J8

Ultimate Java

Ultimate Java

J8 - Version: 1

🕐 6 days Course

Description:

Java8 is the biggest change in the java language, it changesfundamental concepts that we have in java programming. We have new concept of lambda expressions that changes the nature of the language. We have new possibilities of extending the java framework by adding default methods to interfaces.

Working with collections has completely changed by using the Stream API. In this one day workshop we will examine the features that will affect almost any code line that we have in java?

Intended audience:

java developers

Prerequisites:

Working in Object Oriented Programming

Objectives:

Learn how to develop java applications using the tools and frameworks of java 8

Topics:

Introduction to Java

- History
- Key Features
- Java platform independence
- Byte Code
- JVM
- Java API
- Java memory model
- Garbage collector

Java basics

- Classes
- Compiling and running
- Data types
- Strings and StringBuffers
- Arrays
- Operators
- Conditional Statements
- loops
- packaging and compiling

º Eclipse workshop - Appendix

Object Oriented programming in Java

- Referencing and Instantiation
- Object members and methods

- Constructors
- Class members and methods
- The Final keyword
- Access control
- Inner Classes
- The java.lang.Object class
- Polymorphism – Overloading and Overriding.
- Casting and safe casting
- Final classes and methods
- Abstract classes and methods
- Interfaces

Exceptions

- Exceptions vs. traditional error handling.
- Exceptions terminology
- Errors and exceptions
- Available methods
- Nesting blocks
- The finally block
- Creating user-defined exceptions

Extended Interface

- Regenerate java with extended interfaces
- Extended interface and multiple inheritence
- mixins in java

Reflection & Annotation

- The java.lang.Class
- Dynamic instantiation
- modifiers
- Method invocation
- Design Issues and limitations
- Annotation

Lambda expressions

- Anonymus inner class
- Lambda expressions
- Functional interfaces
- Lambda expression best practices

Multithreaded programming in Java

- Threads overview
- Creating threads in Java
- The thread life cycle
- Naming threads
- The Timer and TimerTask classes
- Synchronization overview
- Synchronized blocks
- Synchronized methods
- Synchronized class methods
- Cooperation using wait() and notify()
- Priorities - appendix
- Deadlocks - appendix
- Daemon threads - appendix
- Util concurrent package

º Executores

º Callback function

º Future design pattern

º Executores Service

º ThreadPool

º Thread pool lab

º Barrier and phasers

º Semaphore

º Read & Write Lock

º Util Concurrent 2

    - Fork & Join framework

    - Phasers

    - Transfer Queue

Collections

• The collection framework classes

• Iterator

• List and implementations

• Set and implementations

• Sorting issues (Comparator and Comparable).

• Map and implementations

• Read-only and synchronized collections

• Historical implementations

• Generics

• Streams and parallel programming with collections

    º The purpose of lambda expressions

    º What are streams

    º parallel streams

    º unbound streams

    º stream API

- Additional issues - appendix
  - º For each loop
  - º Autoboxing
  - º EnumSet & EnumMap
- Concurrent Collections

I/O

- I/O overview
- Binary Streams
- FileInputStream/FileOutputStream
- Working with buffers
- RandomAccessFile
- Character streams
- Console IO
- The java.io.File class
- Scanner - Appendix
- Networking
  - º Ports
  - º Sockets
  - º The Server side
  - º The Client side
  - º The Java Networking Model
  - º Communication using streams
  - º Other types of socket - Appendix
- NIO
  - º I/O overview
  - º Buffers
  - º Channels
  - º Selectors
  - º Selection Keys

º NIO lab

º NIO2

   - Using channels for multicasting

   - The File revolution of NIO2

   - Asynchronous I/O