eclipse-jdt / eclipse.jdt.ui Public

Code Issues 105 Pull requests 12 Actions Security Insights

Code

Jump to bottom

Improve the Safety of "Extract Local Variable" refactorings by identifying statements that may change the value of the extracted expressions #432 #439

Merged

jjohnstn merged 22 commits into `eclipse-jdt:master` from `chixiaye:issue432_ChangedValue` on Mar 25

Conversation 62 Commits 22 Checks 2 Files changed 38

chixiaye commented on Feb 26 • edited

Here is the PR based on the previous discussion in the issue-432.

What it does

`ChangedValueChecker` is the core part that is used to identify statements that may change the value of the extracted expressions. It first records the values that the selected expression depends on during the initialization phase. Then it calculates the path between the temp declaration and the last candidate extracted expression. Lastly, it validates intermediate statements in parallel to detect the intersection of the value updated by the intermediate code and the value read by the expression without duplication.

For `ExtractTempRefactoring.java`, in addition to the call of the `ChangedValueChecker` class, we have changed the original refactoring strategy when replacing the selected expression with temp declaration. Considering such example:

```
1 {
2   f();
3   int x=f();
4 }
5 {
6   int y=f();
7 }
```

If we select `f()` at Line 2, the refactored code shown as follows has obviously syntax errors:

```
1 {
2   int f=f();
3   int x=f;
}
{
  int y=f;
}
```

I think it is better to extract only once for the selected expression used as a statement. Because such an expression tends to modify the system state (otherwise it would be useless).

How to test

Tests have been appended in `ExtractTempTests` to verify the correctness of the patch. Test-140 is added to test for a variable the selected expression depends on has been changed within the method invocation. Test-141/Test-142 is to test the case of loop/condition statement to ensure that the path is not just calculated on the basis of offsets. Test-143, Test-144, and Test-145 were designed to test complex scenarios to ensure the robustness of the code concerning calculating the path. Test-146 and Test-147 were added to verify the identification of changed values under different cases.

Notably, we disabled Test-46, Test-110, and Test-111 because of the changed refactoring strategy. Besides, Test-73 was also excluded due to the added more precise flow analysis.

Author checklist

- ☑ I have thoroughly tested my changes
- ☑ The change is following the coding conventions
- ☑ I have signed the Eclipse Contributor Agreement (ECA)

liuhuigmail mentioned this pull request on Feb 28

Improve the Safety of "Extract Local Variable" refactorings by identifying statements that may change the value of the extracted expressions #432

Closed

jjohnstn requested changes on Mar 2

View reviewed changes

jjohnstn left a comment

There is a question of logic brought up by a few tests that you have chosen to ignore that I believe should work as there are no state changes involved. I will wait until those are answered to do more in-depth reviewing as the code may change. There are other minor coding issues.

org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test144_out.java Outdated Show resolved

org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test145_out.java Outdated Show resolved

...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java Outdated Show resolved

...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java Outdated Show resolved

...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java Outdated Show resolved

...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java Outdated Show resolved

....ui/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/UnsafeCheckTester.java Outdated Show resolved

...ore refactoring/org/eclipse/jdt/internal/corext/refactoring/code/ExtractTempRefactoring.java Outdated Show resolved

...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java Outdated Show resolved

....ui/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/SideEffectChecker.java Show resolved

chixiaye commented on Mar 5

@jjohnstn Thanks for your kind comments. In the latest version, I revised the following parts:

1. I updated the naming rules for introducing multiple new variables in one refactoring. This rule is that the smaller the offset value of the variable declaration position, the smaller the additional number, and the variable without an extra number will always come first.

2. For the disabled tests. `f` method in Test-46 will have StackOverflowError at runtime. So I replaced `f` method with a more reasonable method. Besides, as you suggested, I have modified the Test-73 to be valid. As for Test-110 and Test-111, `calculateCount();` is a statement that does nothing. Because its return-value does not read by any variable/object and `calculateCount()` method does not change the state of the program. I think that most of the time, only one method call in a single-line statement indicates that such a method has side effects, I suggest only replacing this expression and keeping other occurrences. So I modified the implementation of `calculateCount()` to make the tested code look closer to actual development.

