Refactor the legacy out!

Jean-Noël Rouvignac

jn.rouvignac@gmail.com

- Software Developer

- http://autorefactor.org

  http://autorefactor.org/html/samples.html

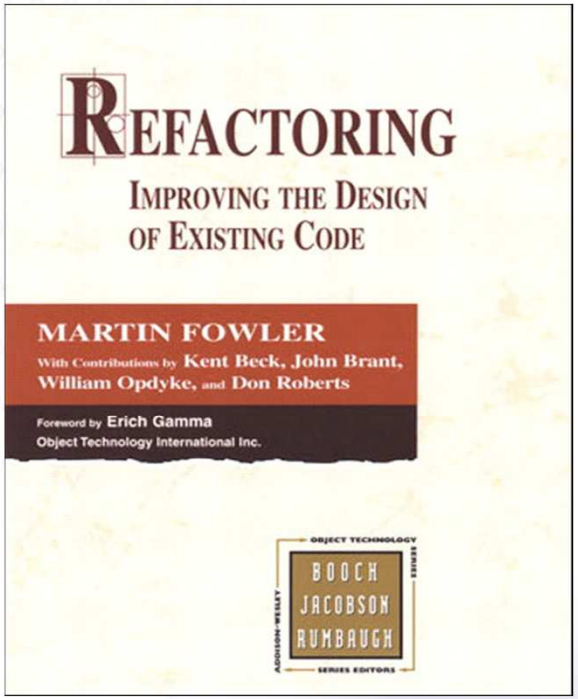
  https://github.com/JnRouvignac/AutoRefactor

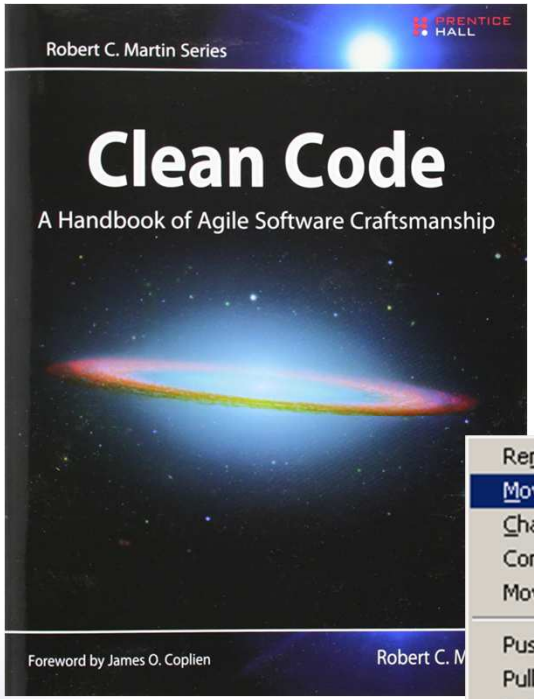
Agenda

- Personal experience
- Refactoring demo
- AutoRefactor

Experience(s)

- One library
  - 120 000 LOC => 100 000 LOC
  - 30-40 bugs fixed
- OpenDJ backend
  - 36 000 LOC => 26 000 LOC
- maintainability ↗





sonar

eclipse

Legacy Software

"Legacy code" often differs from its suggested alternative by actually working and scaling.

— Bjarne Stroustrup

Programming

Programming is the art of telling *human beings* what one wants the computer to do.

— Donald Knuth

Refactoring

If it stinks, change it.

— Grandma Beck, discussing child-rearing philosophy

Java design principles

- Reading code is more important than writing code
- Simplicity matters
- One language the same everywhere

— Brian Goetz

Agenda

- ~~Personal experience~~
- Refactoring demo
- AutoRefactor

Refactoring demo

Agenda

- ~~Personal experience~~
- ~~Refactoring demo~~
- AutoRefactor

Java 5?

```
Iterator it = myList.iterator();
while (it.hasNext()) {
    String s = it.next();
    ...
}
```

⇒

```
for (String s : myList) {
    ...
}
```

Regex?

Search

```
(final\s+)?Iterator\s+(\w+)\s*=\s*(\w+)\.iter
ator\(\)\s*;((\s|.)*)while\s*\(\s*\2\s*\.\s*h
asNext\s*\(\)\s*\)\s*\{\s*(\w+)\s*(\w+)\s*=\s
*\2\s*\.\s*next\s*\(\)\s*;
```

⇒

Replace

```
\4for (\6 \7 : \3) {
```

AutoRefactor

- Eclipse plugin
  - Open source: EPL et GPLV3+
  - v1.1.0 soon

- http://autorefactor.org

  http://autorefactor.org/html/samples.html

  https://github.com/JnRouvignac/AutoRefactor

AutoRefactor

- Java code modernization
- Rewrite for readability
- Coding conventions
- Best practices
- Use Eclipse parameters

- One step beyond formatting

AutoRefactor:

Why?

