

Java Fundamentals

Java 7: Project Coin

- Improved exception handling
- Strings as switch selectors
- Diamond syntax
- TWR (Try With Resource)
- Some more . . .

Java 7: NIO.2

- Easy handling of files , file systems and permissions
- New file visitor interface to traverse filesystems
- Watch services
- The most important one: asynchronous I/O: Futures style and callback style

Dependency Injection

Quick intro

- Inversion of control and Dependency Injection
- JSR-330
- Some examples

The background of the image is a spiral-bound notebook with a light blue textured cover. The spiral binding is visible along the top edge.

Modern Concurrency

Basic Concepts (to be completed)

- Safety
- Liveness
- Performance
- Reusability

Building blocks (I)

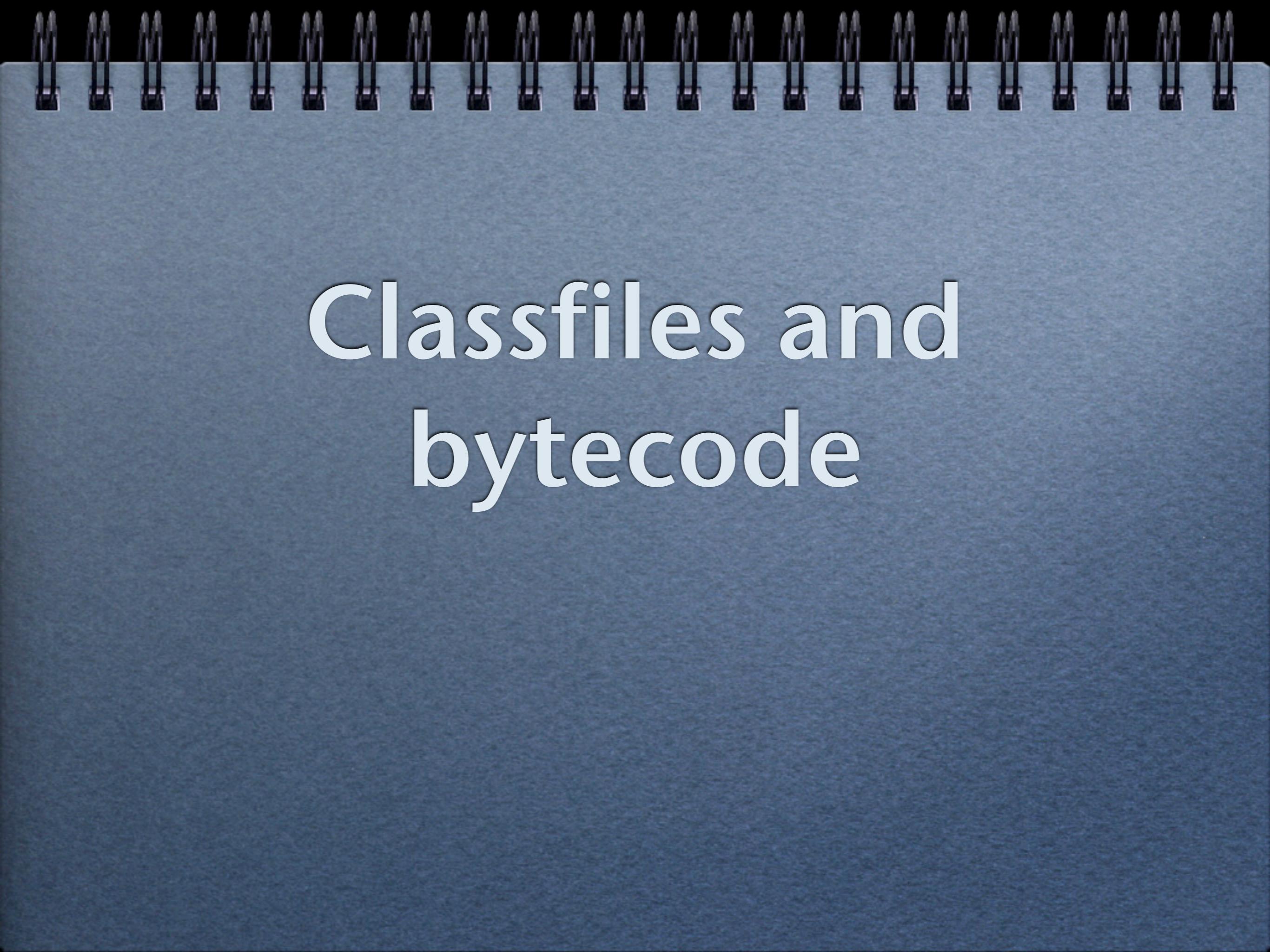
- Atomic class
- Locks
- CountdownLatch
- ConcurrentHashMap
- CopyOnWriteArrayList

Building blocks (II)

- Queues
- Blocking queues
- Transferqueues (added in Java 7)
- Thread executors
- Futures

ForkJoin framework

- Maximize processing capabilities
- Subtasks communicating each other
- Map-Reduce (divide and conquer)
 - Split
 - Collect



Classfiles and bytecode

Classloading

- Loading
 - Reading the classfile
- Linking
 - verification
 - Preparation
 - Resolution

Classloaders

- Bootstrap classloader
 - usually native code
- Extension classloader
 - Standard extensions
- Application classloader
 - Application classes

Method handles

- New API for invoking methods
- What is a MethodHandle?
- Comparative example
- Why choose them?
 - Full access to all methods (context)
 - Typesafe at runtime. No extra permgen

Understanding classfiles

- Binary blobs
- Hard to work with it directly
- A quick intro to javap
- Examining classfiles

A quick look to bytecode

- Intermediate representation
- Output from javac (not highly optimized)
- JIT compiler will optimize it!

A quick look to bytecode

- Runtime environment
- Stack based
- Overview of the opcodes: structure
 - Name
 - Args
 - Stack Layout
 - Description

A quick look to bytecode

- Load/Store opcodes
- Arithmetic opcodes
- Control execution opcodes
- Invocation opcodes
- Platform operations

A quick look to bytecode

- To be completed with . . .
- New invokedynamic bytecode

Coming soon . . .

Coming soon . . .

- Understanding performance
- Polyglot programming
 - Alternative JVM languages
- Unit testing, TDD, CI
- More on modern concurrency
 - Actors, STM