Enterprise Backend with Java

Lambda

Instructor – John Mitchell

Warmup Activity

- Install JDK 11
- Install IntelliJ IDEA Ultimate
- Install SQLite (Windows)
- Install SQLite Studio

Lambda

Java Course Flow

- Combining PreClass video information with Guided Projects
- 25 ish minutes with 5 minute break
- Question Thread in Slack. Zoom comments ignored

Lambda

Why Java

- Arguably more lines of code
- Over 3 billion devices
- Popular Intro language
- Many older, larger companies use it.

Lambda

A Brief History of Java

- First released 1996 by Sun Microsystems which became part of Oracle in 2010
- Version 1.1, 1.2, … 5, 6, 7, 8 (actually 1.8)
- Version 9 – current are radically different
- We are using Version 11 (LTS)

Lambda

A Brief History of Java

- Named for the Beverage or Island
- **Java** is to **Java**Script

like

Lamb is to **Lamb**da School

Car is to **Car**pet

Lambda

A Brief History of Java

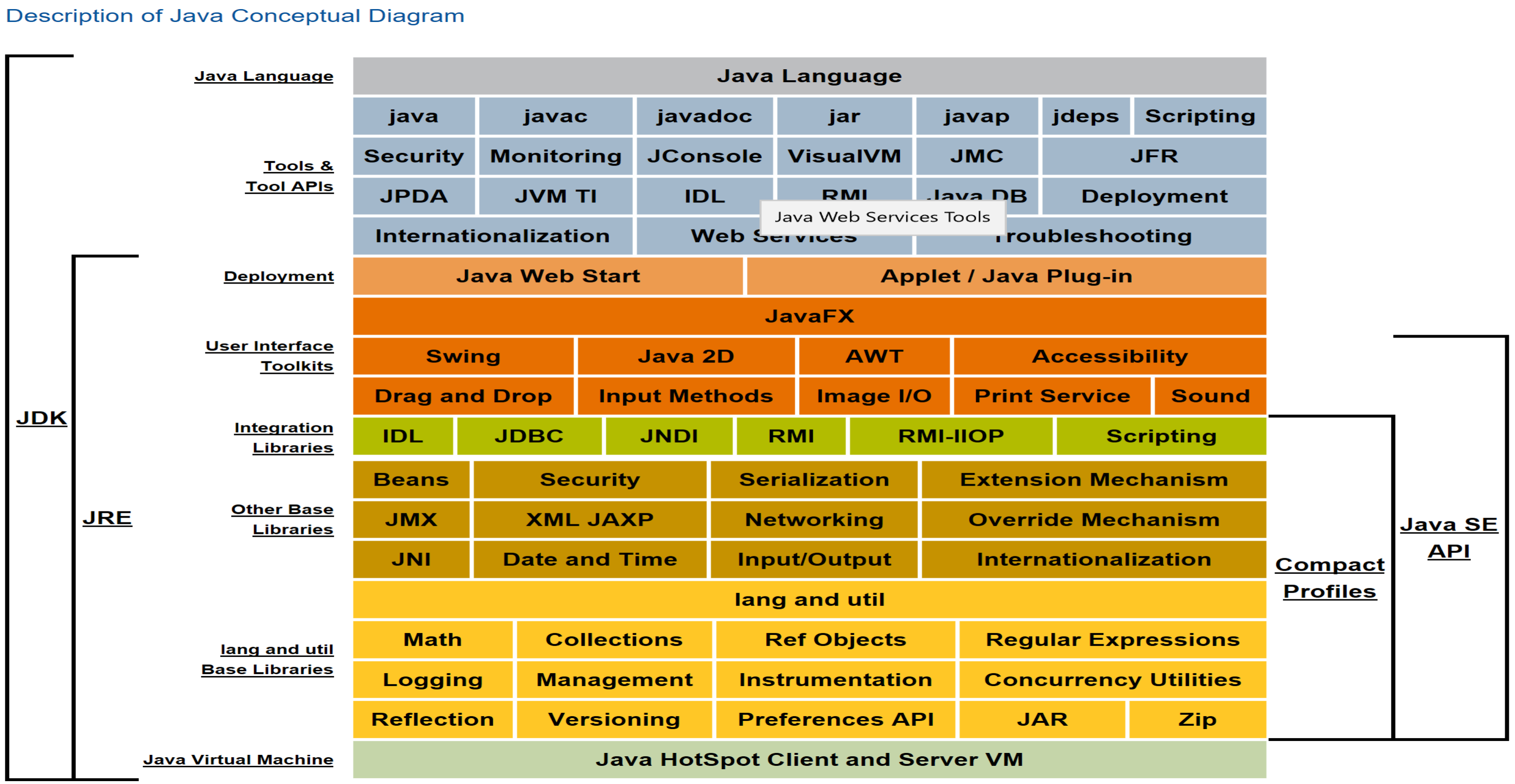
Dynamic web content through applets

Cross platform operations

- Force each web browser to have a java compiler (like JavaScript)
- Force compilation to each type of machine (like C++)
- **Bytecode** – compile to intermediate code and run on virtual machine

