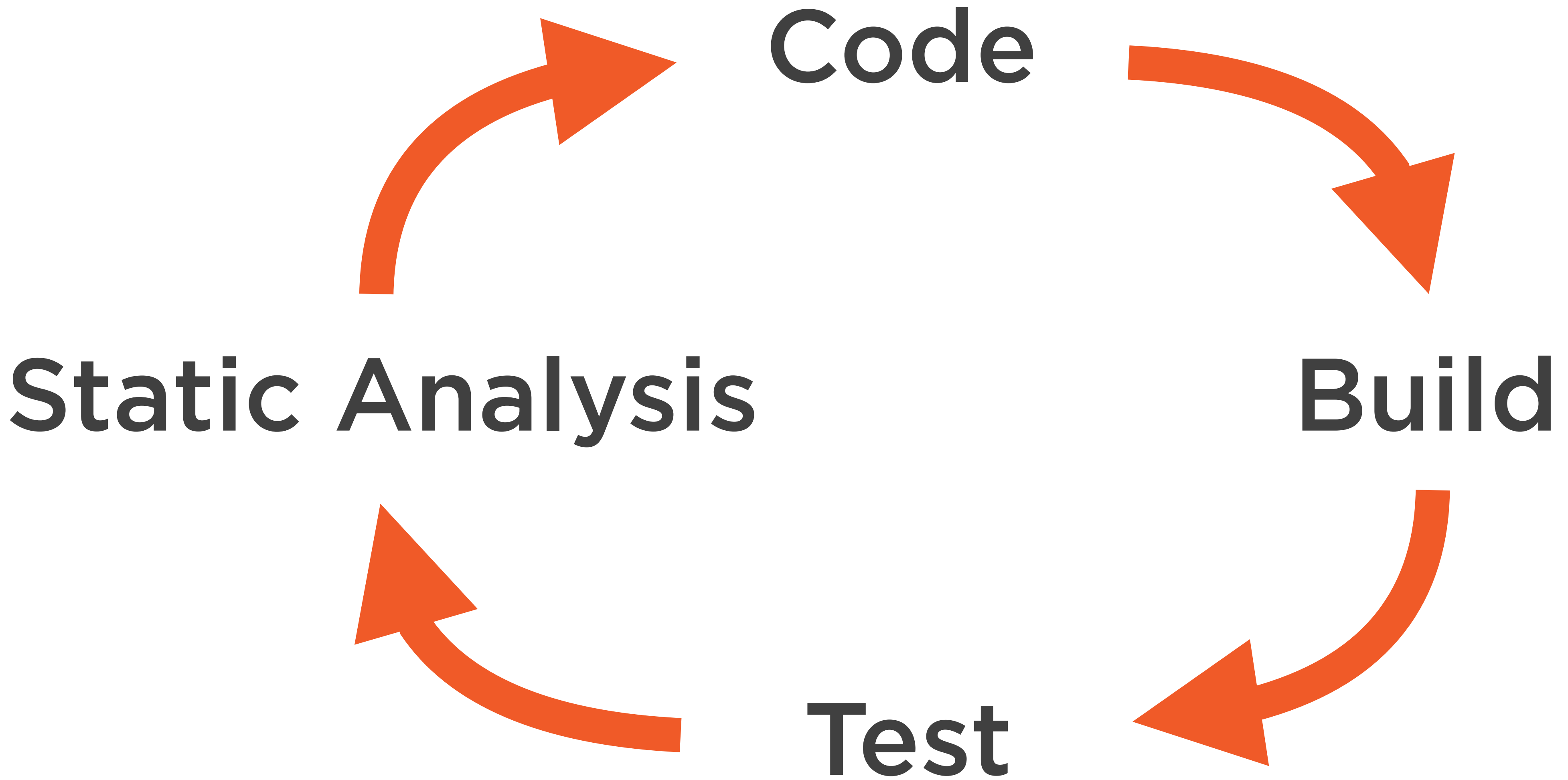
Practices and Common Tools

