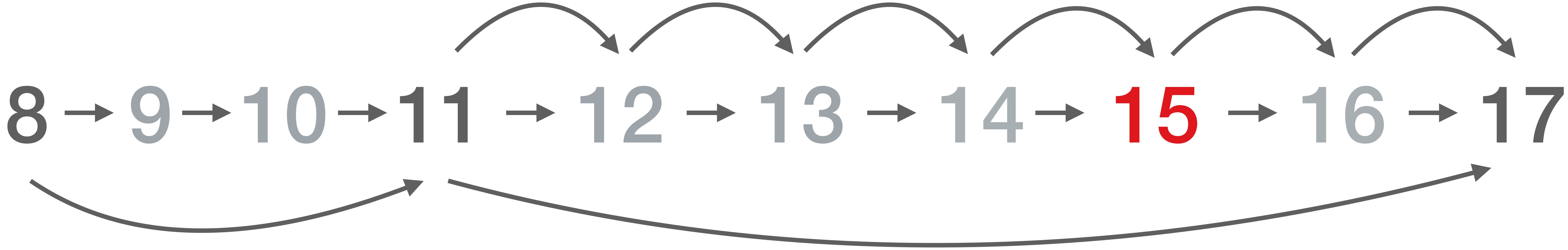
Java 15: What's in It for Me?

@Sander_Mak

Keeping up with Java



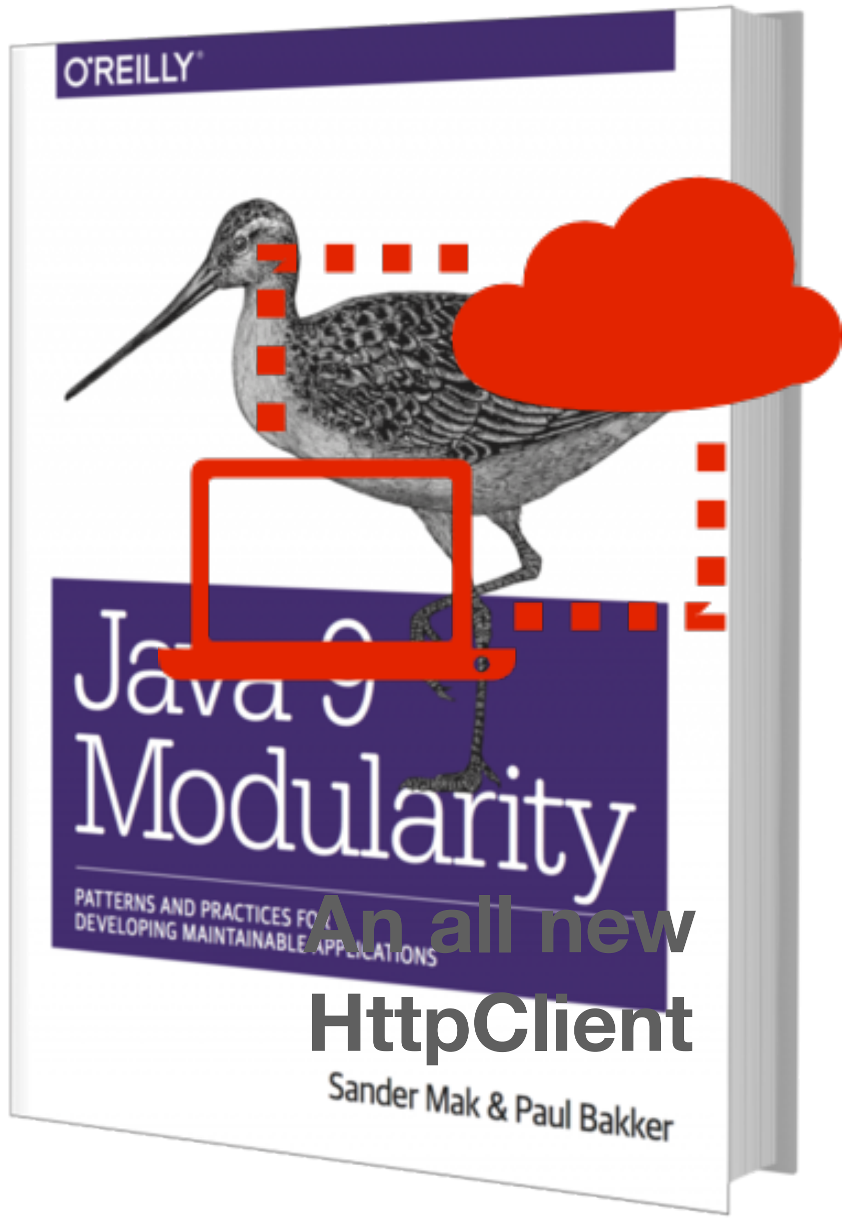
8 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 16 → 17

