	all code → private
Qualified Open Package	none ↘	specified modules → private
Open Module	none ↘	all code → private
Default	none ↘	none ↘

Module types

Java SE and JDK modules - modules provided by JDK: java.base, java.xml, etc.

Named application module - your application modules; contains module-info.class; explicitly exports packages; can't read the unnamed module.

Automatic module - non-modular jar on the module-path; exports all packages; name derived from the **Automatic-Module-Name** MANIFEST.MF entry or the filename; can read all modules.

Unnamed module - all jars/classes on the classpath; can read all modules.



Search or jump to...

Pull requests Issues Marketplace Explore

Bell +

boyarsky / sybex-1Z0-815-chapter-11

Public

Watch 2

Fork 38

Star 29

Code

Issues 1

Pull requests

Actions

Projects

Wiki

Security

Insights

master ▾

1 branch

0 tags

Go to file

Add file ▾



Code ▾

boyarsky document that might have to create "mods" directory

67f95c7 on 19 Jul 2020 17 commits

care	remaining examples	3 years ago
feeding	add single module section examples	3 years ago
staff	add code for Updating our Example for Mul-tiple Modules	3 years ago
talks	fix module name per errata report	2 years ago
.gitignore	start repo	3 years ago
README.md	document that might have to create "mods" directory	2 years ago

README.md

sybex-1Z0-815-chapter-11

Code and commands from OCP Oracle Certified Professional Java SE 11 Programmer I Study Guide:: Exam 1Z0-815 by Jeanne Boyarsky and Scott Selikoff

The code in chapter 11 (modules) is a lot of file creation. Additionally, the commands are hard to type in. Therefore we are providing a repository with the code from the book to facilitate running the example.

Also see:

- Our [main book page](#)

About

Code and commands from OCP Oracle Certified Professional Java SE 11 Programmer I Study Guide:: Exam 1Z0-815 by Jeanne Boyarsky and Scott Selikoff

Readme

29 stars

2 watching

38 forks

Releases

No releases published

Packages

No packages published

Contributors 2

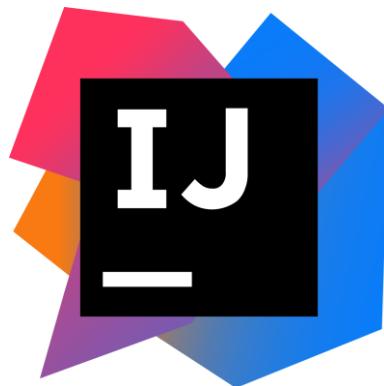
boyarsky Jeanne

atanasm Atanas Gegov

Aké IDE mám použiť?



NetBeans

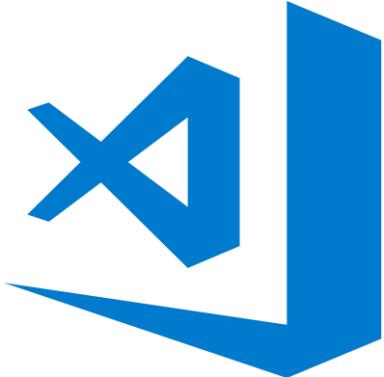
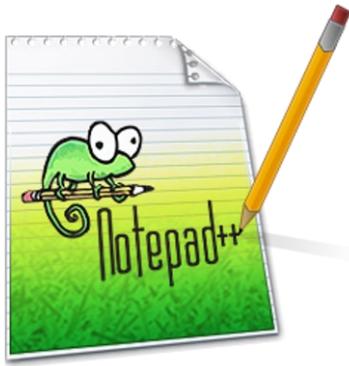


 **Visual Studio**



Integrated development environment

Aký editor mám použiť?



```
:::  
iLE88Dj. :jD88888Dj:  
.LGitE888D.f8GjjjL8888E;  
iE :8888Et. .G888.  
;i E888, ,8888,  
D888, :8888:  
D888, :8888:  
D888, :8888:  
D888, :8888:  
888W, :8888:  
W88W, :8888:  
W88W: :8888:  
DGGD: :8888:  
:8888:  
:W888:  
:8888:  
E888i  
tW88D
```



Updaty a aktualizácie



**JAVA DEVELOPERS NEVER RIP,
THEY JUST GET GARBAGE COLLECTED.**

- I LIKE NITTY-WITTY.COM

Čo sa oplatí prečítať?

