

Code Analysis

Václav Pech

Software Developer, Technology

Evangelist

JetBrains, s.r.o.

vaclav@jetbrains.com

Agenda

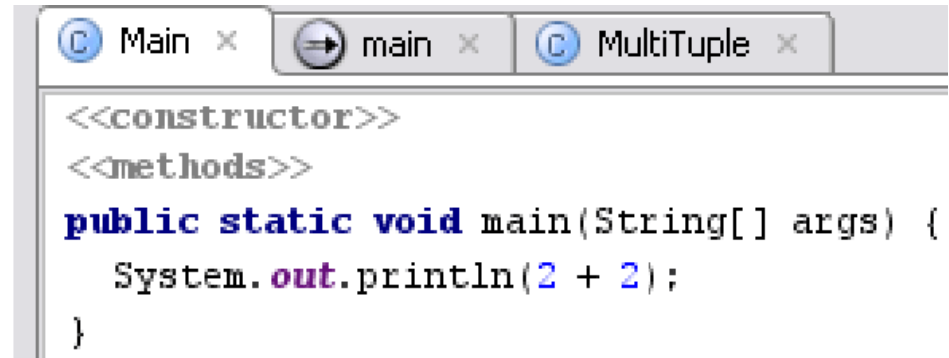
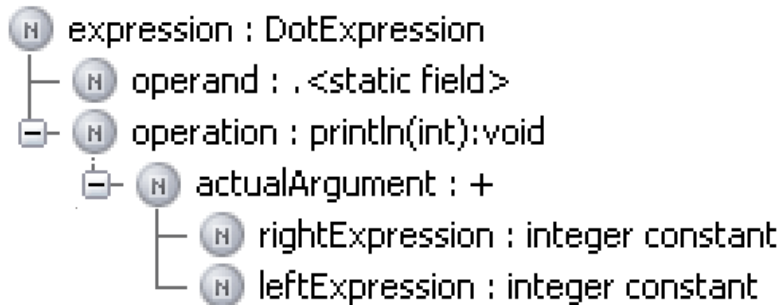
- Static Code Analysis
- Duplicate Detection
- Stacktrace Analysis
- Dataflow Analysis
- Dependency Analysis
- DSM



Static Code Analysis

- Searches AST for bug patterns
 - On demand
 - On the fly

Abstract Syntax Tree



```
<<constructor>>
<<methods>>
public static void main(String[] args) {
    System.out.println(2 + 2);
}
```

A quick peek into the sources

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

Advanced capabilities

- Quick fixes
- Profiles
- Suppressing false positives
- Spell-checker
- Command-line and CI integration

Multithreaded correctness

```
public class SharedLock {  
    private static long count = 0L;  
  
    public synchronized void increment() {  
        count++;  
    }  
}
```

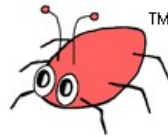
There's more than Java out there

- JavaScript
- CSS, HTML, XML
- GWT, Struts
- EE, ME

```
1.prodname span {  
    display: block;  
    margin: -3px 0 0 0;  
    padding: 0;  
    font-size: 0.8em;  
    font-weight: normal;  
    font-size: 70%;  
}
```

Property font-size is overwritten

```
var selectedQuantityIndex = volumeQuantities.length - 1;  
if ((quantity == null) || (isNaN(quantity))) {  
    // Comparison quantity == null may cause unexpected type coercion
```

Create your own bug patterns

Define

an AST sub-tree to search for

an AST sub-tree to replace it with

Helper annotations - JSR 305 & 308

- @Nullable, @NotNull, @Nls, @NonNls
- @PropertyKey, @Pattern, @Language
- @ThreadSafe, @GuardedBy
- @Immutable

@Pattern

```
abstract class Tool {  
    @Pattern("[a-zA-Z_0-9]+")  
    abstract String getId();  
}
```

```
class MyTool extends Tool {  
    String getId() {  
        return "ID (Not Valid)";  
    }  
}
```

Expression 'ID (Not Valid)' doesn't match pattern: [a-zA-Z_0-9]+ [more...](#) (Ctrl+F1)