Keeping up with Java

Module System

Type inference with var

An all new HttpClient



O'REILLY

Java 9 Modularity

PATTERNS AND PRACTICES FOR DEVELOPING MAINTAINABLE APPLICATIONS

Sander Mak & Paul Bakker

Switch Expressions

Continuous JVM Improvements

@Sander_Mak

A small but very useful feature...

I call it my 'billion dollar mistake'

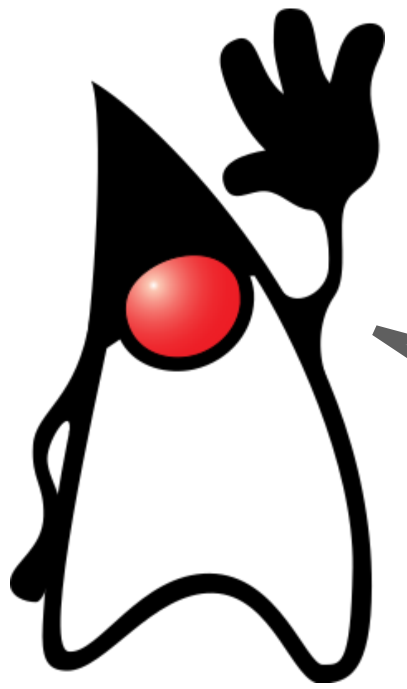
```
Exception in thread "main" java.lang.NullPointerException
        at Example.main(Example.java:5)
```

```
a().b().c.doSomething();
```

```
A a = a();
B b = a.b();
C c = b.c;
c.doSomething();
```

`-XX:+ShowCodeDetailsInExceptionMessages`

```
Exception in thread "main" java.lang.NullPointerException:
   Cannot invoke "Example$C.doSomething()" because
   "Example$A.b().c" is null
        at Example.main(Example.java:5)
```



Enabled by default in Java 15!

@Sander_Mak

Today's topics

1 Text Blocks

2 Records

3 Sealed Classes

4 Pattern Matching

... some odds & ends

Today's topics

1. Text Blocks
2. Records
3. Sealed Classes
4. Pattern Matching

... some odds & ends

Text Blocks

```
String json = "{\n  \"firstName\": \"Sander\",\n   \"lastName\"   ...
```

```
String json = "{\n"
   + "  \"firstName\": \"Sander\",\n"
   + "  \"lastName\": \"Mak\"\n"
   + "}";
```

```
String json = new StringBuilder()
        .append("{\n")
        .append("  \"firstName\": \"Sander\",\n")
        .append("  \"lastName\": \"Mak\"\n")
        .append("}")
        .toString();
```

@Sander_Mak

Text Blocks

Multi-line Strings in Java source code

```
String json = """
   {
     "firstName": "Sander",
     "lastName": "Mak"
   }""";
```

DEMO

Text Blocks @ Picnic?

```java
@Operation(description = """
   Find all users that satisfy the following conditions:
   * Condition 1
   * Condition 2
   * Condition 3
   _Note_: only active users are returned.""")
@GetMapping
Flux<User> findUsers(@RequestBody FindRequest request) {
   // ...
}
```

OpenAPI documentation

```java
assertThat(json).isEqualTo("""
   {
     "productId": "123",
     "name": "Milk",
     "active": false
   }""");
```