Slovensko a Česko

- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

Zahraničie

- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

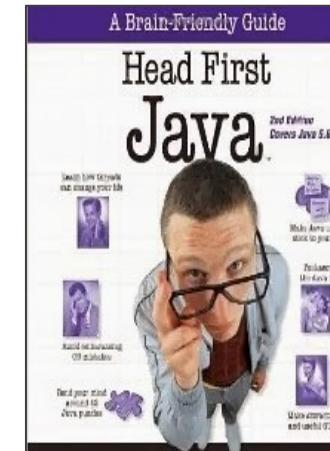
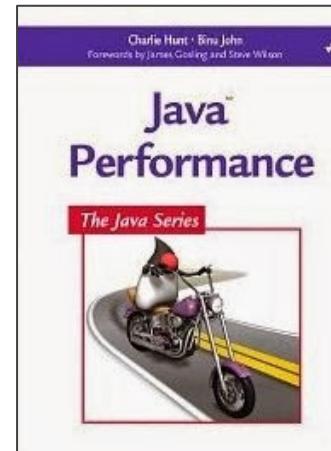
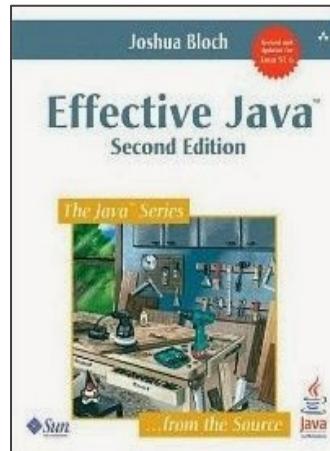
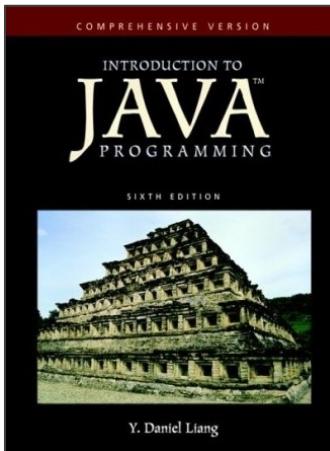
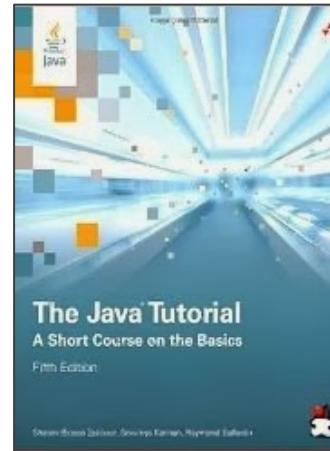
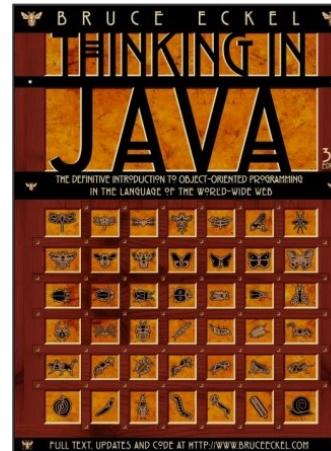
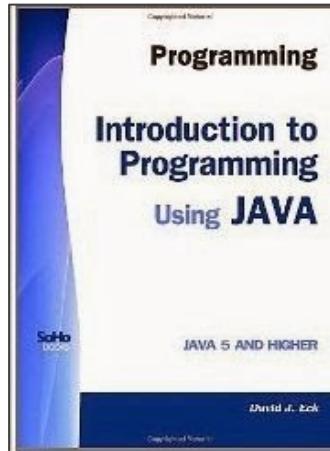
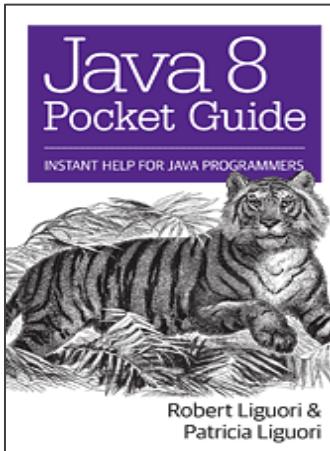
YouTube tutoriály