- No existing tools
- No tools which worked well enough?
- Mass refactoring

AutoRefactor demo

Refactoring rules

Map
String
Boolean
Comments
IfElseIf
Annotation
BigDecimal
Collection
InvertEquals
TestNGAssert
StringBuilder
UseMultiCatch

RemoveUnnecessaryCast
HotSpotIntrinsicedAPIs
RemoveUselessModifiers
RemoveUselessNullCheck
PrimitiveWrapperCreation
NoAssignmentInIfCondition
RemoveFieldsDefaultValues
CommonCodeInIfElseStatement
RemoveUnneededThisExpression
AddBracketsToControlStatement
RemoveUnnecessaryLocalBeforeReturn
WorkWithNullCheckedExpressionFirst

RemoveSemiColon
CommonIfInIfElse
PushNegationDown
RemoveEmptyLines
UseStringContains
VectorOldToNewAPI
CollectionContains
SimplifyExpression
UseDiamondOperator
CollapseIfStatement
DeadCodeElimination
ReduceVariableScope

What was in v1.0.0?

- Loop while there are refactoring opportunities
- Choose refactoring rules to run
- Parallel execution

- Maven + tycho
- Unit tests

What's in v1.1.0?

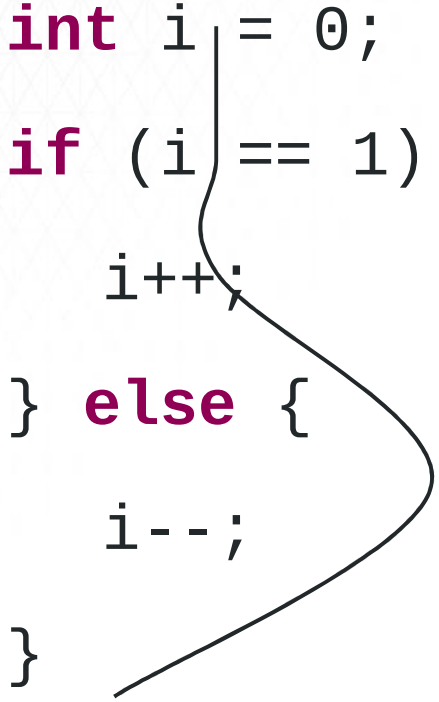
- Rules description inside the UI
- New refactoring rules:
  - multi-catch, diamond operator
  - common if condition in if/else statement
  - TestNG asserts
  - Collection + Map:
    - addAll(), containsAll(), removeAll(), isEmpty()
  - if (set.contains(el)) { set.add(el) } => if (!set.add(el)
  - remove semicolons, empty statements ';'
  - remove empty lines
  - remove useless method overrides

And after?

- Java 7, Java 8
  - TryWithResources
  - Rethrowing exceptions
  - InnerClassesToLambdas
  - LoopsToLambdas?
- Rewrite code for me please
- AutoRefactor only selected text
  - Eclipse quick fixes

And after?

Control flow graph



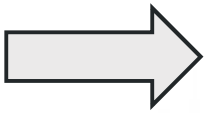
Allows:

- ReduceScopeOfVariable
- InlineVariable
- Foreach
- ...

And after?

Automated extract method

```
boolean found = false;
for (String s1 : strings1) {
    for (String s2 : strings2) {
        if (s1.equalsIgnoreCase(s2)) {
            found = true;
            break;
        }
    }
    if (found) {
        break;
    }
}
```



```
public boolean containsAnyIgnoreCase(
        Collection<String> strings1,
        Collection<String> strings2) {
    for (String s1 : strings1) {
        for (String s2 : strings2) {
            if (s1.equalsIgnoreCase(s2)) {
                returns true;
            }
        }
    }
    return false;
}
```

Call for Contributions

- Developers
  - Ideas
  - Refactoring rules
- Testers
  - Do you have a big code base?
- Designers
  - website
  - logo