Assertion values in code rather than files

```java
String pushMessage = """
  "GCM": {
    "data": {
      "type": "text-push",
      "title": "%s",
      "message": "%s"
   }
  }""".formatted(title, message)
```

Simple templated values

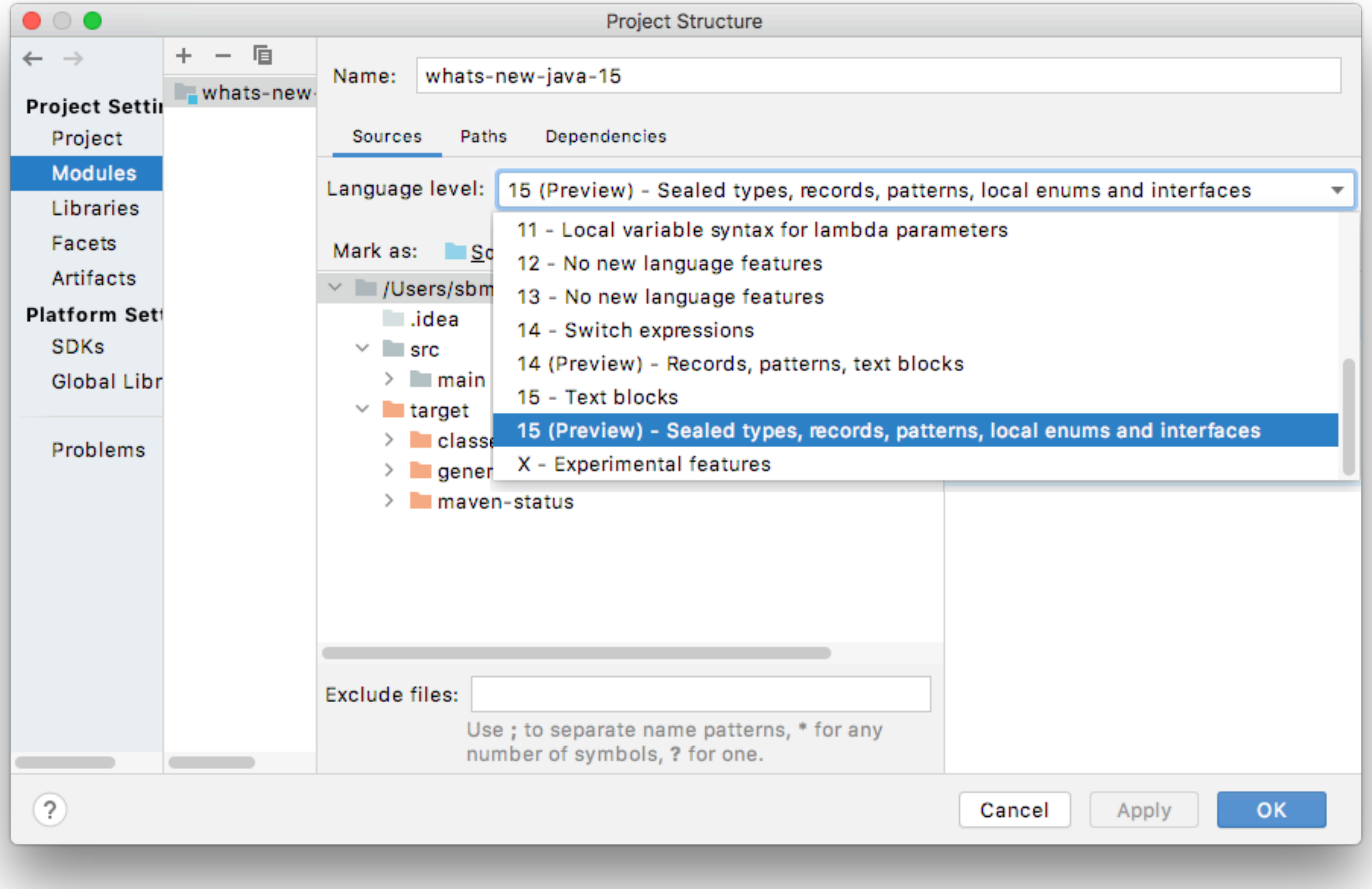
Today's topics

1. Text Blocks
2. Records
3. Sealed Classes
4. Pattern Matching

... some odds & ends

Records

One of the three *preview features* we'll look at.