Language injection

```
@Language("CSS")  
String cssDemo = "h2 { font-style: italic; }";
```

```
@Language("RegExp")  
String regexpDemo = "\\p{Alpha}(abc)\\1\\2";
```

Unresolved backreference

```
@Language("HTML")  
String xmlDemo = "<head><title>Test</title></hed>";
```

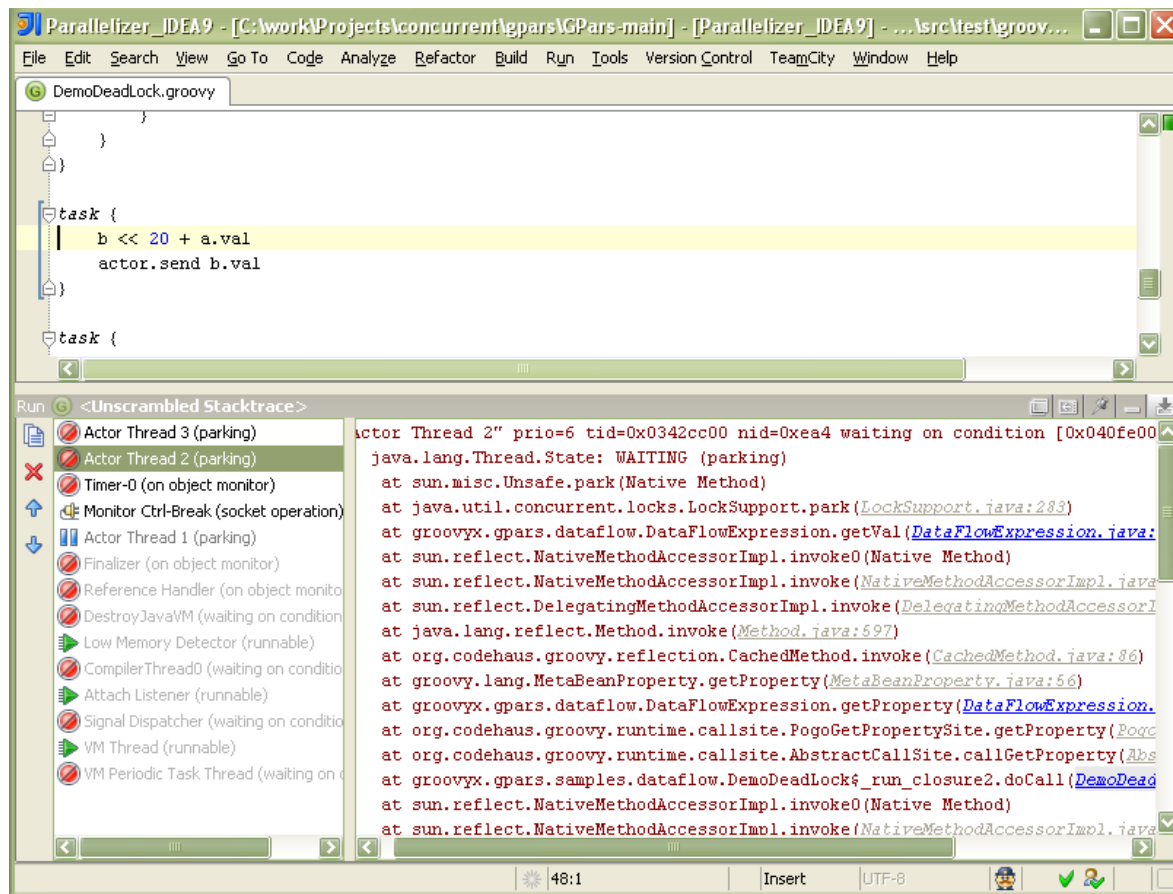
Concurrency – JCIP annotations

```
public class Cart {  
    private final Object cartLock = new Object();  
  
    @GuardedBy("cartLocks")  
    private final Map<String, Integer> contents = new HashMap<String, Integer>();  
  
    public int getProductQuantity(final String productName) {  
        return contents.get(productName);  
    }  
  
    public Set<String> getProducts() {  
        synchronized (this) {  
            return contents.keySet();  
        }  
    }  
  
    public void add()  
        synchronized
```