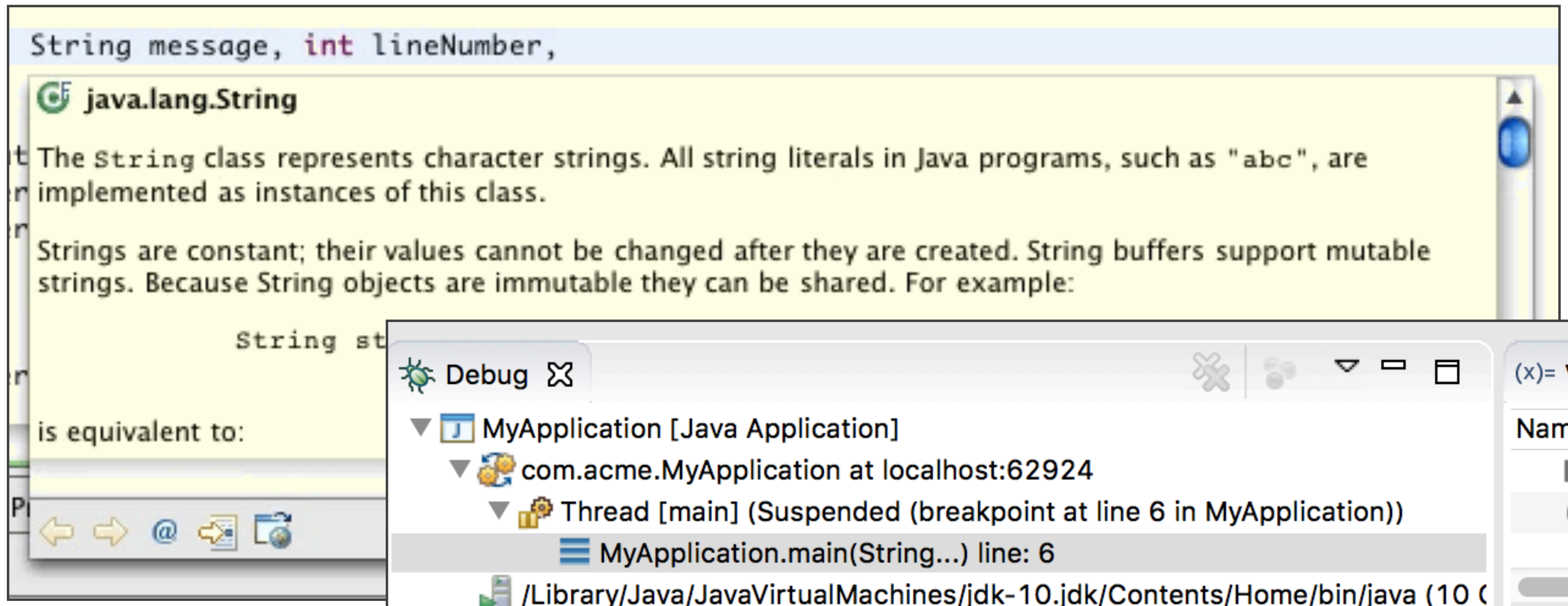
Sander Mak
FELLOW & SOFTWARE ARCHITECT
@Sander_Mak