Project Structure

Name: whats-new-java-15

Language level: 15 (Preview) - Sealed types, records, patterns, local enums and interfaces

- 11 - Local variable syntax for lambda parameters
- 12 - No new language features
- 13 - No new language features
- 14 - Switch expressions
- 14 (Preview) - Records, patterns, text blocks
- 15 - Text blocks
- 15 (Preview) - Sealed types, records, patterns, local enums and interfaces
- X - Experimental features

Exclude files:

Use ; to separate name patterns, * for any number of symbols, ? for one.

@Sander_Mak

How to think about Records

A new restricted form of a class to *model data as data*

Nominal tuples

Immutable

How *not* to think about Records

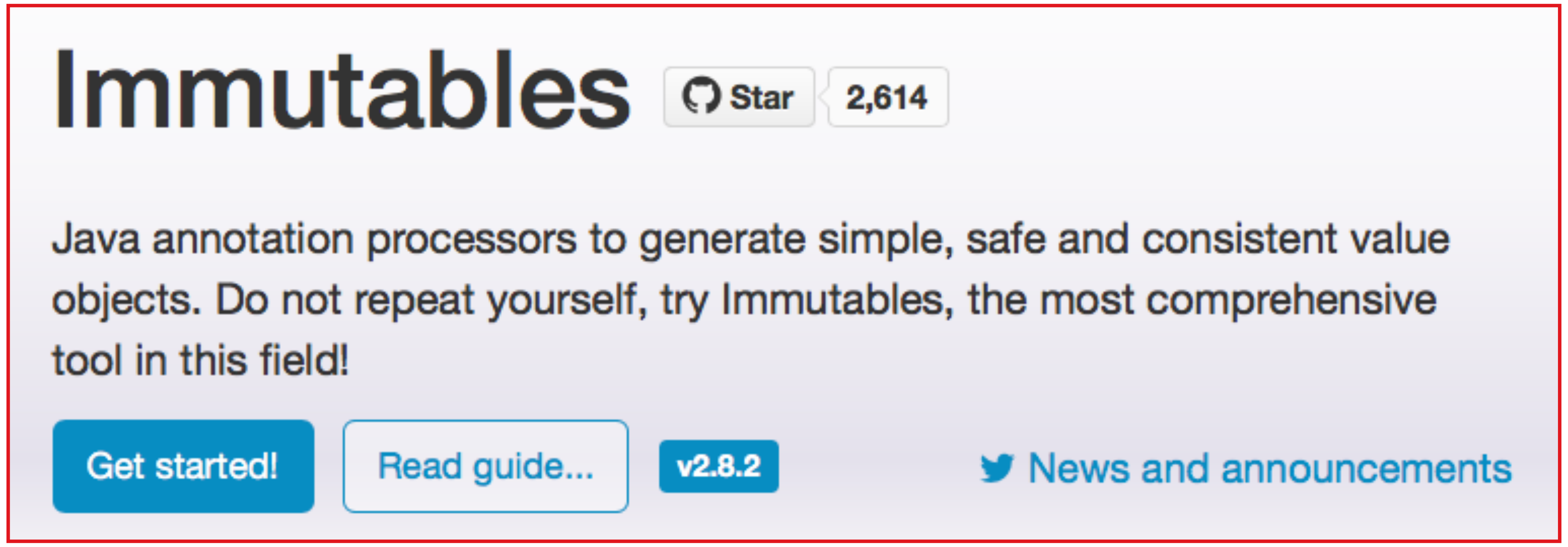
DEMO

Boilerplate reduction mechanism

JavaBeans replacement

Value types (memory layout/identity)

Records @ Picnic?



Immutables

Java annotation processors to generate simple, safe and consistent value objects. Do not repeat yourself, try Immutables, the most comprehensive tool in this field!

http://immutables.org

DTOs

Document classes

Compound Map keys

Intermediate results in stream/reactive pipelines

Today's topics

1 Text Blocks

2 Records

3 Sealed Classes

4 Pattern Matching

... some odds & ends

Why Sealed Classes?