Lambda

Java Conceptual Diagram



Description of Java Conceptual Diagram

Layer	Components
Java Language	Java Language
Tools & Tool APIs	java · javac · javadoc · jar · javap · jdeps · Scripting Security · Monitoring · JConsole · VisualVM · JMC · JFR JPDA · JVM TI · IDL · RMI · Java DB · Deployment Internationalization · Web Services · Troubleshooting
Deployment	Java Web Start · Applet / Java Plug-in
User Interface Toolkits	JavaFX Swing · Java 2D · AWT · Accessibility Drag and Drop · Input Methods · Image I/O · Print Service · Sound
Integration Libraries	IDL · JDBC · JNDI · RMI · RMI-IIOP · Scripting
Other Base Libraries	Beans · Security · Serialization · Extension Mechanism JMX · XML JAXP · Networking · Override Mechanism JNI · Date and Time · Input/Output · Internationalization
lang and util Base Libraries	lang and util 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 Web Services Tools

https://docs.oracle.com/en/java/javase/11/

Lambda

Java Development

- Core Java
- Android
- Web Java and Spring

Lambda

Java Development

- Text Editor -> JDK -> JVM(JRE)
- Full IDEs
  - NetBeans
  - Eclipse
  - IntelliJ IDEA
- Special IDEs
  - BlueJ
  - Greenfoot
  - JGasp

Lambda

Java Employee Console App

Create directory	`EmployeeAPP/src/employeeApp`
Create text file	`Employee.java`
Create text file	`Company.java`
Create text file	`Healthplan.java`
Create text file	`Main.java`

Load PrePopulated data code

Lambda

Java Employee Console App (notes)

Fields, Constructors, Methods

Private, Public, <nothing>

Class variables, methods plus static

Variable Scope

Class, Local, Block

Java Employee Console App

Concat First and Last Name into a Name Field

Give Employees Raise

Give Raise Method

Estimate 401K Contribution

Lambda

Java Hello World Console App

```
javac employeeApp/*.java


jar cvfe EmpApp.jar employeeApp.Main employeeApp/*.class
```

- C – create new archive file with given name
- V – generate verbose output
- F – specifies the jar output file to be created
- E – sets the main class also called the entry point

```
java -jar EmpApp.jar
```

Lambda

Primitive Data Types

Type	Description	Default	Size	Example Literals
boolean	true or false	false	1 bit	true, false
byte	twos complement integer	0	8 bits	(none)
char	Unicode character	\u0000	16 bits	'a', '\u0041', '\101', '\\', '\"', '\n', 'ß'
short	twos complement integer	0	16 bits	(none)
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d

Java Data Types Console App

Compound Operators

```
a += b => a = a + b // +, -, *, /, %
```

Increment Operators

```
a++ => a = a + 1

a—, —a, ++a
```

Lambda

Snack Bar Project

Snack has id, name, quantity, cost, vending machine id
Snack can set name, get quantity, add quantity given quantity, buy snack given quantity, get total cost given a quantity.

Vending machine has id, name
Vending Machine set and return name

Customer has id, name, cash on hand
Customer can add cash, buy given total cash used in purchase, get and set name, get cash on hand.

- Instantiate 2 customers
  - Jane with $45.25
  - Bob with $33.14
- Instantiate 3 Vending Machines
  - Food
  - Drink
  - Office
- Instantiate 5 snacks
  - In Vending Machine Food
    - 36 Chips at $1.75
    - 36 Chocolate Bar at $1.00
    - 30 Pretzel at $2.00
  - In Vending Machine Drink
    - 24 Soda at $2.50
    - 20 Water at $2.75

Snack Bar Project

Customer 1 buys 3 of snack 4. Print Customer 1 Cash on hand. Print quantity of snack 4.
Customer 1 buys 1 of snack 3. Print Customer 1 Cash on hand. Print quantity of snack 3.
Customer 2 buys 2 of snack 4. Print Customer 2 Cash on Hand. Print quantity of snack 4.
Customer 1 finds $10. Print Customer 1 Cash on Hand.
Customer 1 buys 1 of snack 2. Print Customer 1 Cash on Hand. Print quantity of snack 2.
Snack 3 gets 12 more. Print quantity of snack 3.
Customer 2 buys 3 of snack 3. Print Customer 2 Cash on hand. Print quantity of snack 3.