Java Development Lifecycle



IDEs

Inline documentation



```
String message, int lineNumber,
```

java.lang.String

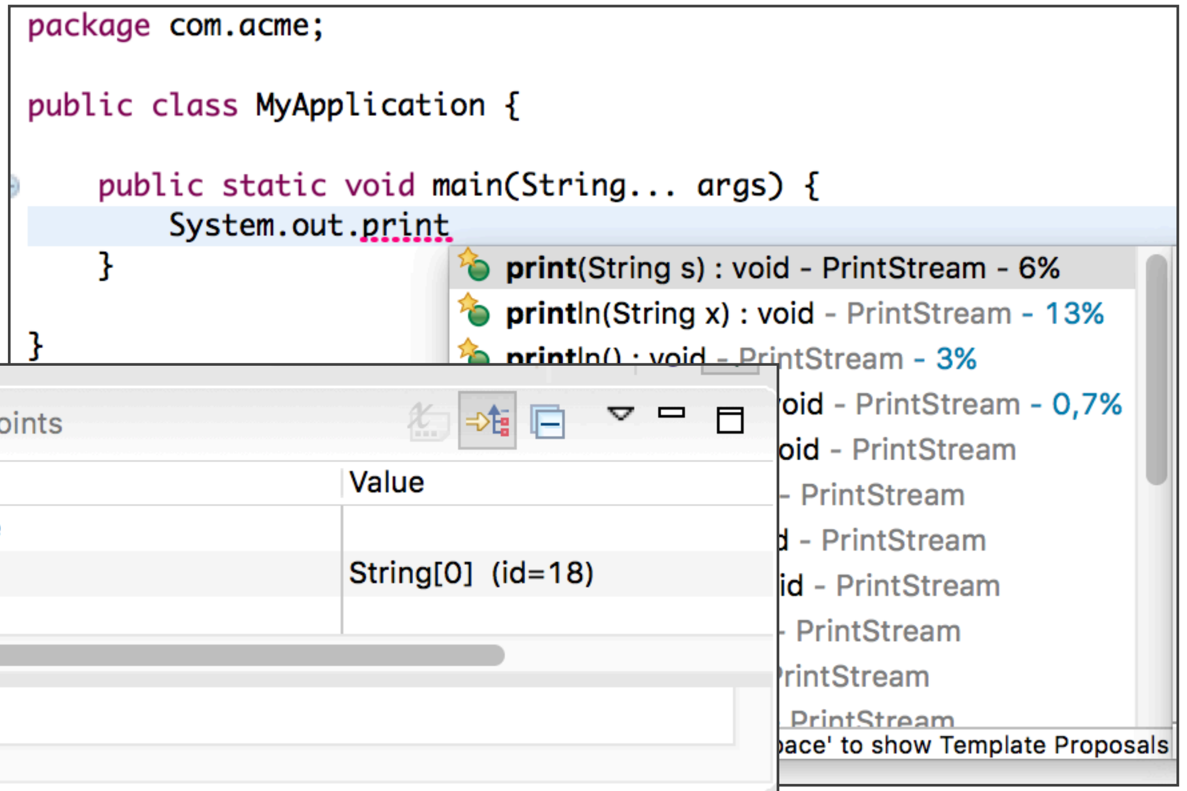
The `String` class represents character strings. All string literals in Java programs, such as `"abc"`, are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
String st
```

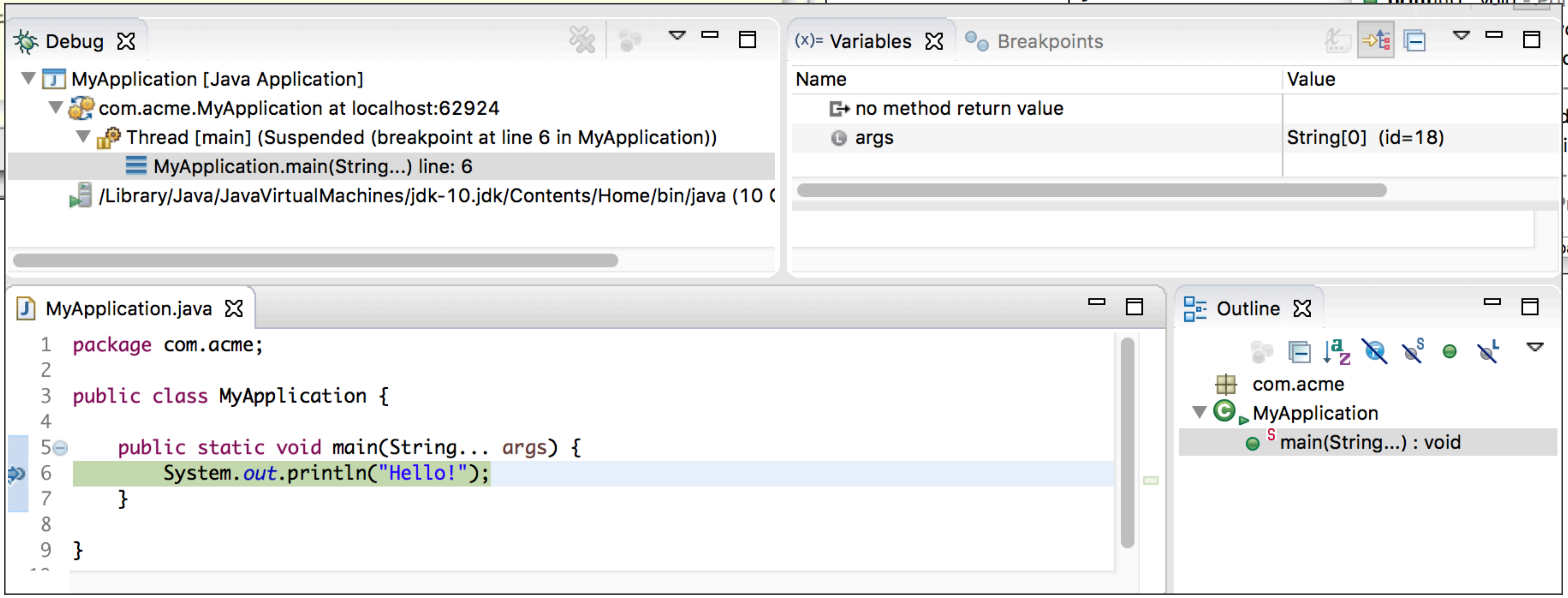
is equivalent to:

Code completion



```java
package com.acme;

public class MyApplication {

    public static void main(String... args) {
        System.out.print
    }

}
```

```
print(String s) : void - PrintStream - 6%
println(String x) : void - PrintStream - 13%
println() : void - PrintStream - 3%
oid - PrintStream - 0,7%
oid - PrintStream
- PrintStream
d - PrintStream
id - PrintStream
- PrintStream
rintStream
PrintStream
ace' to show Template Proposals
```