Goal: limit unrestriced inheritance

Use final? No inheritance at all

Supertype + subtypes in same package: limiting

```
public abstract class Option<T> {

    private Option() {}

    public final static class Some<T> extends Option<T> { ... }

    public final static class Empty<T> extends Option<T> { ... }

    // ...
}
```

Sealed classes

```java
public sealed class Option<T> permits Some, Empty {
  // ..
}
```

```java
public final class Empty<T> extends Option<T> {
  // ..
}
```

```java
public final class Some<T> extends Option<T> {
  // ..
}
```

Sealed classes

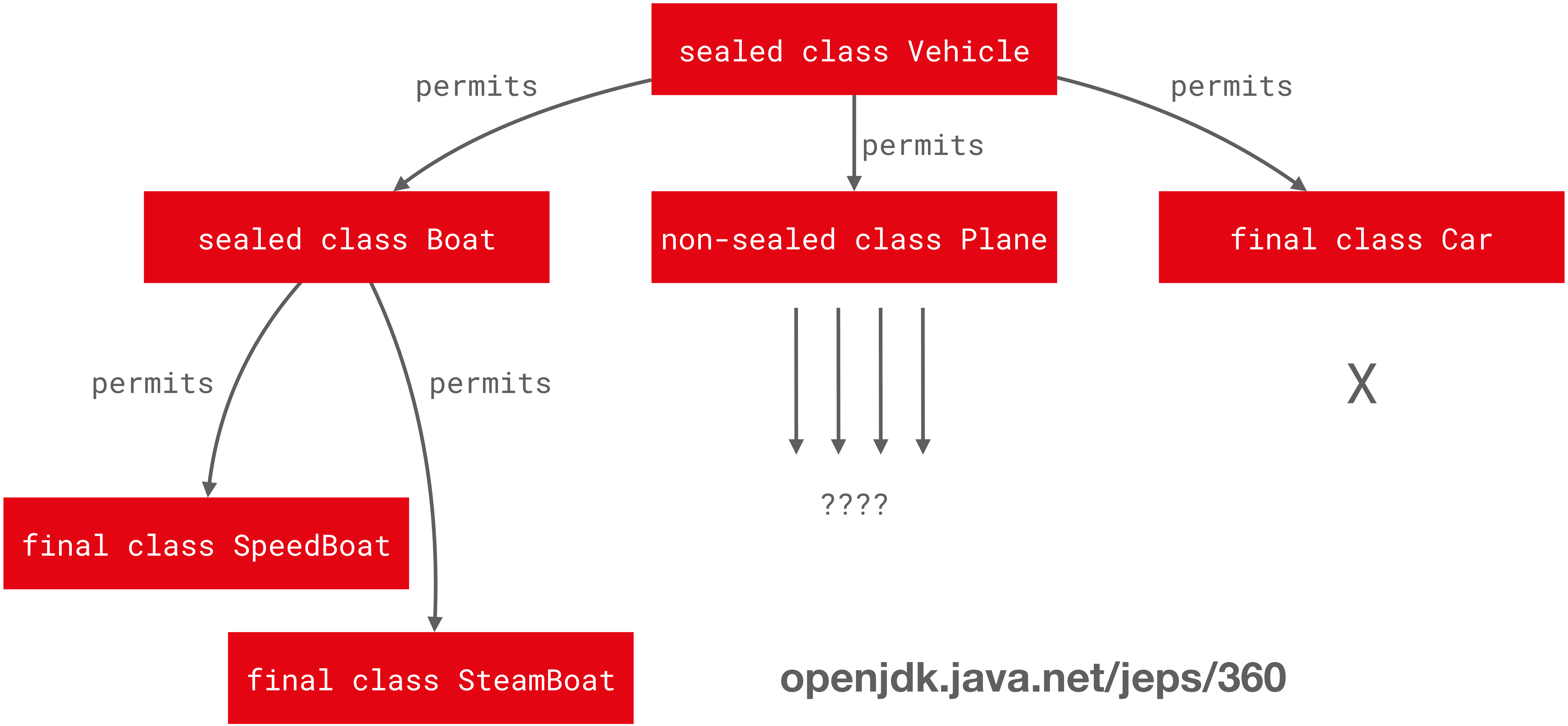
With subclasses in same source file, `permits` not required:

```
public sealed class Option<T> {
  // ..
}

final class Empty<T> extends Option<T> {}
final class Some<T> extends Option<T> {}
```

