JetBRAINS
Develop with pleasure!

IntelliJ IDEA
Static Code Analysis

Hamlet D'Arcy
Canoo Engineering AG
@HamletDRC
http://hamletdarcy.blogspot.com

Static Code Analysis

Code Inspections

JSR 305 and 308 Annotations

Duplicate Detection

Stack Trace Analysis

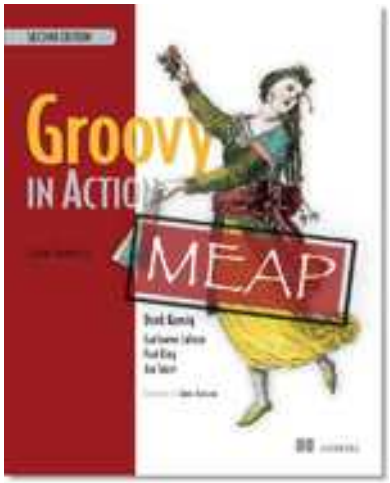
Dataflow Analysis

Dependency Analysis

JetBRAINS

About Me

JETBRAINS
DEVELOPMENT ACADEMY
MEMBER



Groovy

CODENARC
Less Bugs Better Code

canoo
› your provider for business web solutions ›

Static Code Analysis

Code Inspections

JSR 305 and 308 Annotations

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis

JetBRAINS

```
class _01Example {

  private static long count = 0L;

  public synchronized void increment() {
    count++;
  }
}
```

JetBRAINS

```
class _02Example {

  private boolean active = false;

  public boolean isActive() {
    return active;
  }

  public synchronized void activate() {
    active = true;
  }
}
```

JetBRAINS

```
class _03Example {
  private final ReentrantLock lock = new ReentrantLock();
  private boolean active = false;

  public boolean isActive() throws Exception {
    lock.lock();
    boolean result = active;
    lock.unlock();
    return result;
  }

  public void activate() {
    lock.lock();
    active = true;
    lock.unlock();
  }
}
```

JetBRAINS

```
class _04Example {
  private static final boolean DEFAULT = true;

  void myMethod(Boolean value) {
    if (value == null)
      System.out.println("value: null");
      value = DEFAULT;

    System.out.println("received: " + value);
  }
}
```

JetBRAINS

```
class _05Example {

  Frame makeFrame(int height, int width) {
    Frame frame = new Frame();
    frame.setSize(height, width);
    return frame;
  }

  Rectangle makeRectangle() {
    int x = 0;
    int y = 0;
    return new Rectangle(y, x, 20, 20);
  }
}
```

```
class _06Example {
    {
        try {
            doSomething();
        } catch (UnsupportedOperationException e) {
            handleError(e);
        } catch (IllegalStateException e) {
            handleError(e);
        } catch (IllegalArgumentException e) {
            handleError(e);
        }
    }
    ...
}
```

```
class _07Example {
    private def Object lock = new Object()

    def method() {
        synchronized(lock) {
            // do something
        }
    }
}
```

JetBRAINS

```
class _08Example {
    var property: String = null

    def getProperty() {
        println(property)
    }
}
```

Correctness

Multi-threaded correctness

Malicious code vulnerability

Bad practice

Internationalization

Performance

Code style violations

Dodgy

* Bill Pugh, FindBugs

… and more

- Suppress False Positives
- Define profiles and scopes
- Run on demand
- Run from command line
- Team City integration
- FindBugs, PMD & CheckStyle plugins
- Language and framework support...

Supported Frameworks

- Android
- Ant
- Application Server Inspections
- CDI(Contexts and Dependency Injection)
- CSS
- Faces Model
- FreeMarker
- Google App Engine,
- Google Web Toolkit
- Groovy
- Guice
- Hibernate
- HTML
- J2ME
- Java EE
- JavaScript
- JSF
- JSP
- Junit
- LESS
- Maven
- OSGi
- RELAX NG
- SCSS
- Spring Model
- Spring Web Services
- SQL
- TestNG
- Velocity
- Java WebServices
- Webflow Model
- WSDL
- XML
- Xpath
- XSLT
- **... and many more**

Write Your Own

IntelliJ IDEA Static Analysis:
Custom Rules with Structural Search & Replace

On http://JetBrains.tv

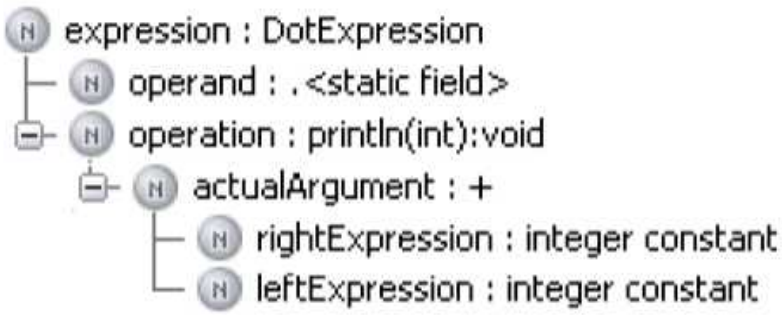
10 Best Unknown Inspections

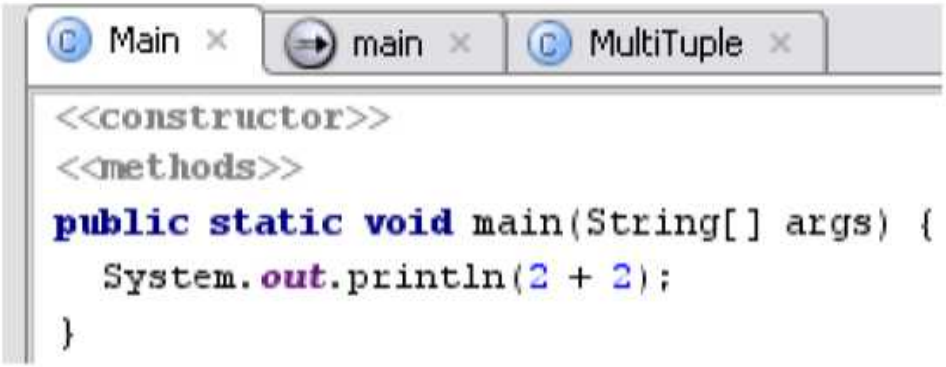
- Illegal package dependencies
- 'this' reference escapes constructor
- Field accessed in both synched & unsynched contexts
- non private field accessed in synched context
- Synchronization on 'this' and 'synchronized' method
- return of collection or array field
- call to 'Thread.run()'
- expression.equals("literal") rather than "literal".equals(expression)
- equals method does not check class of parameter
- method may be static

http://hamletdarcy.blogspot.com/2008/04/10-best-idea-inspections-youre-not.html

How it Works

- Searches AST for Bug Patterns





How it Works

```
@Override
public void visitMethod(@NotNull final PsiMethod method) {
  super.visitMethod(method);
  if (method.hasModifierProperty(PsiModifier.ABSTRACT)) {
    return;
  }
  if (!RecursionUtils.methodMayRecurse(method)) {
    return;
  }
  if (!RecursionUtils.methodDefinitelyRecurses(method)) {
    return;
  }
  super.registerMethodError(method);
}
```

JetBRAINS

Static Code Analysis

Code Inspections

JSR 305 and 308 Annotations

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis

JetBRAINS

@Immutable and @GuardedBy

```
@Immutable
public class GuardedByExample {

  private final Object lock = new Object();

  @GuardedBy("lock")
  private final List<Object> myList = new ArrayList<Object>();

  public Object getElement(int index) {
    synchronized (lock) {
      return myList.get(index);
    }
  }

  public void addElement(Object e) {
    synchronized (lock) {
      myList.add(e);
    }
  }
}
```

JetBRAINS

@Nullable and @NotNull

```
public class NullableExample {
  @Nullable Integer getId() {
    return 1;
  }

  @NotNull String getName() {
    return "name";
  }

  @Override public String toString() {
    if (getName() == null) {
      return getId().toString() + "<unknown>";
    } else {
      return getId().toString() + getName();
    }
  }
}
```

JetBRAINS

@Pattern

```
class PatternExample {

  @Pattern("[a-zA-Z]+")
  String getName() {
    return "my name";
  }
}
```

@Language

```
public class LanguageExample {

  @Language("Groovy")
  String getScript() {
    return "5.times { i -> println \"Hello $i\" } ";
  }

  String getMarkup() {
    @Language("XML")
    String markup = "<root><body>Some Text</body></root>";
    return markup;
  }
}
```

