Java Features

Mr. Pankaj Jagasia

Java 5 Features (Tiger)

- Accessing static methods and variables using ***Static import***
- Method Over-ridding additional Support
  - @Override annotation
  - Covariant returns (allows you to use a subclass return type while overriding)
- Method overloading issues
  - Auto boxing & unboxing
  - Var-arg parameter method
- Collection programming enhancement
  - For-each loop / enhanced for loops
  - Generics (Typed Parameter, Type Safety)

- New Style Class
  - Enum
  - Annotations
- C Languages style enhacement
  - C style formatting I/P and O/p
  - Scanner class
- Collection Enhancement
  - Queue type Collections
  - Java.lang.concurrent package

Java 6 (Mustang)

- Primarily to Improve Performance and to overcome the lags in Java 5
- Classes introduced
  - Console
  - NavigableSet
  - NavigableMap

Java 7 Features (Dolphin)

- Execution flow change with respective to main method.
  - Upto Java 7 if no main method is there start execution using Static block. Prior to Java 7 static variables initialized then static block fired and then check for the main method. In Java 7 first check if main method is present or not if not raise an exception, else continue with the static variables, static block, and then main.
- Underscore in number literal is allowed for eg int tmp = 1_00_000;
- Switch allows String
- Try with resources
- Catch with multiple exceptions
- Bug fixing for var-arg in overloading
- Auto Boxing casting improved (direct assignment without having to cast in Wrapper type)
- Generic type inference (ArrayList<String> list = new ArrayList<>());

Java 8 Features (No codenames from this version)

- Interface issues resolution
  - Default method support in Interface
  - Static method in interfaces
- Functional Programming
  - Functional interface
  - @FunctionalInterface
  - Lambda expressions
  - Method & Constructor Reference (::)
- Stream API (bulk operations on Collection)
  - forEach() & splIterator() method in Iterable interface
  - Stream API (java.util.stream) [filtering, mapping]
  - Functional API (java.util.function)

Predicate	BiPredicate
Function	BiFunction
Consumer	BiConsumer
Supplier	-
UnaryOperator	BinaryOperator

- New Date & Time API solves issues of thread safety (in association with joda.org)
- Multicore processing
  - Base64 API
- Security API
- Annotation Enhacement
  - RepeatedAnnotation
  - TypeAnnotation
  - PluggableAnnotation
- Improvement done in HashMap (Key collision implemented using BinaryTree)
- Removal of jdbc:odbc bridge
- Adding JavaScript support using Nashorn JavaScript Engine

Ideology behind Java 8

- Simplify Programming
- Utilizing the benefits of Functional Programming
- To utilize the capabilities of Multi-core-processors, and enable parallel processing using Streams

Java 9 features

- **Java 9 REPL - Read Evaluate Print Loop (JShell) (It is used to** execute and test any Java Constructs like class, interface, enum, object, statements etc.)
- **Factory Methods for Immutable List, Set, Map and Map.Entry**
- **Private methods in Interfaces**
- **Java 9 Module System**
- **Process API Improvements (**java.lang.ProcessHandle & java.lang.ProcessHandle.Info) allows to manage and control OS processes
- **Try With Resources Improvement**
- **CompletableFuture API Improvements**
- **Reactive Streams (used for Publish/Subscribe framework)**
- **Diamond Operator for Anonymous Inner Class (**return new List(emp){ })
- **Optional Class Improvements**
- **Stream API Improvements**
- **Enhanced @Deprecated annotation**
- **HTTP 2 Client**
- **Multi-Resolution Image API**

Java 10 features

- **Time-Based Release Versioning**
- **Local-Variable Type Inference** (var numbers = List.of(1, 2, 3, 4, 5))
- **Experimental Java-Based JIT Compiler** (Dynamic change of JIT)
- **Application Class-Data Sharing** (reduces boot time for independent apps)
- **Parallel Full GC for G1** (performance improvement by using parallel invocation)
- **Garbage-Collector Interface** (for future enhancements of GC)
- **Additional Unicode Language-Tag Extensions** (cu for currency,fw first day of week, rg region override, tz timezone)
- **Root Certificates**
- **Thread-Local Handshakes** (Internal enhancement to improve JVM performance)
- **Heap Allocation on Alternative Memory Devices**

Java 11 features

- **Running Java File with single command**
- **Java String Methods Enhanced**
- **Local-Variable Syntax for Lambda Parameters (var x, var y)-> x + y;**
- **Nested Based Access Control** (Reflection issues resolved to access private members of enclosing class from inner class)
- **Remove the Java EE and CORBA Modules**
- **Flight Recorder**
- **HTTP Client (Improvements)**
- **Reading/Writing Strings to and from the Files** (Files.readString(path);)

Java 12 features

- Switch expressions (switch(name){case CDAC,PUNE->4;case MUMBAI->6;})
- **File.mismatch method**
- **Compact Number Formatting (eg, 2952 will be shown as 2.6k)**
- **Java Strings New Methods**
- **Other improvements for Performance**