Access to field 'contents' outside of declared guards [less...](#) (Ctrl+F1)

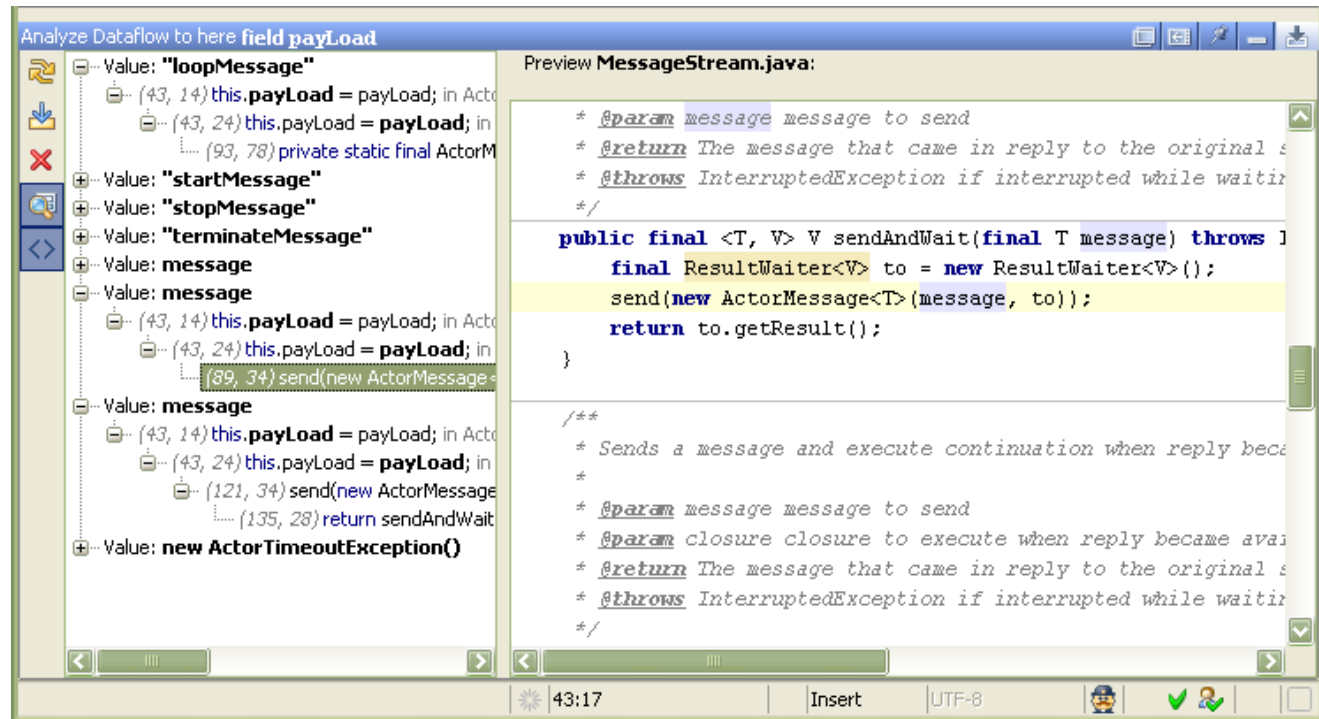
This inspection reports any accesses of fields declared as `@net.jcip.annotations.GuardedBy` which are not guarded by an appropriate synchronization structure.