@Nls, @NonNls, @PropertyKey

- Resource bundle & i18n integration
- Extracting hard-coded String literals: http://goo.gl/VZDln
- Documentation: http://goo.gl/NWzsv

Static Code Analysis

Code Inspections

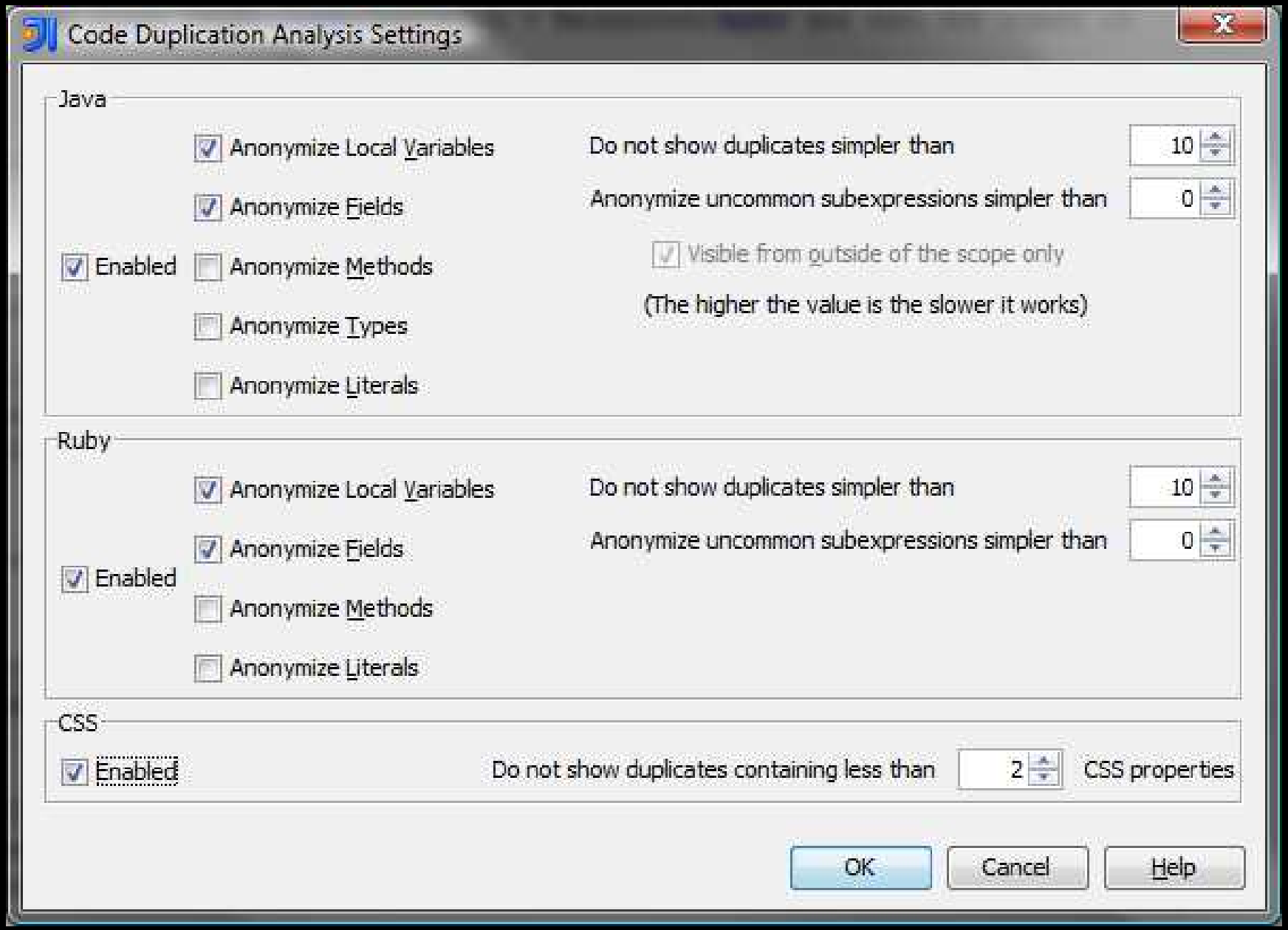
JSR 305 and 308 Annotations

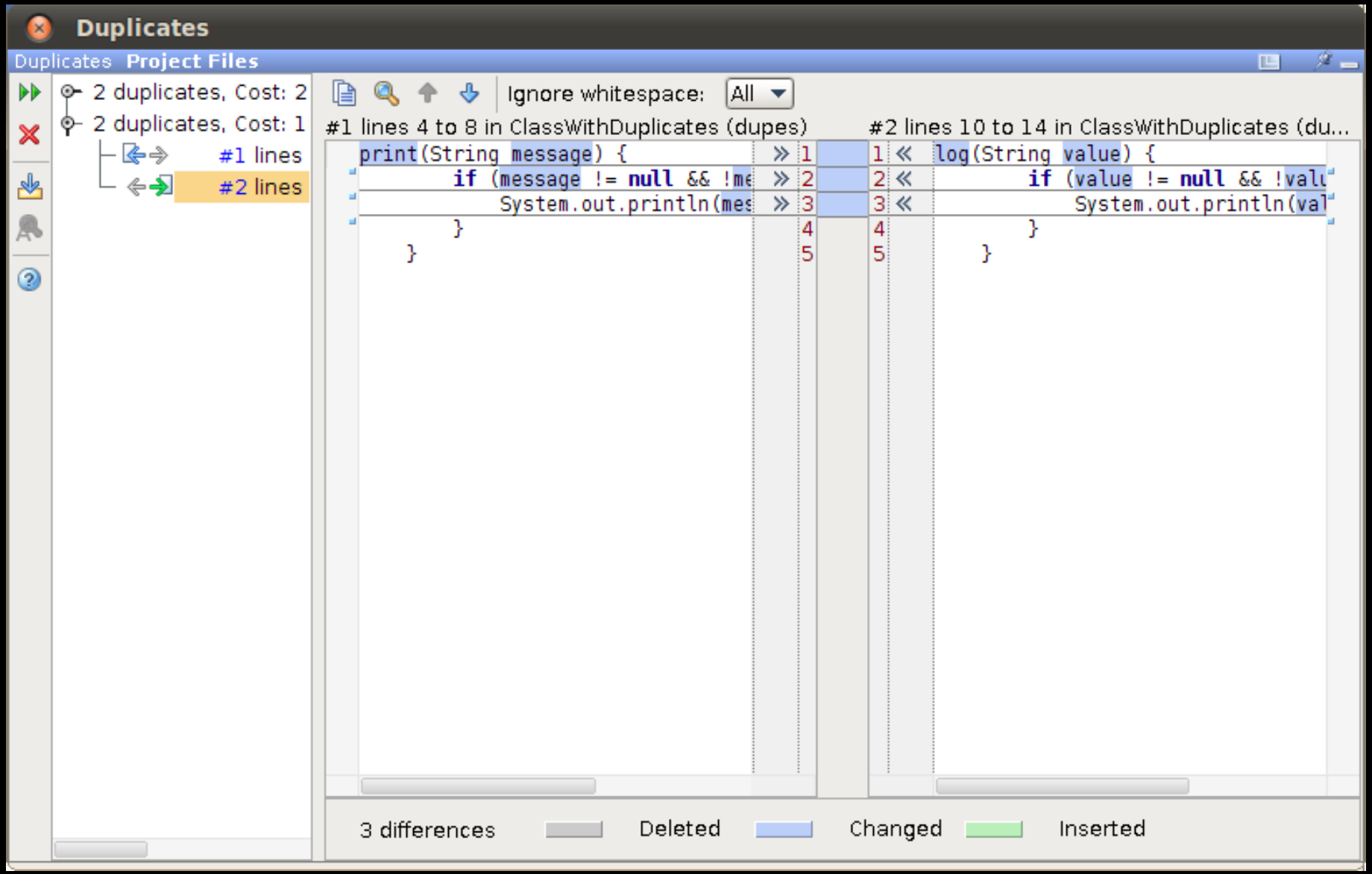
Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis


Code Duplication Analysis Settings
Java
Enabled
Anonymize Local Variables
Anonymize Fields
Anonymize Methods
Anonymize Types
Anonymize Literals
Do not show duplicates simpler than
10
Anonymize uncommon subexpressions simpler than
0
Visible from outside of the scope only
(The higher the value is the slower it works)
Ruby
Enabled
Anonymize Local Variables
Anonymize Fields
Anonymize Methods
Anonymize Literals
Do not show duplicates simpler than
10
Anonymize uncommon subexpressions simpler than
0
CSS
Enabled
Do not show duplicates containing less than
2
CSS properties
OK
Cancel
Help