DEMO

Sealed classes



openjdk.java.net/jeps/360

@Sander_Mak

Today's topics

1 Text Blocks

2 Records

3 Sealed Classes

4 Pattern Matching

... some odds & ends

Pattern Matching

* with `instanceof`

```
if (obj instanceof String) {
    String text = (String) obj;
    text.startsWith("Java 15 rules!");
} else {
    // not a String
}
```

- Test
- Cast
- New variable

```
if (obj instanceof String text) {
    text.startsWith("Java 15 rules!");
} else {
    // still not a String
}
```

- Type pattern
- Binding variable

Pattern Matching

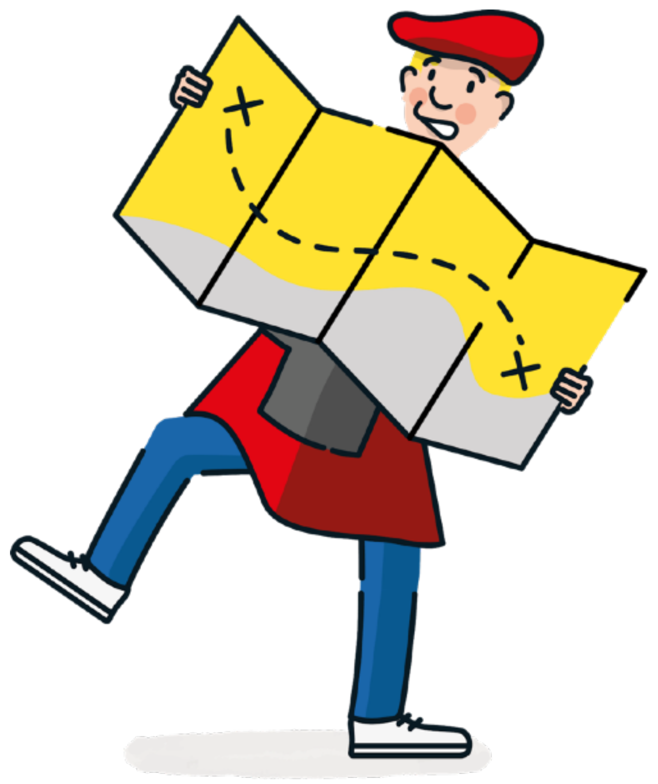
DEMO

Familiar use-case for Pattern Matching

```
@Override
public boolean equals(Object o) {
    return (o instanceof MyClass) &&
        ((Myclass) o).name.equals(name);
}
```

```
@Override
public boolean equals(Object o) {
    return (o instanceof MyClass m) &&
        m.name.equals(name);
}
```

Beyond Pattern Matching with instanceof



```
int value = switch (object) {
   case BigDecimal b -> b.intValue();
   case String s -> Integer.parseInt(s);
   case Integer i -> i;
}
```

Records & Pattern Matching & Switch Expressions

```
int eval(Expr e) {
    return switch (e) {
        case ConstantExpr(var i) -> i;
        case PlusExpr(var a, var b) -> eval(a) + eval(b);
        case MulExpr(var a, var b) -> eval(a) * eval(b);
        case NegExpr(var e) -> -eval(e);
        // no default needed, Expr is sealed
    }
}
```