Čo sa oplatí/neoplatí prečítať SK/CZ



Mistrovství a Kuchárka

Čo sa oplatí/neoplatí prečítať EN



Head First

Vývojári

Verejná skupina

Informácie

Diskusia

Oznámenia

Členovia

Podujatia

Videá

Fotky

Súbory

Hľadať v tejto skupine



Ste člen ▾

✓ Upozornenia

↗ Zdieľať

... Viac



Napísat' príspěvok...

Pridať fotku/vi...

Živé video

Viac



Napište niečo...

Fotka/video

Divácka páry

Označiť priat...

...

Skratky

Podnikanie na Slove...

2

UK Manazment Externe...

Testovacia firma

VITA - Virtual It Academy

Startupisti

2

Rubyslava

2

Vývojári

▼ Zobrazit viac

NOVÁ AKTIVITA ▾



Roland Mondek

10 h

POZVAŤ ČLENOV

+ Zadajte meno alebo e-mailovú adresu...



ČLENOVIA

5 505 členov



POPIS

Skupina softvérových vývojárov. Táto skupina by mala byť miestom... [Zobraziť viac](#)

TYP SKUPINY

Všeobecné

VAŠE STRÁNKY



IT Academy



VITA - Virtual It Academy

KONTAKTY



Evka Rybárska



Jarmila Palenčárová



Stefan Orosi



Ivana Ivka Jasaňová



Hrá Word Blitz



Ivana Pavlíková



Martin Vanko



Lucia Kovačičová

4 h



Lošák Filip



Andrej Nejedlik



Gabika Zubrikova

SKUPINOVÉ KONVERZÁCIE



Vytvoriť novú skupinu

Hľadať



Home

PUBLIC

Stack Overflow

Tags

Users

FIND A JOB

Jobs

Companies

TEAMS

What's this?

Free 30 Day Trial

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

[Show all tag synonyms](#)

tomcat

Popular

Name

New

tomcat

for questions about Apache Tomcat (or simply Tomcat, formerly also Jakarta Tomcat) which is an open source Servlet Container developed by the...

40936 questions 11 asked today, 47 this week

tomcat7

Version 7.x (June 2010) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

5541 questions 10 asked this month, 143 this year

tomcat8

Version 8.x (August 2013 onwards) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

2576 questions 6 asked this week, 20 this month

tomcat6

Version 6.x (December 2006) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

1877 questions 11 asked this year

tomcat9

Version 9.x (August 2017 onwards) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

784 questions 8 asked this week, 32 this month

tomcat5.5

Version 5.5.x (August 2004) of the Apache Tomcat servlet container. Use only if your question is specifically related to features of this version.

252 questions 4 asked this year

embedded-tomcat-8

Embedded Apache Tomcat 8

209 questions 21 asked this year

maven-tomcat-plugin

The Tomcat Maven Plugin provides goals to manipulate WAR projects within the Tomcat servlet container.

202 questions 2 asked this year

embedded-tomcat-7

Questions about running Apache Tomcat 7 as an embedded server in...

tomcat8.5

Version 8.5.x (June 2018 onwards) of the Apache Tomcat servlet container

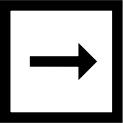
tomcat-valve

a type of component that can be inserted into Tomcat's...

tomcat-jdbc

about tomcat and jdbc working together

Mrkni na náš YouTube kanál a daj odber

 [WWW.YOUTUBE.COM/C/IT-ACADEMYSK](https://www.youtube.com/c/IT-ACADEMYSK) 