```
MyApplication [Java Application]
  com.acme.MyApplication at localhost:62924
    Thread [main] (Suspended (breakpoint at line 6 in MyApplication))
      MyApplication.main(String...) line: 6
  /Library/Java/JavaVirtualMachines/jdk-10.jdk/Contents/Home/bin/java (10 (
```

Name	Value
no method return value	
args	String[0] (id=18)

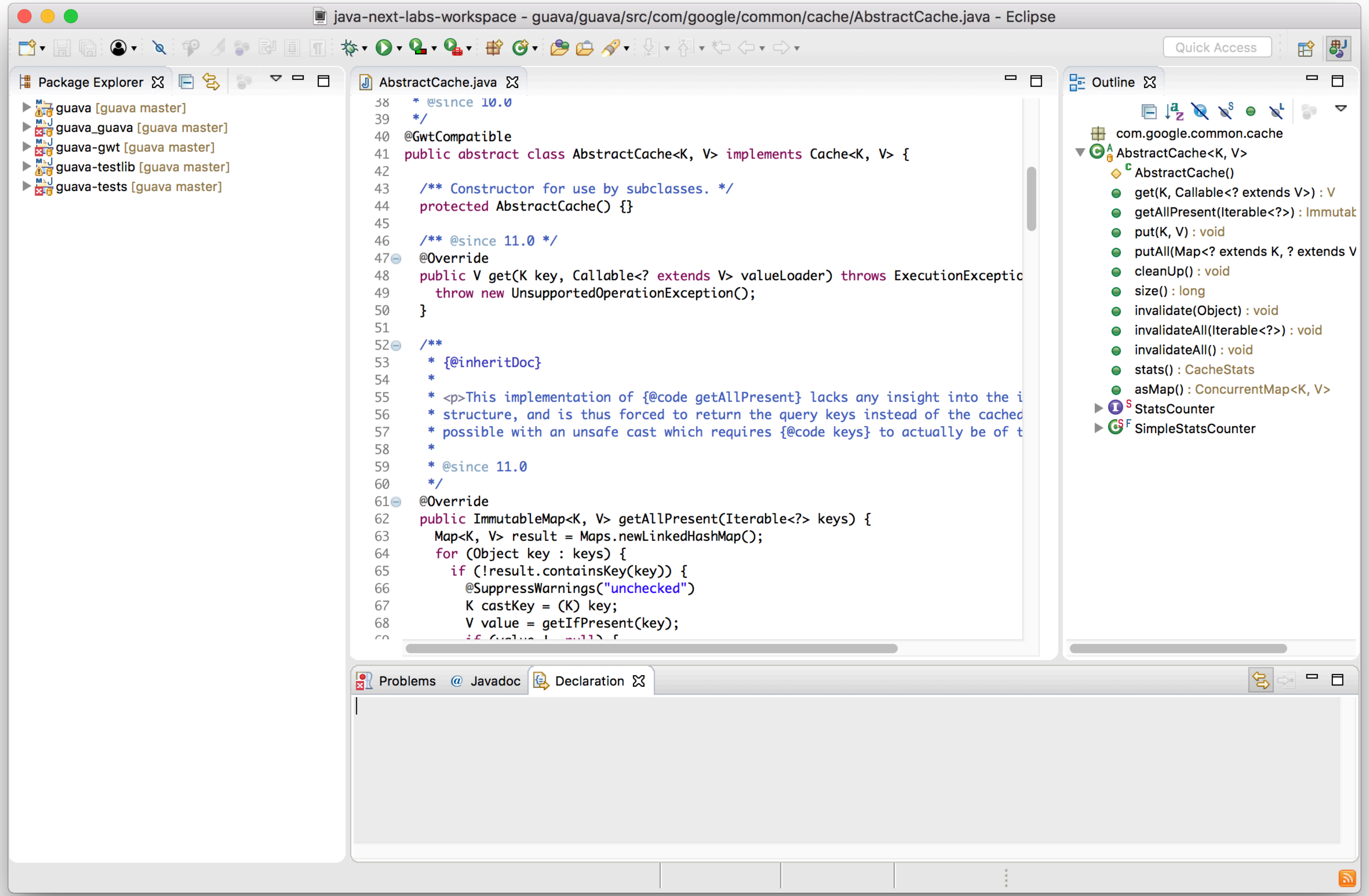
MyApplication.java

```java
package com.acme;

public class MyApplication {

    public static void main(String... args) {
        System.out.println("Hello!");
    }