Today's topics

1. Text Blocks
2. Records
3. Sealed Classes
4. Pattern Matching

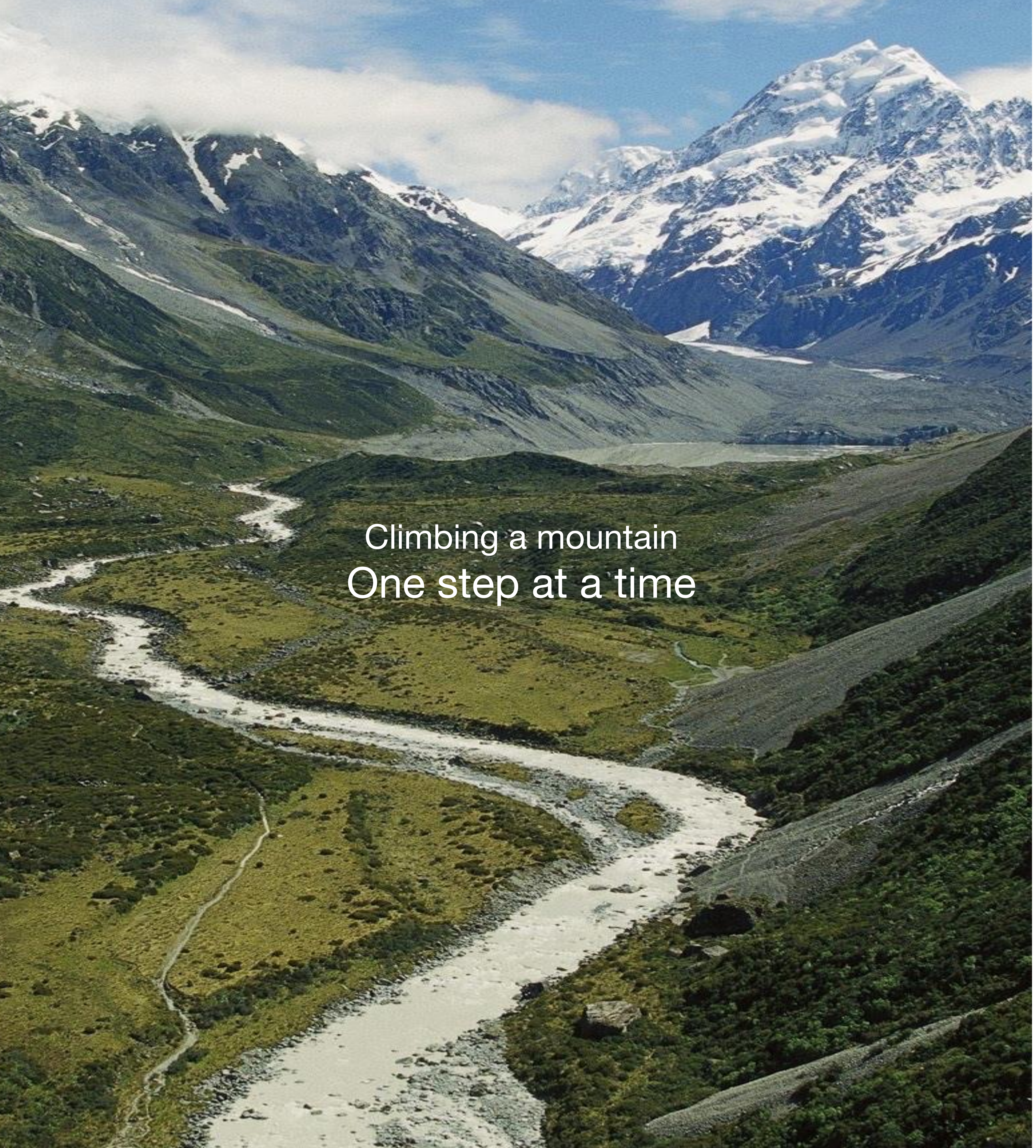
... some odds & ends

But wait, there's more!

- ZGC marked non-experimental
- Shenandoah GC marked non-experimental
- Hidden classes
- Nashorn JS engine removed

Java Enhancement Proposals (JEPs)

openjdk.java.net/projects/jdk/15/



Climbing a mountain
One step at a time

Help build Picnic Tech:

join.picnic.app

**Building the best milkman on earth
serving millions of families**

PLURALSIGHT

bit.ly/ps-sander