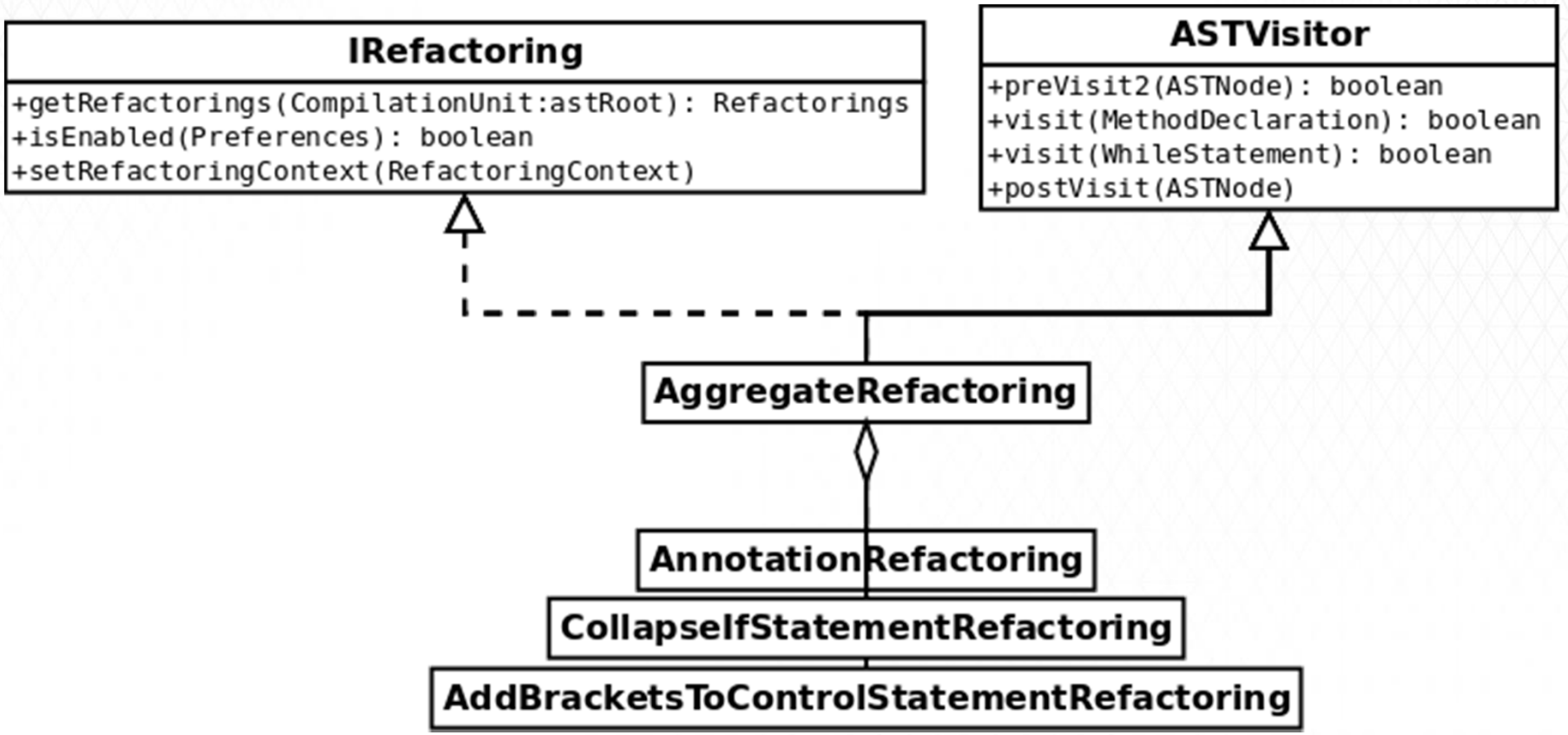
Questions?


eclipsecon Europe
Ludwigsburg, Germany, 3 - 5 November 2015
Evaluate the sessions at www.eclipsecon.org
+1
0
-1