Stacktrace Analysis

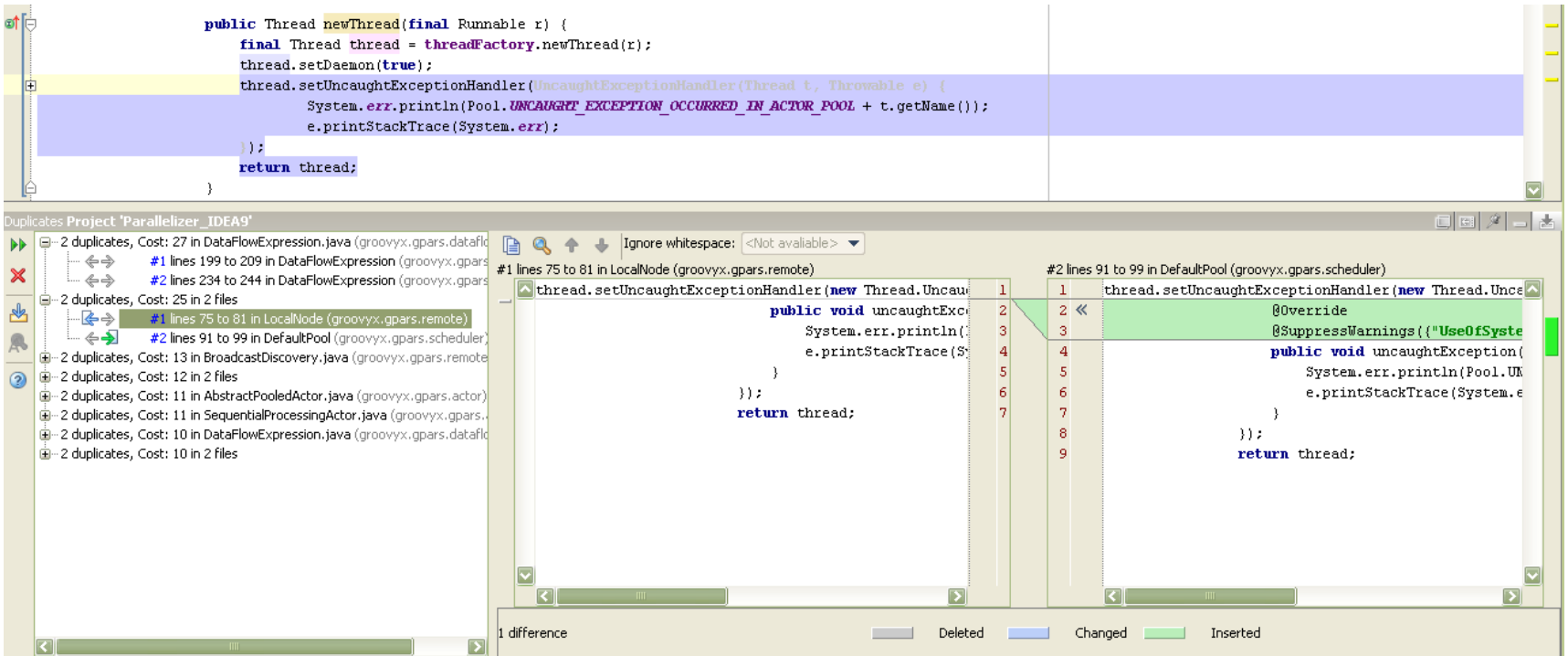


Dataflow Analysis

- From / To here



Duplicate detection



```

public Thread newThread(final Runnable r) {
    final Thread thread = threadFactory.newThread(r);
    thread.setDaemon(true);
    thread.setUncaughtExceptionHandler(new UncaughtExceptionHandler(Thread t, Throwable e) {
        System.err.println(Pool.UNCAUGHT_EXCEPTION_OCCURRED_IN_ACTOR_POOL + t.getName());
        e.printStackTrace(System.err);
    });
    return thread;
}

```

Duplicates Project 'Parallelizer_IDEA9'

- 2 duplicates, Cost: 27 in DataFlowExpression.java (groovyxx.gpars.dataflow)
- 2 duplicates, Cost: 25 in 2 files
- 2 duplicates, Cost: 13 in BroadcastDiscovery.java (groovyxx.gpars.scheduler)
- 2 duplicates, Cost: 12 in 2 files
- 2 duplicates, Cost: 11 in AbstractPooledActor.java (groovyxx.gpars.actor)
- 2 duplicates, Cost: 11 in SequentialProcessingActor.java (groovyxx.gpars.actor)
- 2 duplicates, Cost: 10 in DataFlowExpression.java (groovyxx.gpars.dataflow)
- 2 duplicates, Cost: 10 in 2 files