}
```

```
com.acme
  MyApplication
    main(String...) : void
```

Debugging

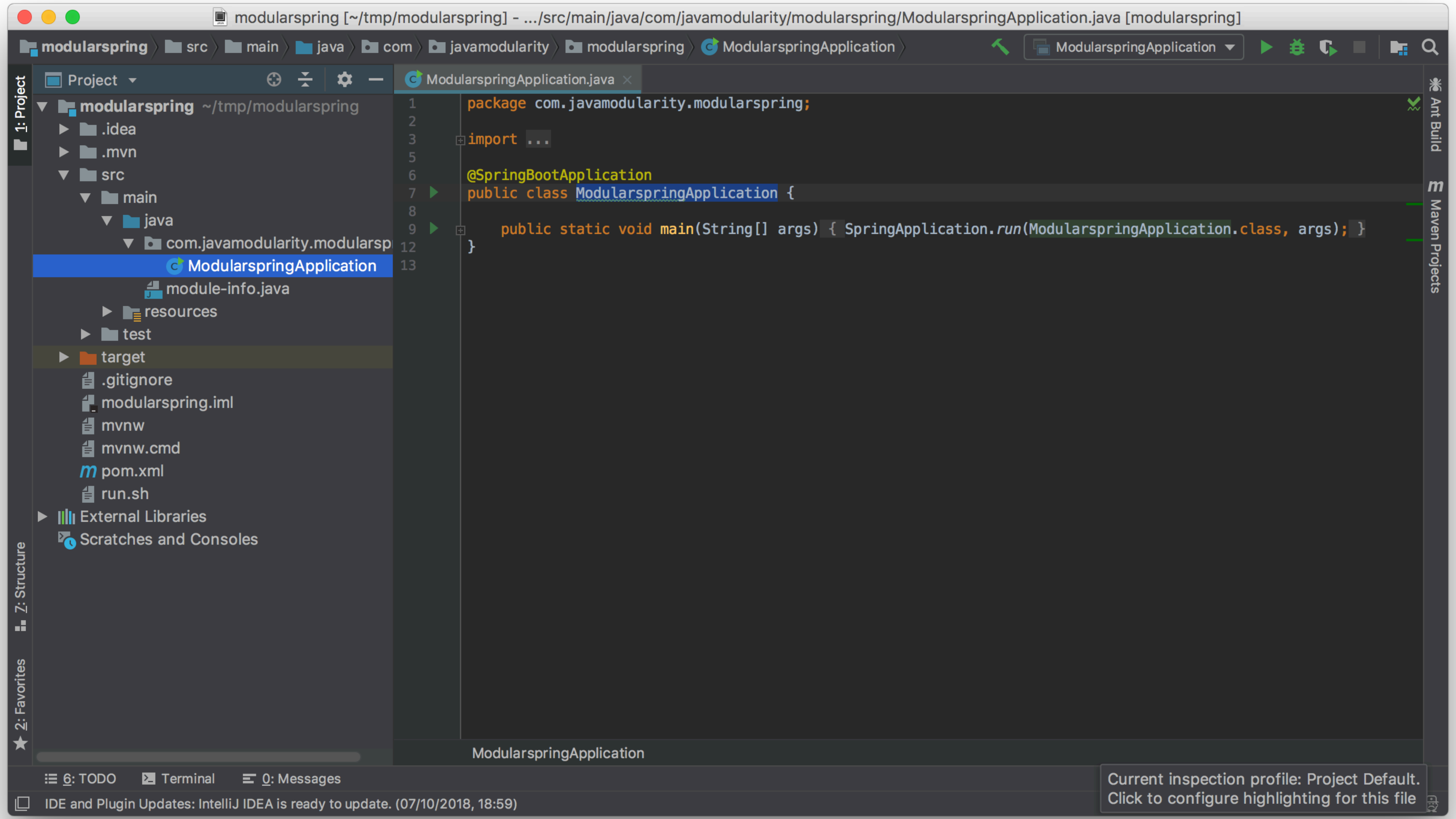
Eclipse



java-next-labs-workspace - guava/guava/src/com/google/common/cache/AbstractCache.java - Eclipse
Quick Access
Package Explorer
guava [guava master]
guava_guava [guava master]
guava-gwt [guava master]
guava-testlib [guava master]
guava-tests [guava master]
AbstractCache.java
Outline
com.google.common.cache
AbstractCache<K, V>
AbstractCache()
get(K, Callable<? extends V>) : V
getAllPresent(Iterable<?>) : Immutab
put(K, V) : void
putAll(Map<? extends K, ? extends V
cleanUp() : void
size() : long
invalidate(Object) : void
invalidateAll(Iterable<?>) : void
invalidateAll() : void
stats() : CacheStats
asMap() : ConcurrentMap<K, V>
StatsCounter
SimpleStatsCounter
Problems
Javadoc
Declaration