Duplicates
Duplicates Project Files
2 duplicates, Cost: 2
2 duplicates, Cost: 1
#1 lines
#2 lines
Ignore whitespace:
All
#1 lines 4 to 8 in ClassWithDuplicates (dupes)
#2 lines 10 to 14 in ClassWithDuplicates (du...
print(String message) {
if (message != null && !me
System.out.println(mes
}
}
log(String value) {
if (value != null && !valu
System.out.println(val
}
}
3 differences
Deleted
Changed
Inserted

Duplicate Detection

- Anonymizes Local Variables, Fields, Methods, Types, and Literals
- Provides weighted/scored analysis
- Supports several languages

- More info: http://goo.gl/qmhhd

Static Code Analysis

Code Inspections

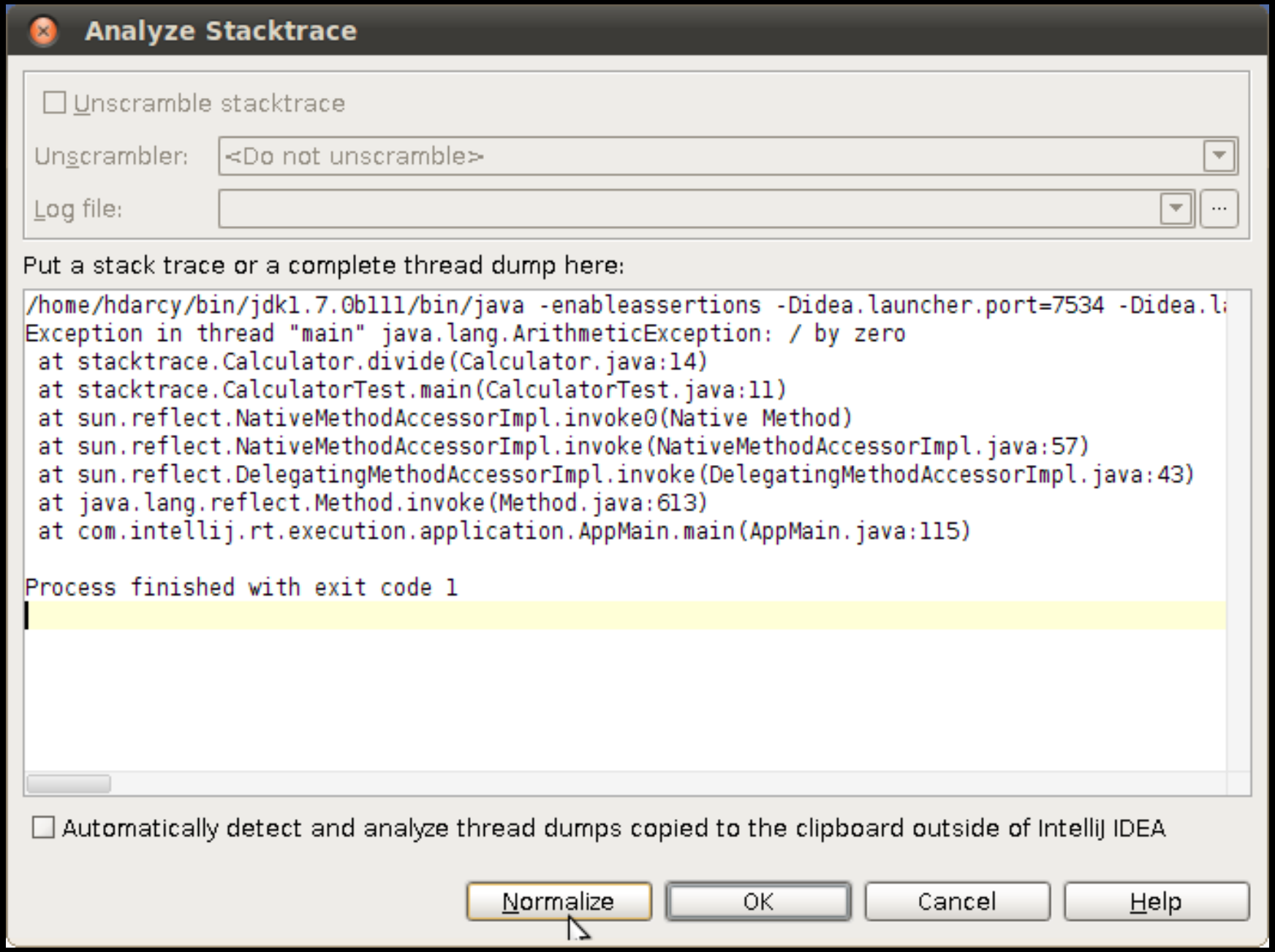
JSR 305 and 308 Annotations

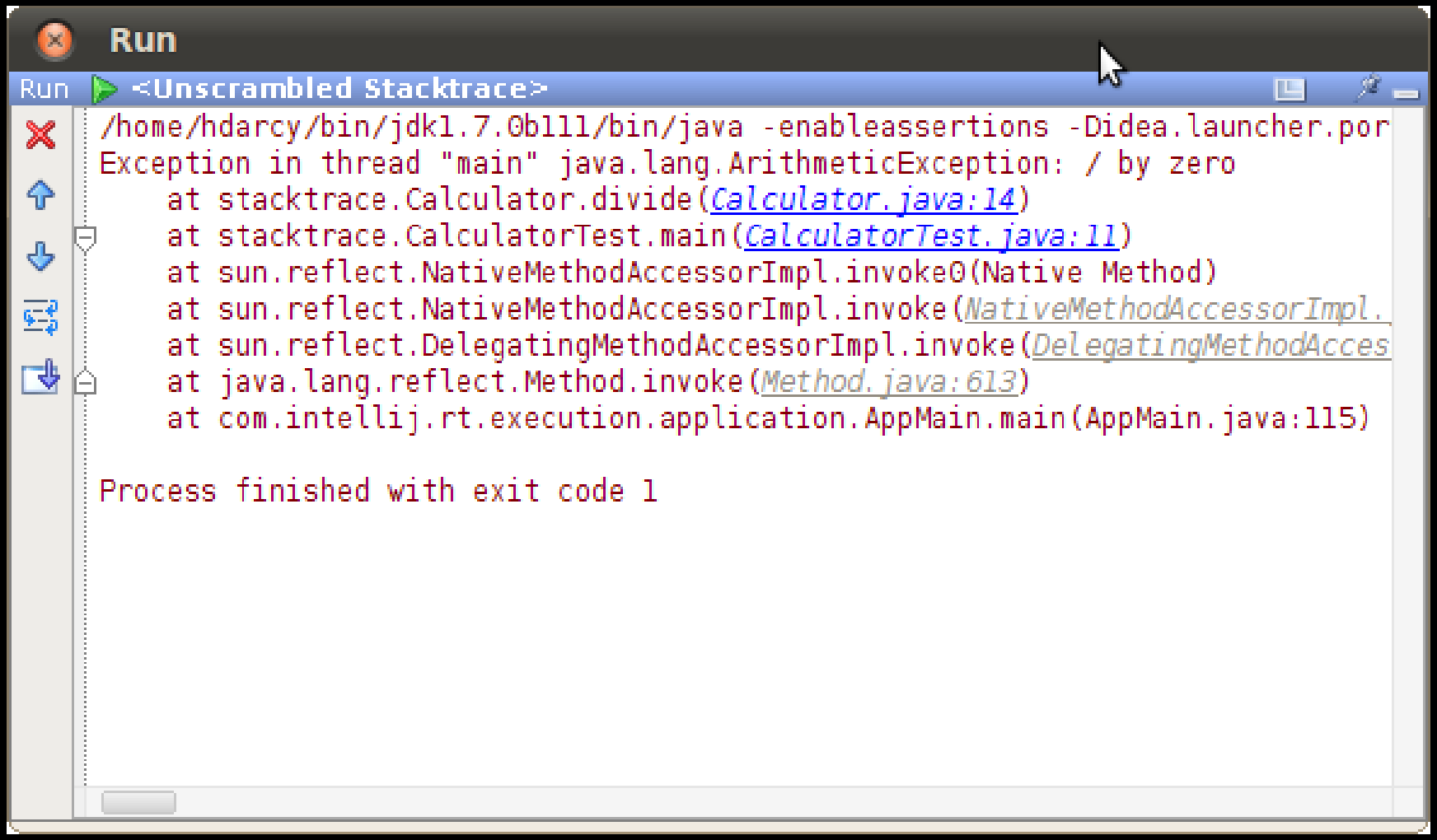
Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis


Analyze Stacktrace
Unscramble stacktrace
Unscrambler:
<Do not unscramble>
Log file:
Put a stack trace or a complete thread dump here:
/home/hdarcy/bin/jdk1.7.0b111/bin/java -enableassertions -Didea.launcher.port=7534 -Didea.l
Exception in thread "main" java.lang.ArithmeticException: / by zero
at stacktrace.Calculator.divide(Calculator.java:14)
at stacktrace.CalculatorTest.main(CalculatorTest.java:11)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:613)
at com.intellij.rt.execution.application.AppMain.main(AppMain.java:115)
Process finished with exit code 1
Automatically detect and analyze thread dumps copied to the clipboard outside of IntelliJ IDEA
Normalize
OK
Cancel
Help


Run
Run
<Unscrambled Stacktrace>
/home/hdarcy/bin/jdk1.7.0b111/bin/java -enableassertions -Didea.launcher.por
Exception in thread "main" java.lang.ArithmeticException: / by zero
at stacktrace.Calculator.divide(Calculator.java:14)
at stacktrace.CalculatorTest.main(CalculatorTest.java:11)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
at java.lang.reflect.Method.invoke(Method.java:613)
at com.intellij.rt.execution.application.AppMain.main(AppMain.java:115)
Process finished with exit code 1

Analyze Stacktrace

- Copy and paste log files into IDEA
- ZKM Unscramble support (& others)

- More Info: http://goo.gl/A8i87

Static Code Analysis

Code Inspections

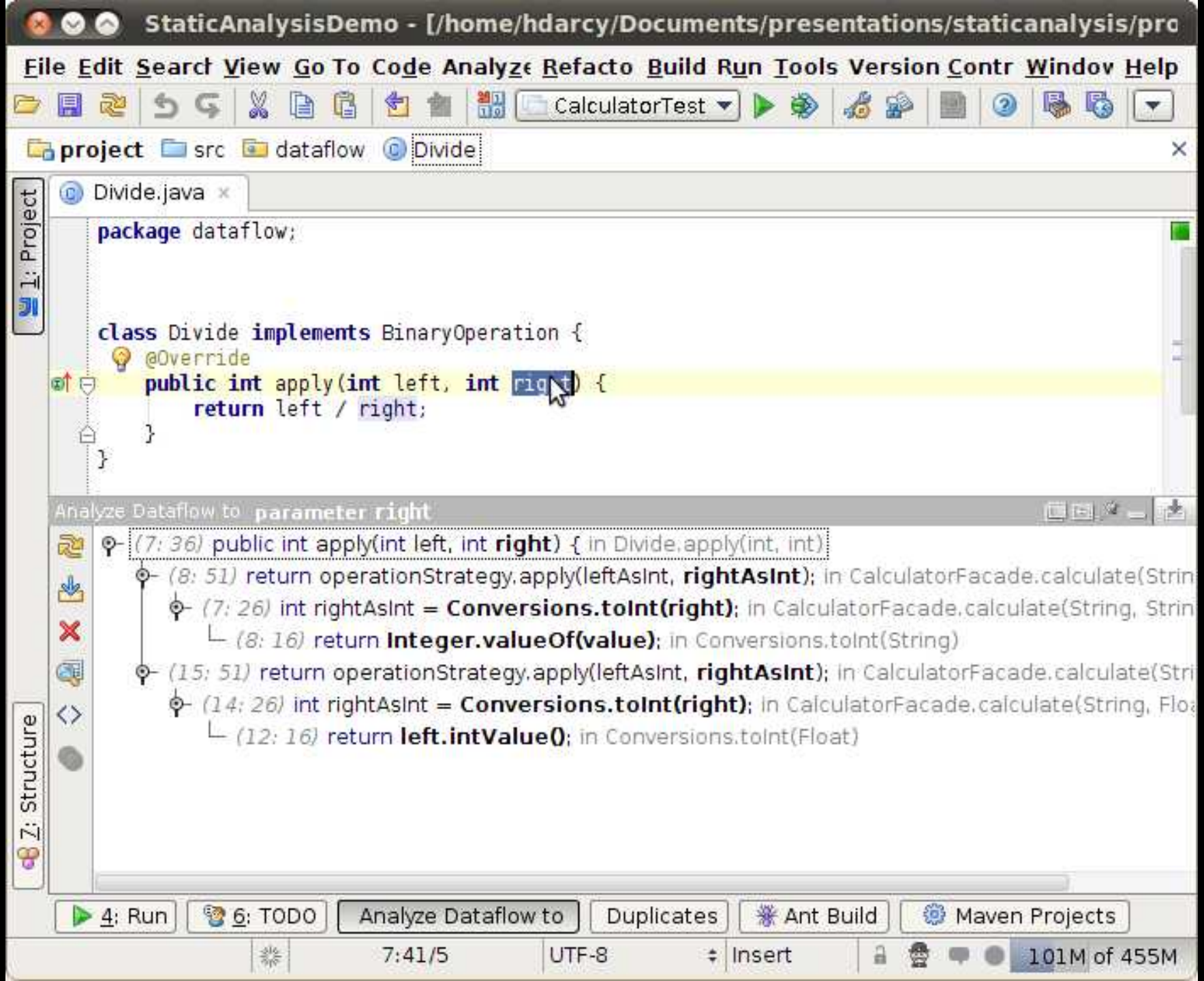
JSR 305 and 308 Annotations

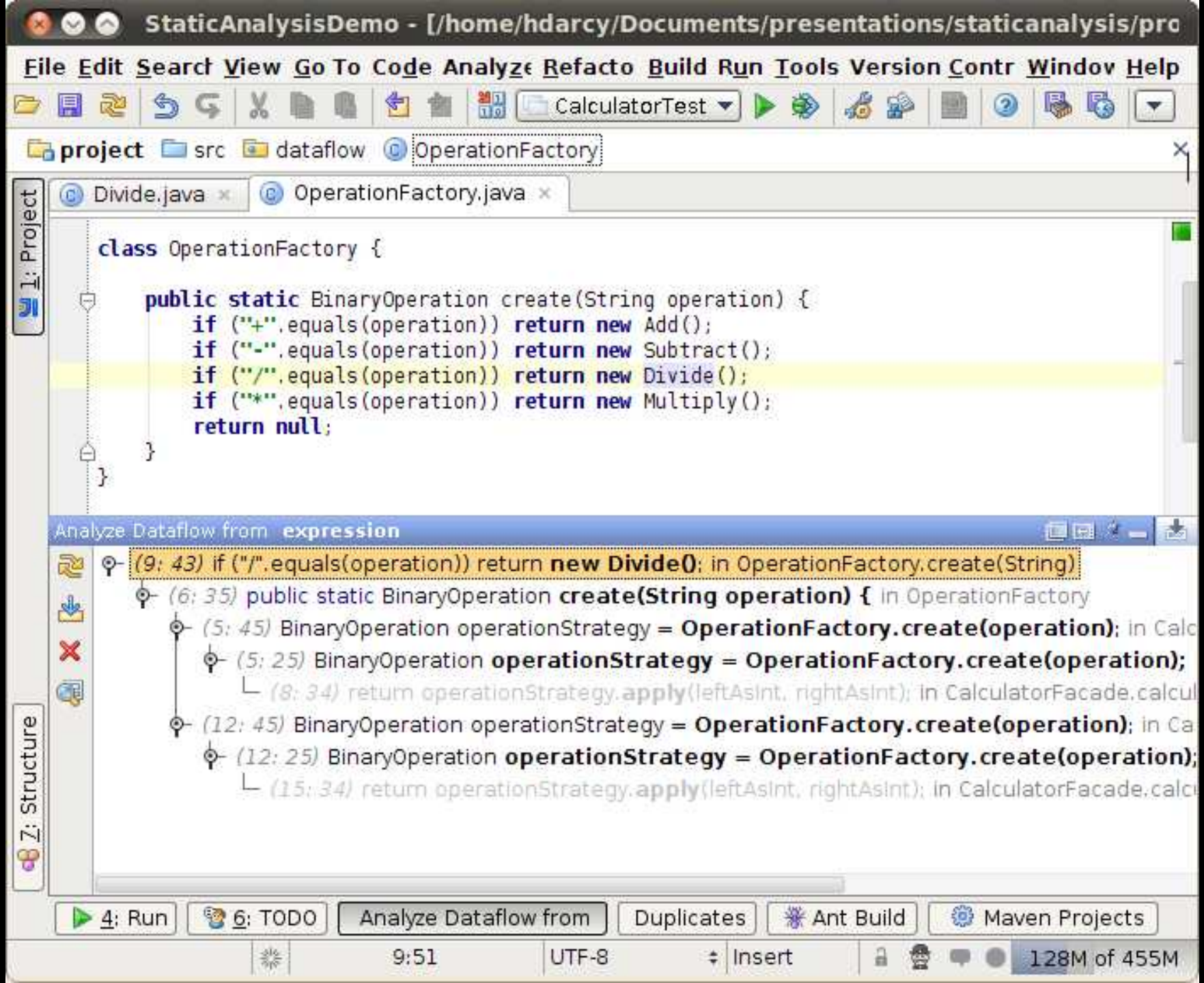
Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis


StaticAnalysisDemo - [/home/hdarcy/Documents/presentations/staticanalysis/pro
File Edit Search View Go To Code Analyze Refacto Build Run Tools Version Contr Window Help
CalculatorTest
project src dataflow Divide
Divide.java
1: Project
package dataflow;
class Divide implements BinaryOperation {
@Override
public int apply(int left, int right) {
return left / right;
}
}
Analyze Dataflow to parameter right
(7: 36) public int apply(int left, int right) { in Divide.apply(int, int)
(8: 51) return operationStrategy.apply(leftAsInt, rightAsInt); in CalculatorFacade.calculate(Strin
(7: 26) int rightAsInt = Conversions.toInt(right); in CalculatorFacade.calculate(String, Strin
(8: 16) return Integer.valueOf(value); in Conversions.toInt(String)
(15: 51) return operationStrategy.apply(leftAsInt, rightAsInt); in CalculatorFacade.calculate(Stri
(14: 26) int rightAsInt = Conversions.toInt(right); in CalculatorFacade.calculate(String, Flo
(12: 16) return left.intValue(); in Conversions.toInt(Float)
7: Structure
4: Run
6: TODO
Analyze Dataflow to
Duplicates
Ant Build
Maven Projects
7:41/5
UTF-8
Insert
101M of 455M


StaticAnalysisDemo - [/home/hdarcy/Documents/presentations/staticanalysis/pro
File Edit Search View Go To Code Analyze Refacto Build Run Tools Version Contr Windov Help
CalculatorTest
project
src
dataflow
OperationFactory
Divide.java
OperationFactory.java
class OperationFactory {
public static BinaryOperation create(String operation) {
if ("+".equals(operation)) return new Add();
if ("-".equals(operation)) return new Subtract();
if ("/".equals(operation)) return new Divide();
if ("*".equals(operation)) return new Multiply();
return null;
}
}
Analyze Dataflow from expression
(9: 43) if ("/".equals(operation)) return new Divide(); in OperationFactory.create(String)
(6: 35) public static BinaryOperation create(String operation) { in OperationFactory
(5: 45) BinaryOperation operationStrategy = OperationFactory.create(operation); in Calc
(5: 25) BinaryOperation operationStrategy = OperationFactory.create(operation);
(8: 34) return operationStrategy.apply(leftAsInt, rightAsInt); in CalculatorFacade.calcul
(12: 45) BinaryOperation operationStrategy = OperationFactory.create(operation); in Ca
(12: 25) BinaryOperation operationStrategy = OperationFactory.create(operation);
(15: 34) return operationStrategy.apply(leftAsInt, rightAsInt); in CalculatorFacade.calc
1: Project
7: Structure
4: Run
6: TODO
Analyze Dataflow from
Duplicates
Ant Build
Maven Projects
9:51
UTF-8
Insert
128M of 455M