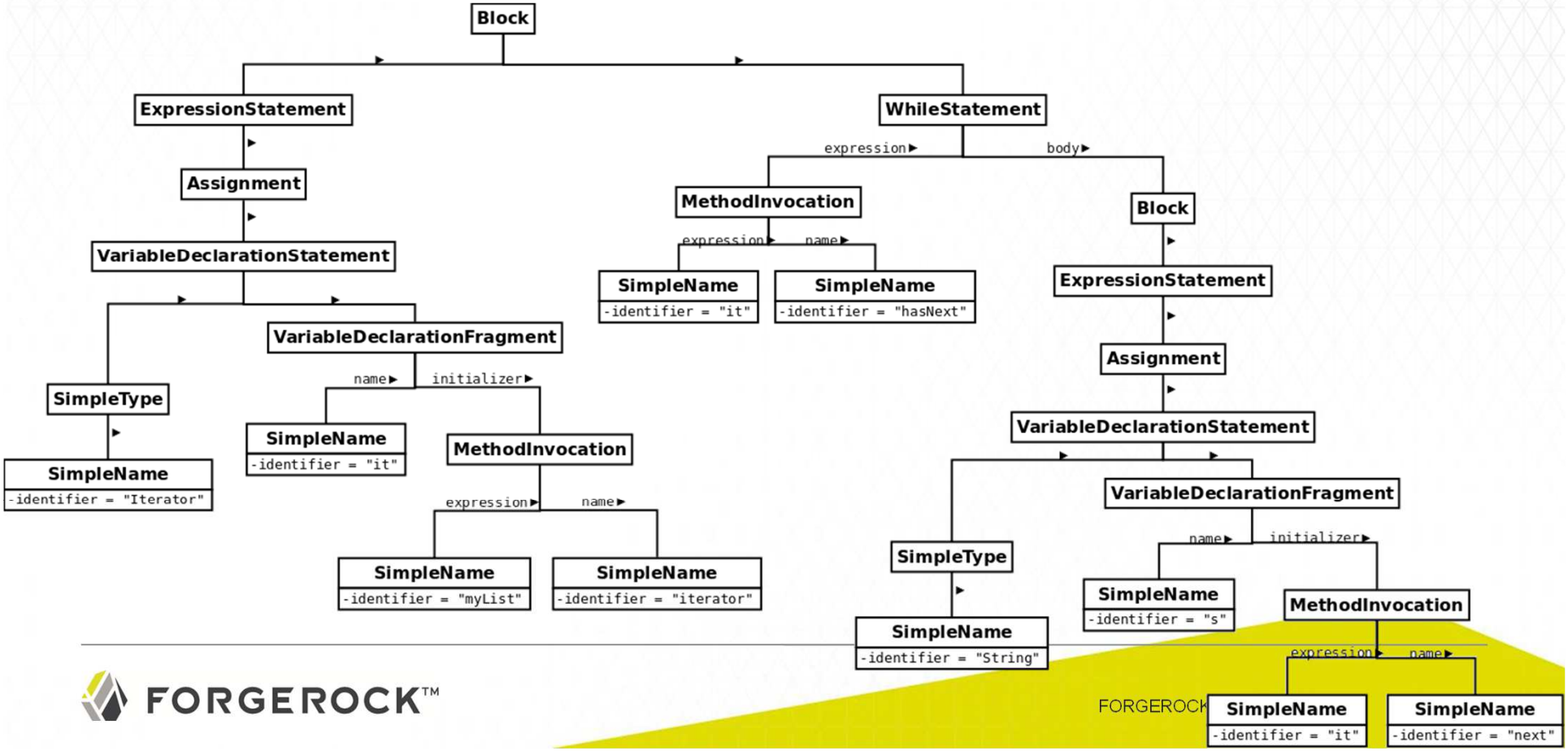
Architecture

- Eclipse platform
- Eclipse JDT
  - org.eclipse.jdt.core.dom
  - ASTParser
  - ASTVisitor
  - ASTRewrite

Architecture

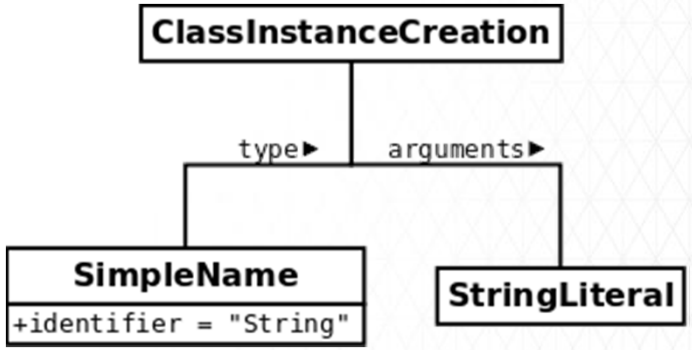


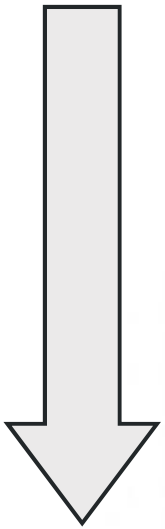
```
Iterator it = myList.iterator();
while (it.hasNext()) {
    String s = it.next();
    ...
}
```



Pattern matching

```
new String("Info")
```






```
"Info"
```

Pattern matching

```
public boolean visit(ClassInstanceCreation cic) {
    String string = "java.lang.String";
    if (hasType(cic, string)
            && cic.arguments().size() == 1
            && hasType(cic.arguments().get(0), string)) {

        ASTRewrite r = ...;

        r.replace(cic,

            ast.copySubtree(cic.arguments(0)));

        return DO_NOT_VISIT_SUBTREE;

    }
    return VISIT_SUBTREE;
}
```