```java
 * @since 10.0
 */
@GwtCompatible
public abstract class AbstractCache<K, V> implements Cache<K, V> {

  /** Constructor for use by subclasses. */
  protected AbstractCache() {}

  /** @since 11.0 */
  @Override
  public V get(K key, Callable<? extends V> valueLoader) throws ExecutionExceptic
    throw new UnsupportedOperationException();
  }

  /**
   * {@inheritDoc}
   *
   * <p>This implementation of {@code getAllPresent} lacks any insight into the i
   * structure, and is thus forced to return the query keys instead of the cached
   * possible with an unsafe cast which requires {@code keys} to actually be of t
   *
   * @since 11.0
   */
  @Override
  public ImmutableMap<K, V> getAllPresent(Iterable<?> keys) {
    Map<K, V> result = Maps.newLinkedHashMap();
    for (Object key : keys) {
      if (!result.containsKey(key)) {
        @SuppressWarnings("unchecked")
        K castKey = (K) key;
        V value = getIfPresent(key);
```

IntelliJ


modularspring [~/tmp/modularspring] - .../src/main/java/com/javamodularity/modularspring/ModularspringApplication.java [modularspring]
modularspring › src › main › java › com › javamodularity › modularspring › ModularspringApplication
Project
modularspring ~/tmp/modularspring
.idea
.mvn
src
main
java
com.javamodularity.modularsp
ModularspringApplication
module-info.java
resources
test
target
.gitignore
modularspring.iml
mvnw
mvnw.cmd
pom.xml
run.sh
External Libraries
Scratches and Consoles
ModularspringApplication.java
6: TODO
Terminal
0: Messages
Ant Build
Maven Projects
1: Project
7: Structure
2: Favorites

```java
package com.javamodularity.modularspring;

import ...

@SpringBootApplication
public class ModularspringApplication {

    public static void main(String[] args) { SpringApplication.run(ModularspringApplication.class, args); }
}
```

Current inspection profile: Project Default.
Click to configure highlighting for this file

IDE and Plugin Updates: IntelliJ IDEA is ready to update. (07/10/2018, 18:59)

Unit Testing

Write code to test code

Individual classes and methods

Isolate dependencies: mocking

JUnit

Mockito

Demo

Unit Testing

Build Tools

Repeatable builds

Managing multiple modules

Managing external dependencies

Running tests

Maven

pom.xml

```
<project>
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>

  <dependencies>
     ...
  </dependencies>
</project>
```

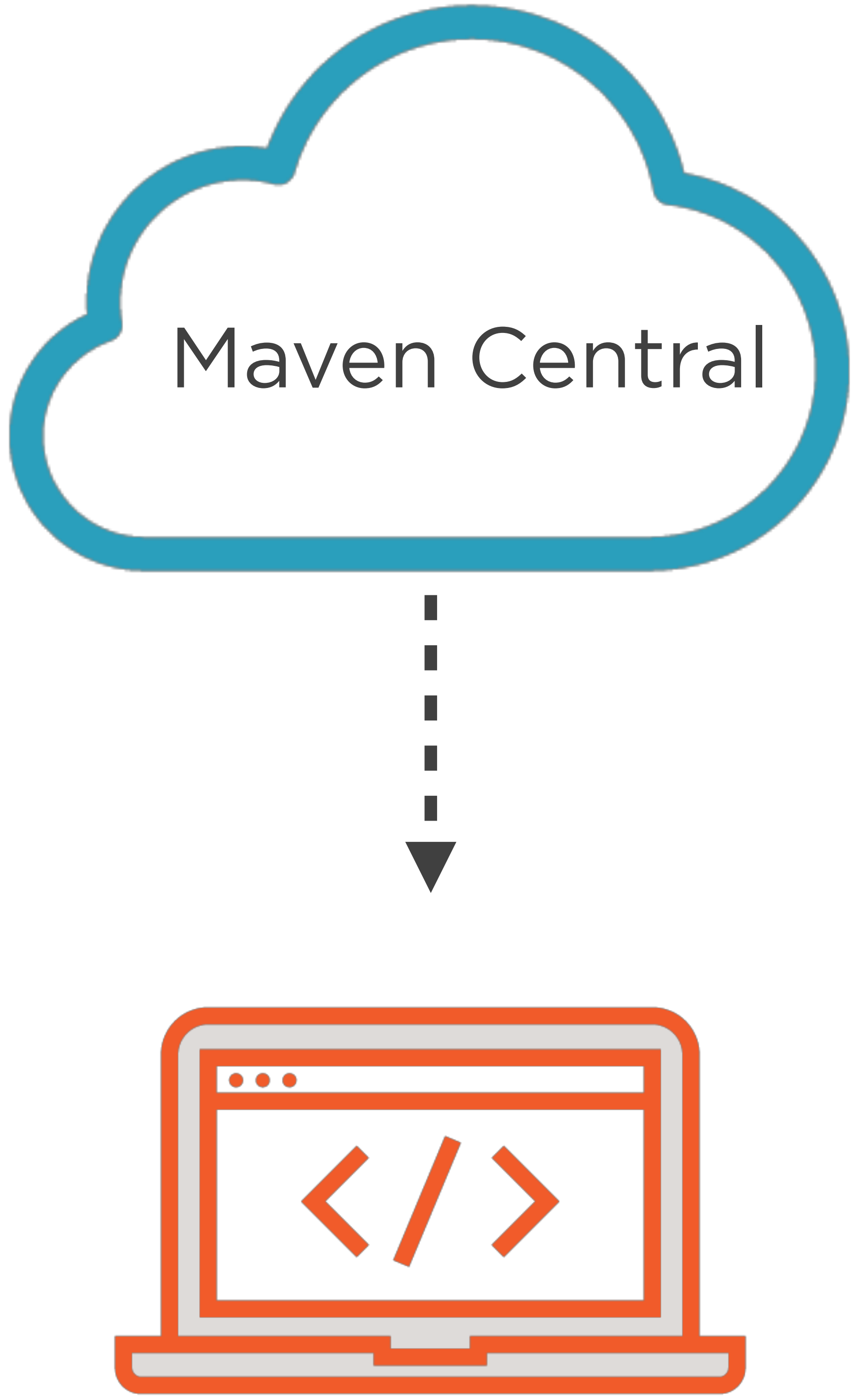
```
src/
  main/
    java/
      ...
  test/
    java/
      ...
  resources/
      ...
```

Maven

Download dependencies Compile Test Package

Maven