Dataflow Analysis

- *Code archeology*

- *to here* – how a reference gets set
- *from here* – where a reference goes to

- More info: http://goo.gl/Cp92Q

Static Code Analysis

Code Inspections

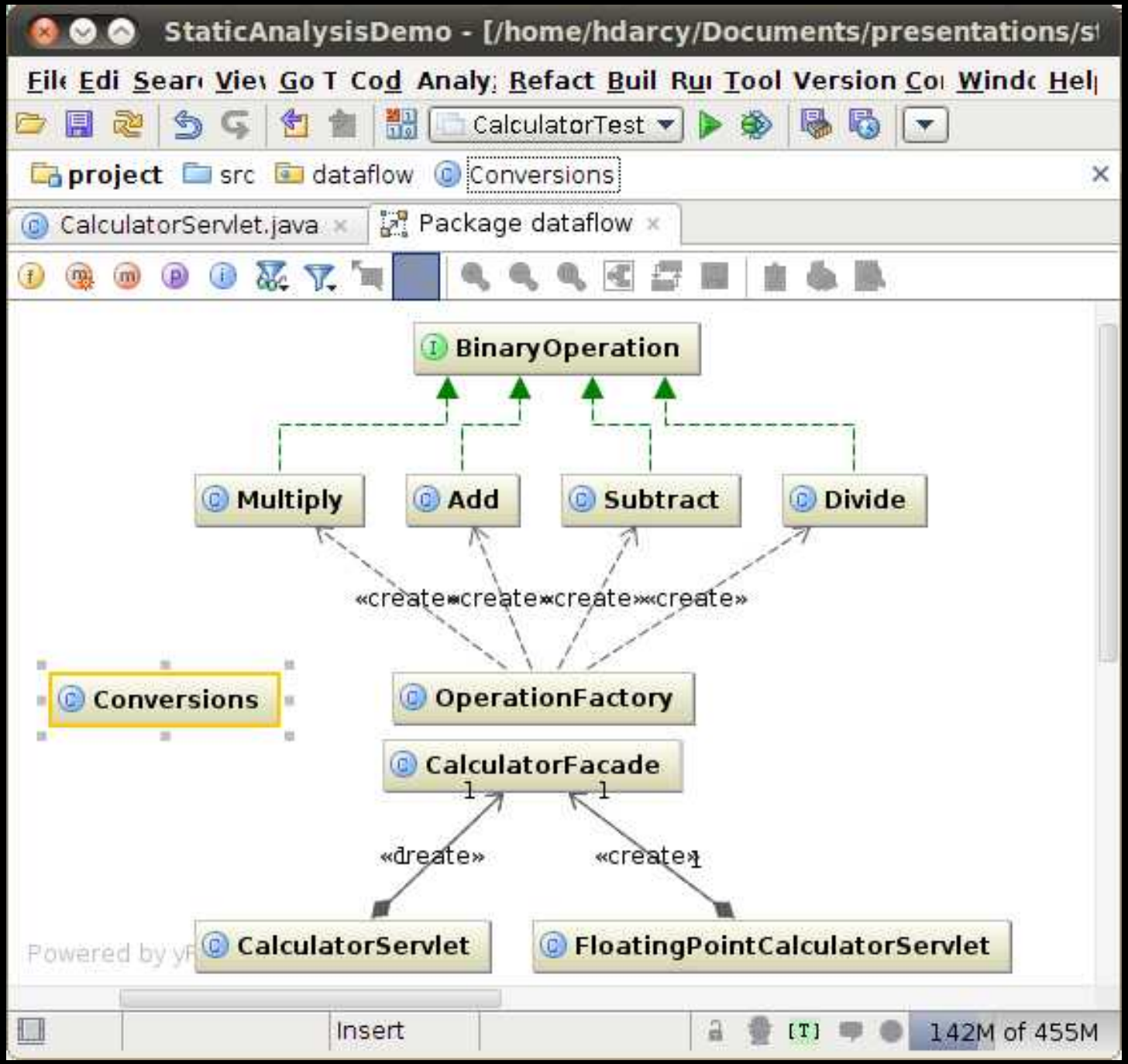
JSR 305 and 308 Annotations

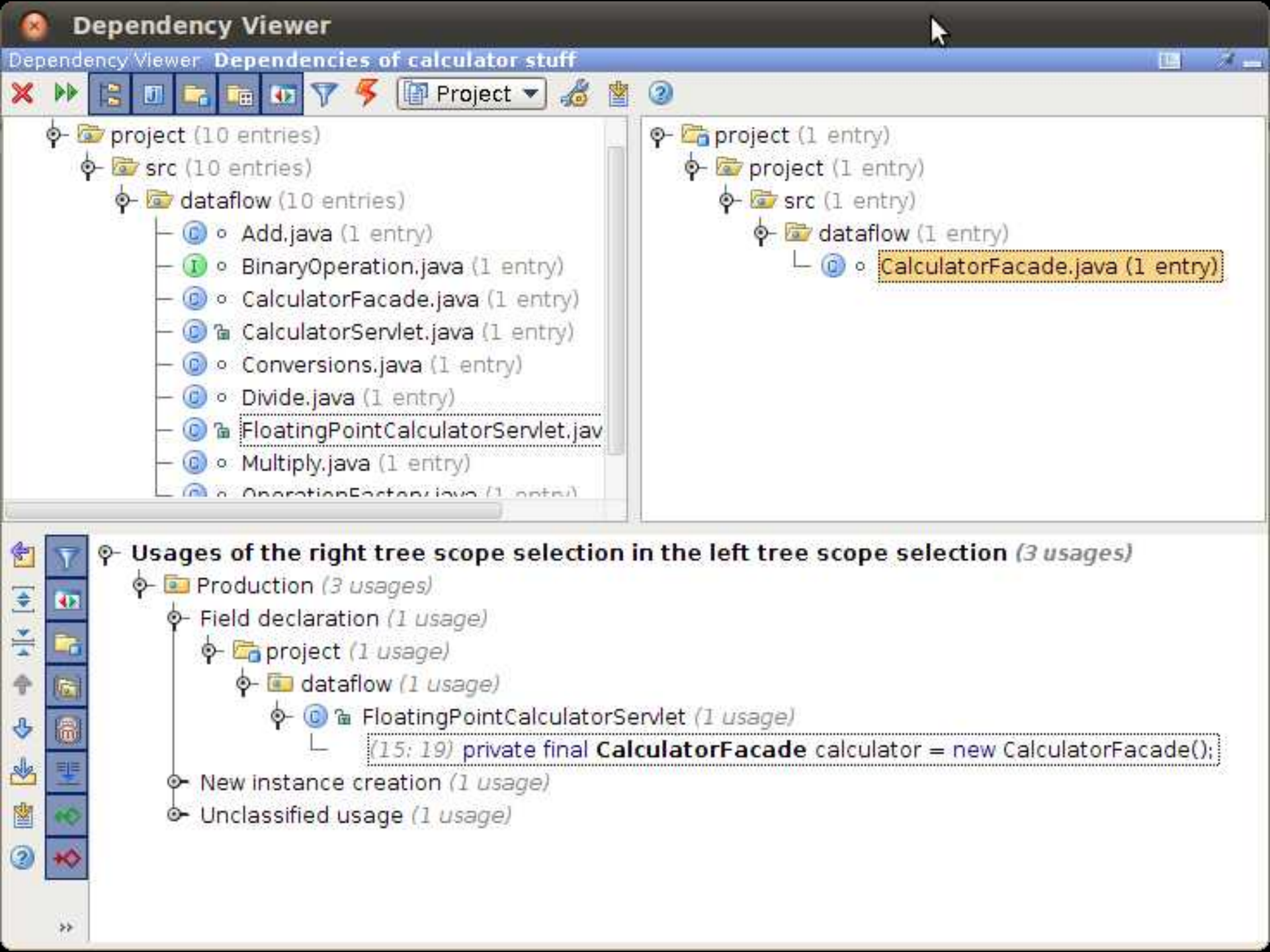
Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis


StaticAnalysisDemo - [/home/hdarcy/Documents/presentations/s
File Edi Sear Vie Go T Cod Analy Refact Buil Ru Tool Version Co Windc Hel
CalculatorTest
project
src
dataflow
Conversions
CalculatorServlet.java
Package dataflow
BinaryOperation
Multiply
Add
Subtract
Divide
«create»
«create»
«create»
«create»
Conversions
OperationFactory
CalculatorFacade
1
1
«create»
«create»
1
CalculatorServlet
FloatingPointCalculatorServlet
Powered by yF
Insert
142M of 455M


Dependency Viewer
Dependency Viewer Dependencies of calculator stuff
Project
project (10 entries)
src (10 entries)
dataflow (10 entries)
Add.java (1 entry)
BinaryOperation.java (1 entry)
CalculatorFacade.java (1 entry)
CalculatorServlet.java (1 entry)
Conversions.java (1 entry)
Divide.java (1 entry)
FloatingPointCalculatorServlet.jav
Multiply.java (1 entry)
project (1 entry)
project (1 entry)
src (1 entry)
dataflow (1 entry)
CalculatorFacade.java (1 entry)
Usages of the right tree scope selection in the left tree scope selection (3 usages)
Production (3 usages)
Field declaration (1 usage)
project (1 usage)
dataflow (1 usage)
FloatingPointCalculatorServlet (1 usage)
(15; 19) private final CalculatorFacade calculator = new CalculatorFacade();
New instance creation (1 usage)
Unclassified usage (1 usage)

JetBRAINS

UML Generation

- Dynamically generates diagram
- Standard Show/Hide options
- Integrated with Refactorings

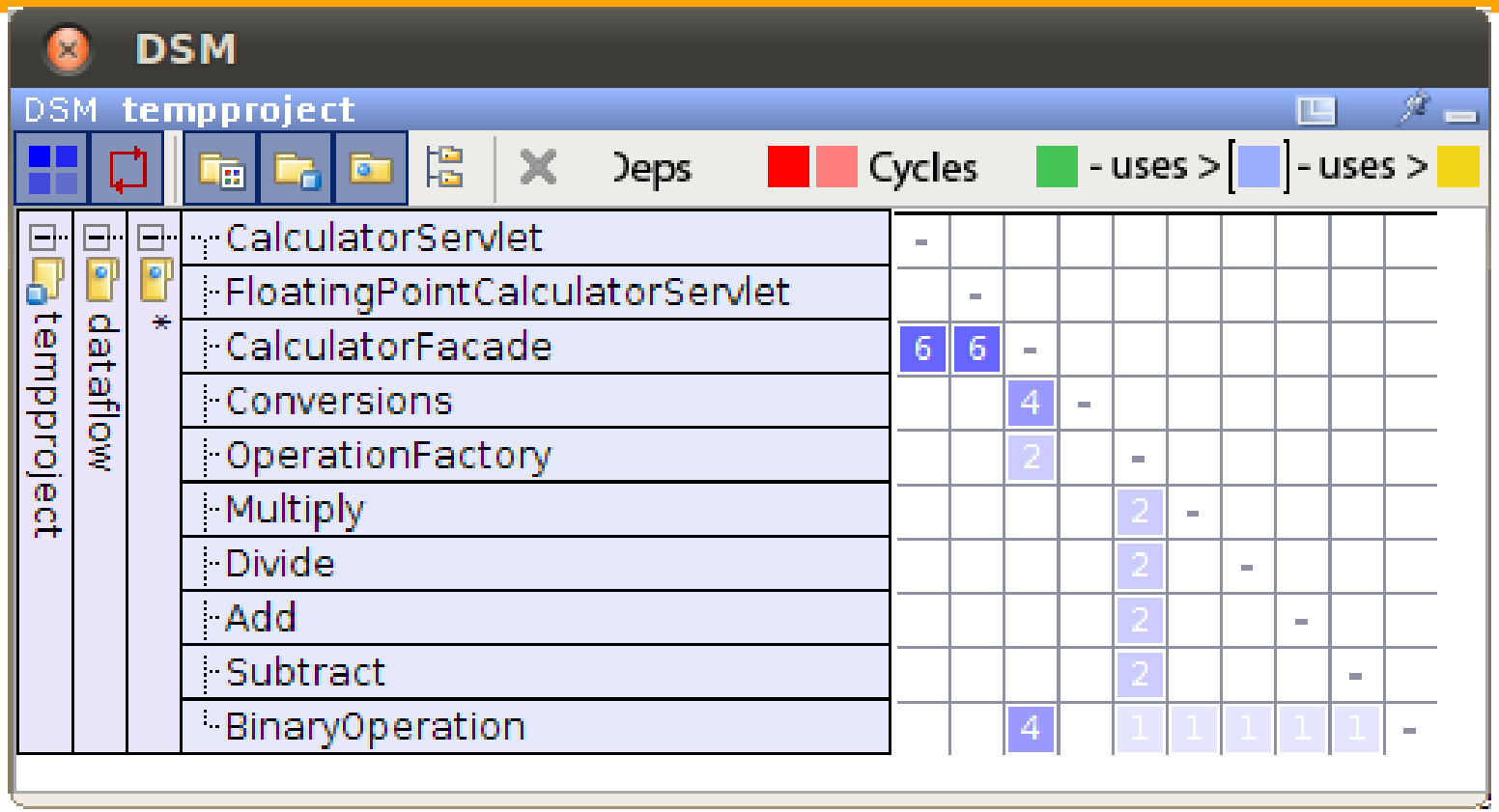
Dependency Analysis

- Shows all classes your code depends on
- Shows specific usages in your classes
- Allows jump to source