Ignore whitespace: <Not available>

#1 lines 75 to 81 in LocalNode (groovyxx.gpars.remote)

```

thread.setUncaughtExceptionHandler(new Thread.UncaughtExceptionHandler() {
    public void uncaughtException(Thread t, Throwable e) {
        System.err.println(Pool.UNCAUGHT_EXCEPTION_OCCURRED_IN_ACTOR_POOL + t.getName());
        e.printStackTrace(System.err);
    }
});
return thread;

```

#2 lines 91 to 99 in DefaultPool (groovyxx.gpars.scheduler)

```

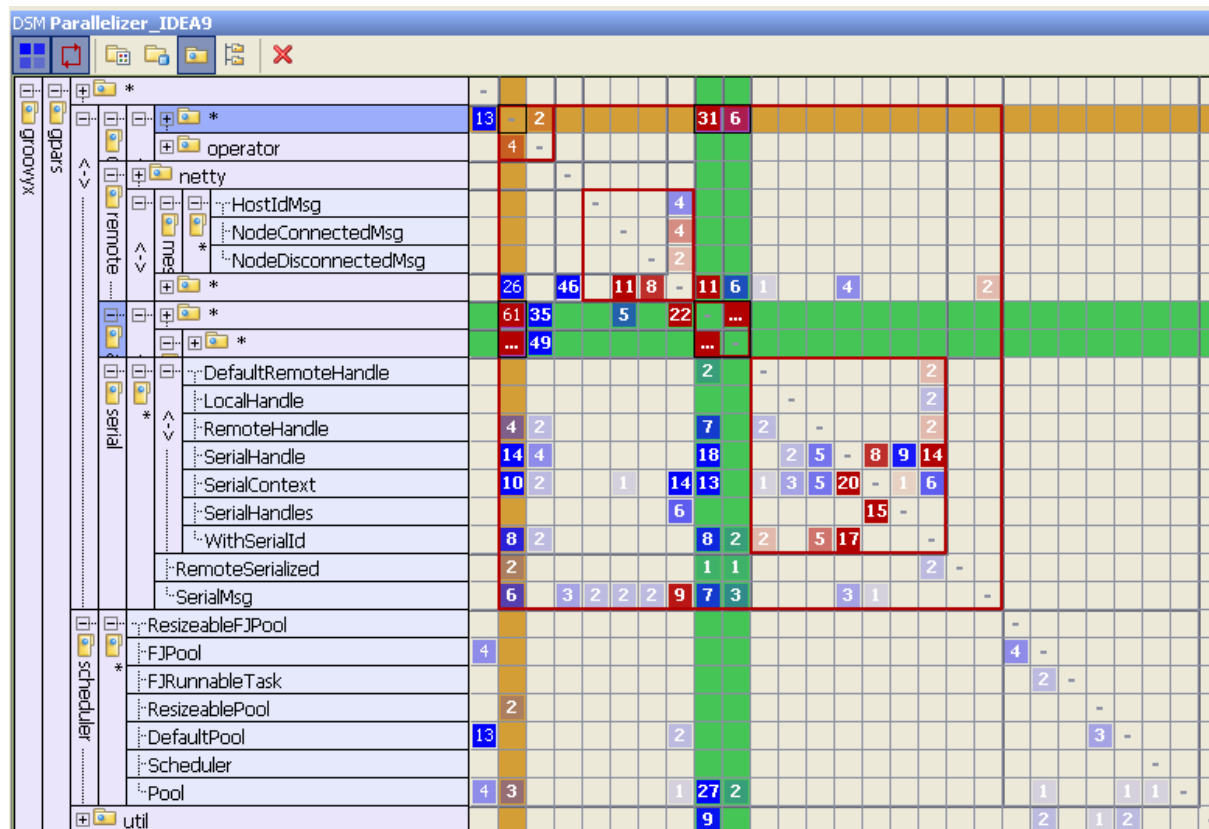
@Override
@SuppressWarnings({"UseOfSystemErr"})
public void uncaughtException(Thread t, Throwable e) {
    System.err.println(Pool.UNCAUGHT_EXCEPTION_OCCURRED_IN_ACTOR_POOL + t.getName());
    e.printStackTrace(System.err);
}
return thread;

```

1 difference

Deleted Changed Inserted

Dependency Structure Matrix



Summary

- Customize
- Use permanently
- Run continuously

So that you can sleep better!



Questions