```
<dependency>
    <groupId>io.netty</groupId>
    <artifactId>netty-all</artifactId>
    <version>4.1.30</version>
</dependency>
```



Gradle

Define builds with Groovy scripts

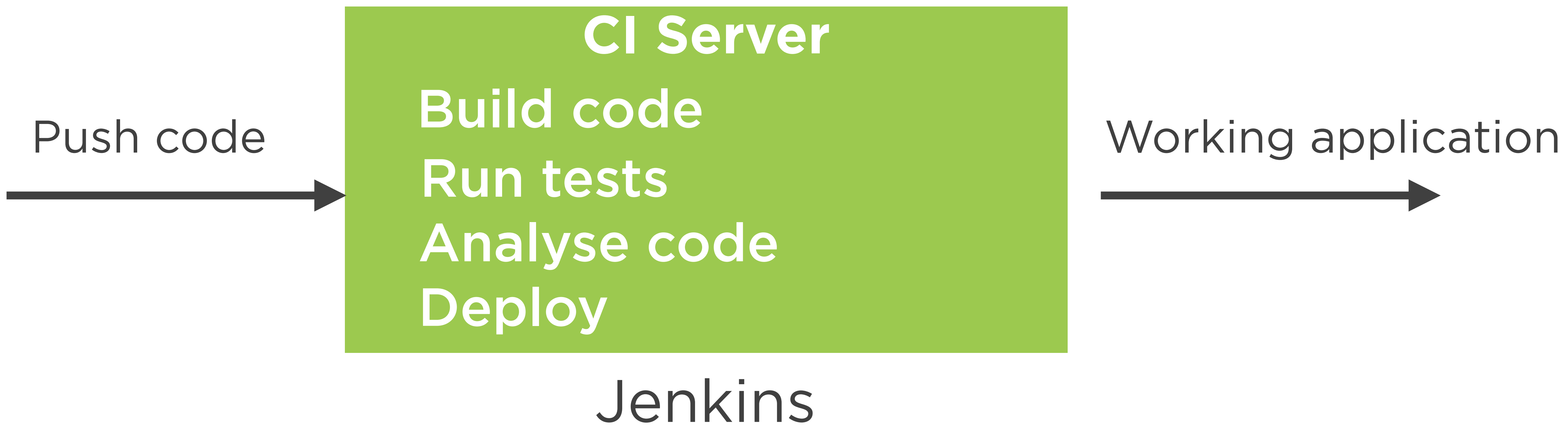
Incremental builds

Maven default source layout

Can use Maven Central

Continuous Integration & Quality Control

"Works on my machine!"

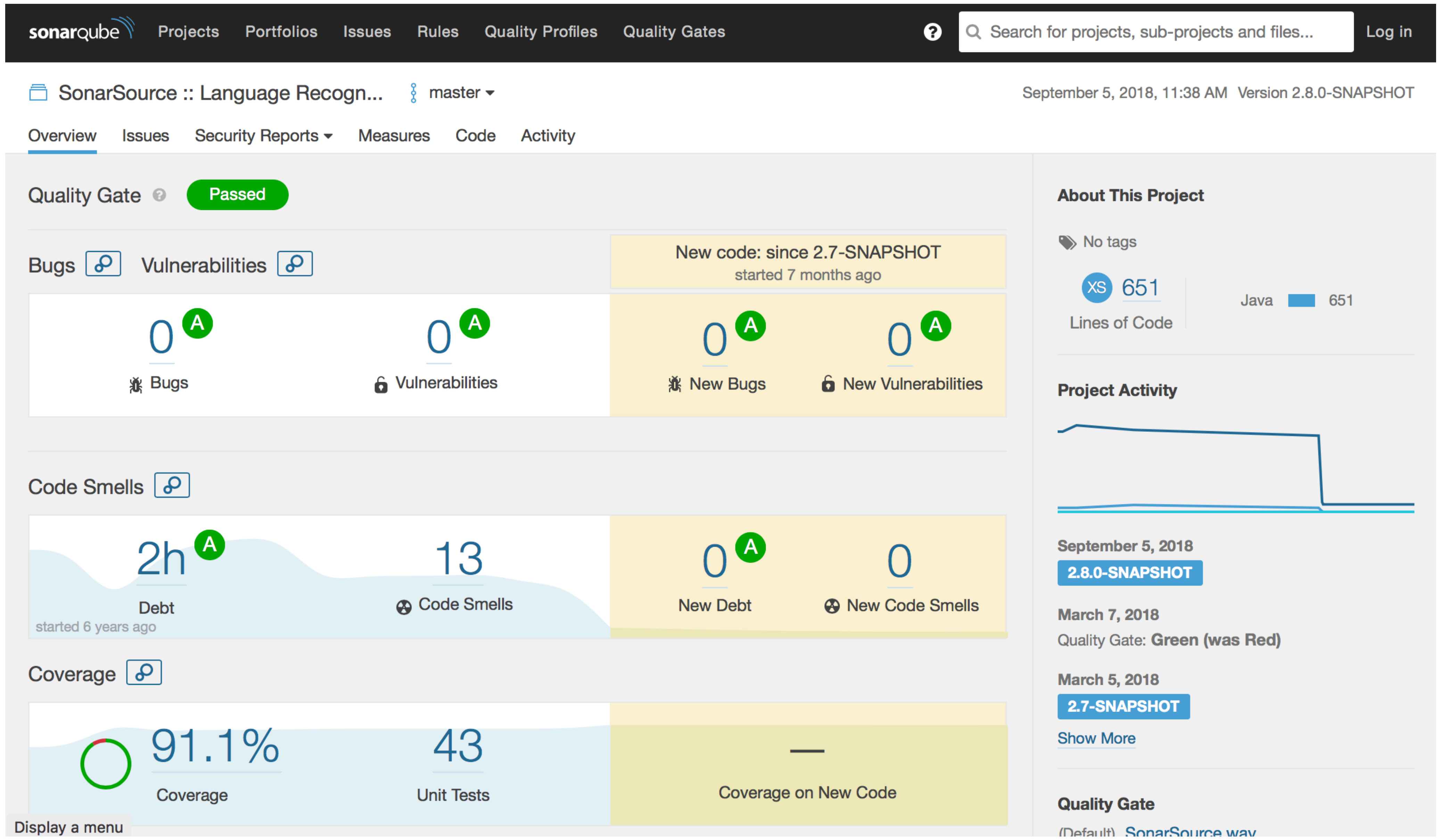


Static Code Analysis

Checkstyle

Spotbugs

PMD



SonarSource :: Language Recogn... master September 5, 2018, 11:38 AM Version 2.8.0-SNAPSHOT

Quality Gate Passed

Bugs Vulnerabilities

New code: since 2.7-SNAPSHOT
started 7 months ago

0 A Bugs

0 A Vulnerabilities

0 A New Bugs

0 A New Vulnerabilities

Code Smells

2h A Debt
started 6 years ago

13 Code Smells

0 A New Debt

0 New Code Smells

Coverage

91.1% Coverage

43 Unit Tests

— Coverage on New Code

About This Project

No tags

XS 651 Lines of Code

Java 651

Project Activity

September 5, 2018
2.8.0-SNAPSHOT

March 7, 2018
Quality Gate: **Green (was Red)**

March 5, 2018
2.7-SNAPSHOT

Show More

Quality Gate
(Default) SonarSource way

SonarQube: Continuous Inspection

Summary

Productivity: IDEs

Unit testing

Build tools

Continuous Integration