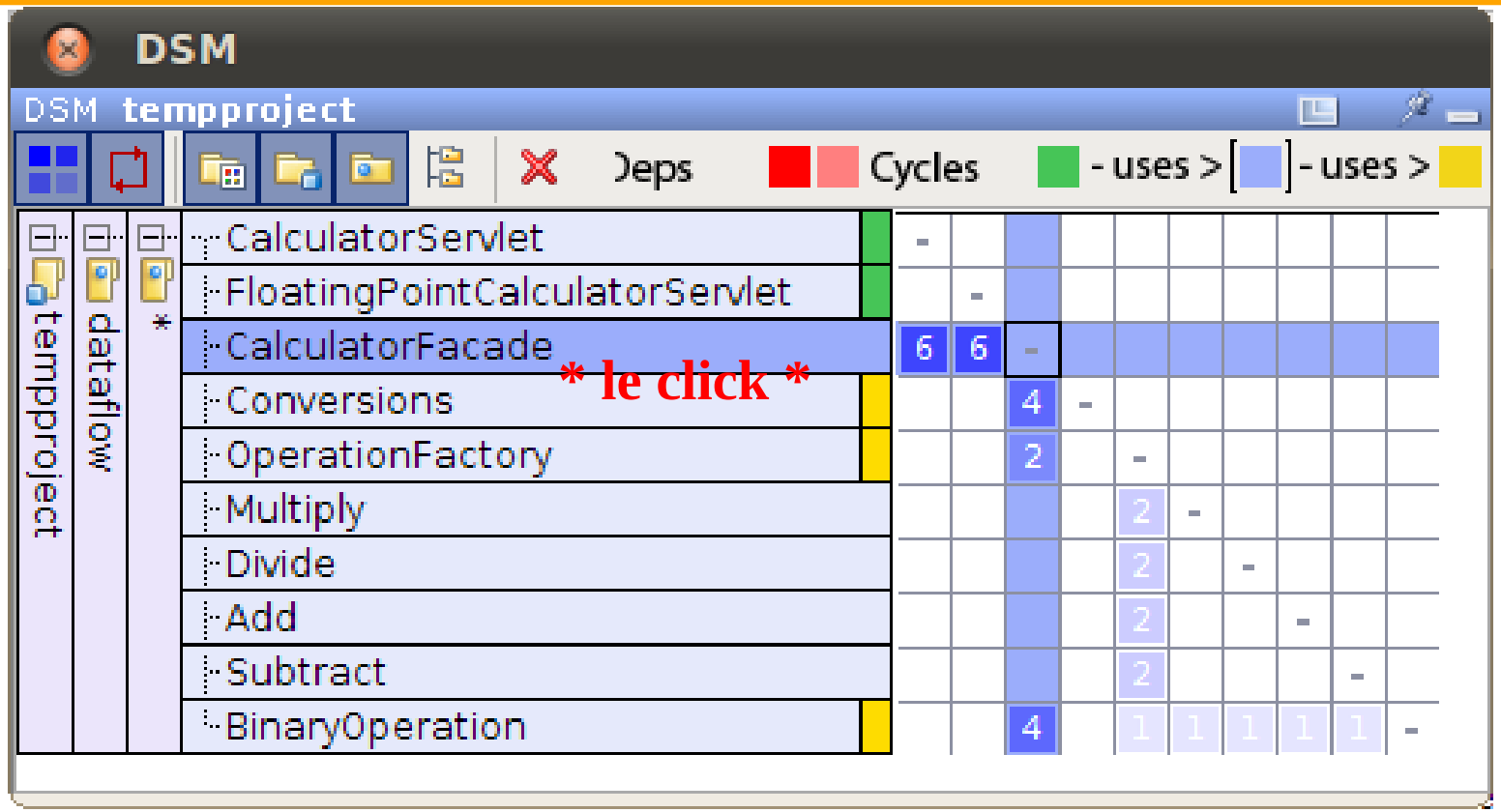
JetBRAINS

Dependency Structure Matrix

- Analyzes structure of complex projects
- Shows module, package, class dependencies
- Shows cyclic & backwards dependencies
- Helps eliminate illegal dependencies

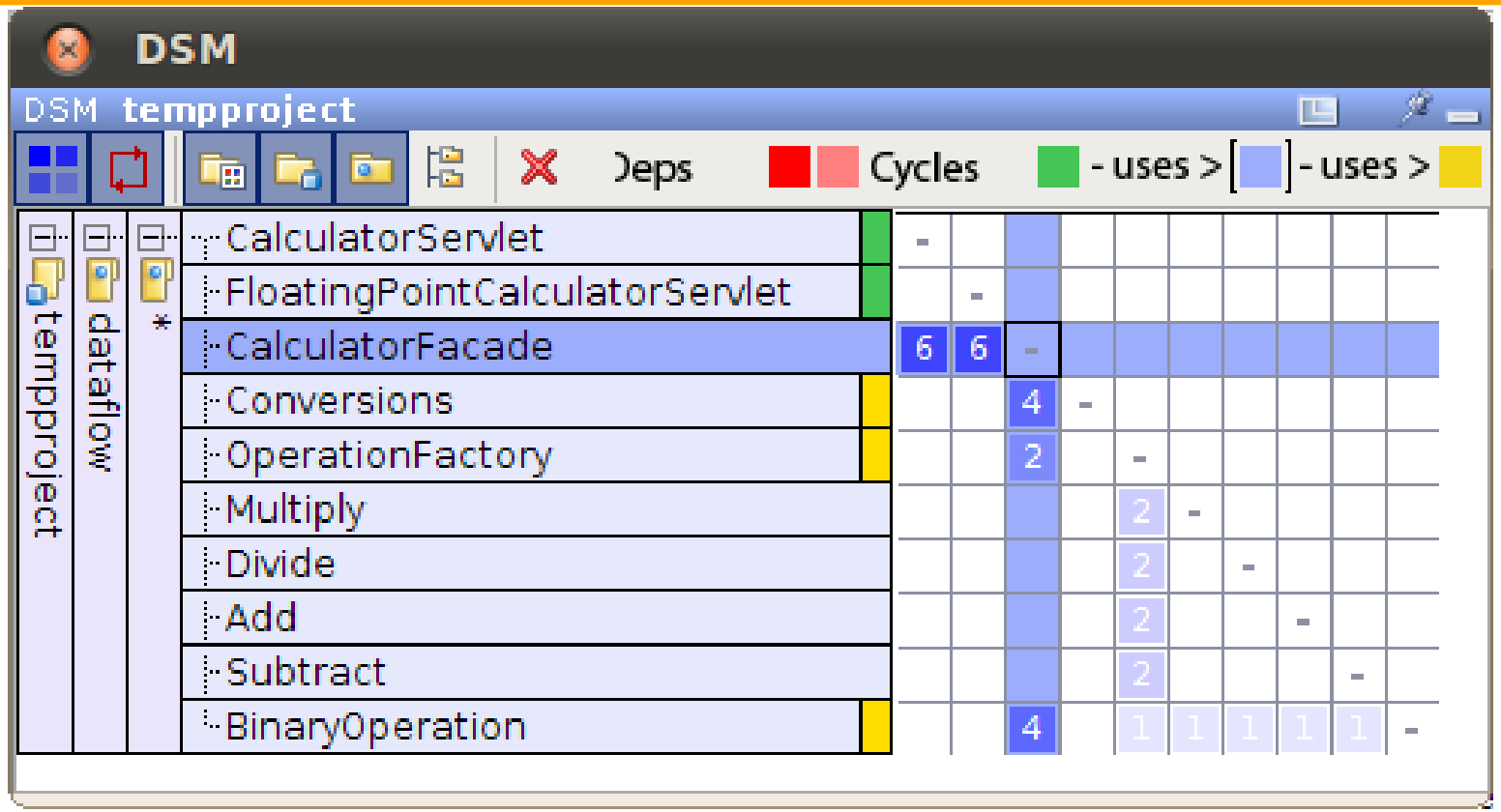


Classes on top depend-on classes below



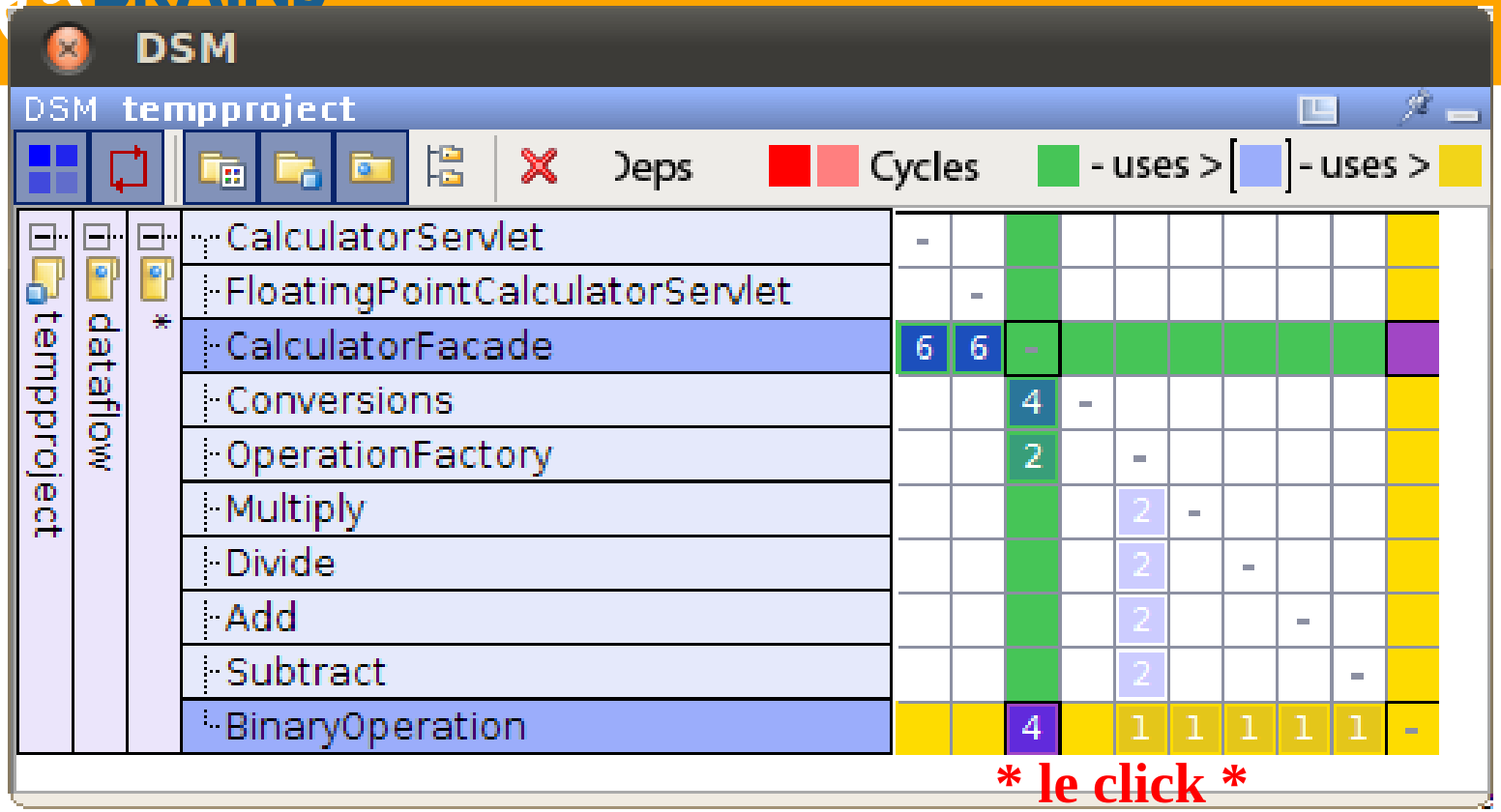
CalculatorFacade uses:

– Conversions, OperationsFactory & BinaryOperation



CalculatorFacade is used by

– CalculatorServlet & FPCalculatorServlet

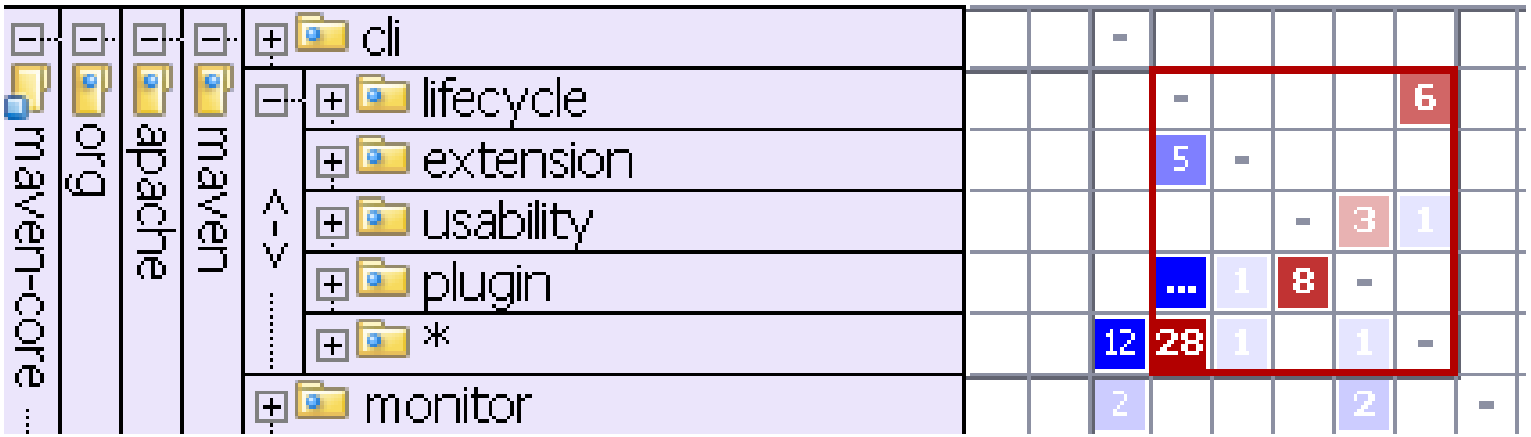


BinaryOperation is used 4 times by Facade

– Darker color == more dependencies

Green shows who BinaryOperation is “used by”

Yellow shows who BinaryOperation “uses”



Cyclic Dependencies can be highlighted

Modules can be collapsed/expanded

Dependency Structure Matrix

- Demos on JetBrains site & booth

- Feature Overview: http://goo.gl/0bcz3
- JetBrains Blog Post: http://goo.gl/fdj26
- Canoo Blog Post: http://goo.gl/M1hTY

JetBRAINS

Static Code Analysis

Code Inspections

JSR 305 and 308 Annotations

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis

JetBRAINS

Software Lifecycle

Code Inspections

JSR 305 and 308 Annotations

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis

Software Lifecycle

Code Inspections **every second**

JSR 305 and 308 Annotations **every second**

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis

Software Lifecycle

Code Inspections **every debug**

JSR 305 and 308 Annotations **every debug**

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis **every debug**

Dependency Analysis

Software Lifecycle

Code Inspections **every build**

JSR 305 and 308 Annotations

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis

JetBRAINS

Software Lifecycle

Code Inspections

JSR 305 and 308 Annotations

Duplicate Detection **every day**

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis

Software Lifecycle

Code Inspections

JSR 305 and 308 Annotations

Duplicate Detection

Stack Trace Analysis

Dataflow Analysis

Dependency Analysis **every release**

Learn More – Q & A

- My JetBrains.tv Screencasts: http://tv.jetbrains.net/tags/hamlet
- My IDEA blog: http://hamletdarcy.blogspot.com/search/label/IDEA
- Work's IDEA blog: http://www.canoo.com/blog/tag/idea/
- Main blog: http://hamletdarcy.blogspot.com
- YouTube channel: http://www.youtube.com/user/HamletDRC
- Twitter: http://twitter.com/hamletdrc
- IDEA RefCard from DZone: http://goo.gl/Fg4Af
- IDEA Keyboard Stickers: JetBrains Booth

- Share-a-Canooie – http://people.canoo.com/share/
- Hackergarten – http://www.hackergarten.net/