3. For the change descriptions, I updated all the descriptions(including previous PRs) to make sure they are consistent with the patches.

jjohnstn requested changes on Mar 9

View reviewed changes

jjohnstn left a comment

Thanks. Looks much better. Good new tests of the new functionality. Just a few old tests to put back and a new recursive test to add. Please reply to each review comment either by stating it is done and then we can resolve the comment or add some information if you don't agree with the comment or there is further information required.

`org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test110_in.java` Outdated Show resolved

`org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test110_out.java` Outdated Show resolved

`org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test111_in.java` Outdated Show resolved

`org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test46_in.java` Outdated Show resolved

`...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java` Outdated Show resolved

jjohnstn requested changes on Mar 11

View reviewed changes

`org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test110_out.java` Outdated Show resolved

iloveeclipse commented on Mar 17

@chixiaye : unfortunately I saw this a bit late.
I see you are always doing merge of master into your branch. Please avoid doing that, instead please **rebase** your branch on top of master. This is needed to have a clean git history state.

☺

chixiaye commented on Mar 17

> **@chixiaye** : unfortunately I saw this a bit late. I see you are always doing merge of master into your branch. Please avoid doing that, instead please **rebase** your branch on top of master. This is needed to have a clean git history state.

Okay. Thanks for the reminder. I am sorry and promise this won't happen again.

☺ 👍 1

chixiaye force-pushed the `issue432_ChangedValue` branch from **`be15ebe`** to **`a2c939c`** Compare
2 months ago

chixiaye requested a review from **jjohnstn** 2 months ago

jjohnstn requested changes on Mar 22

View reviewed changes

`...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java` Outdated Show resolved

`....ui/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/UnsafeCheckTester.java` Show resolved

`...t.ui.tests.refactoring/test cases/org/eclipse/jdt/ui/tests/refactoring/ExtractTempTests.java` Outdated Show resolved

`org.eclipse.jdt.ui.tests.refactoring/resources/ExtractTemp/canExtract/A_test110_out.java` Outdated Show resolved

jjohnstn approved these changes on Mar 22

View reviewed changes

jjohnstn requested changes on Mar 22

View reviewed changes

jjohnstn left a comment

Thanks for the changes. The code is working as expected and tests are working without disablement. I have added some review comments/questions regarding the ChangedValueChecker. I also have one test below that isn't working as expected:

```
        public void foo(int k) {
                {
                        f();
                        switch (k) {
                        case 3:
                                f();
                                int x=f();
                                break;
                        default:
                                int y=f();
                        }
                }
        }
```

The test works fine if I choose the top f() call. If I choose the one in case 3: it does not change the following statement declaring x for some reason.

...i/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/Changed ValueChecker.java Outdated

```
+                                       });
+                               }
+                               try {
+                                       threadPool.shutdown();
```

jjohnstn on Mar 22

The orderly shutdown could take time. A check should be made for fConflict being true and then just do a shutdownNow(). Have you done any performance checks on a large routine being checked?

chixiaye on Mar 23

I added the new check for `fConflict`. `awaitTermination` can ensure the time cost is limited to less than 5s. `shutdownNow` is used in `finally` block to avoid other exceptions. As for time performance, about 97% of refactorings can be finished within 5 seconds according to the evaluation in real projects in the previous experiment. Considering the submitted patch is much more simple than our original implementation, the actual time cost would be less.

Reply...

...i/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/ChangedValueChecker.java Outdated Show resolved

...i/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/ChangedValueChecker.java Outdated Show resolved

...i/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/ChangedValueChecker.java Show resolved

...i/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/ChangedValueChecker.java Show resolved

...i/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/ChangedValueChecker.java Show resolved

chixiaye force-pushed the `issue432_ChangedValue` branch from **72b5934** to **0b94333** 2 months ago Compare

chixiaye commented on Mar 23

> Thanks for the changes. The code is working as expected and tests are working without disablement. I have added some review comments/questions regarding the ChangedValueChecker. I also have one test below that isn't working as expected:
>
> ```
>         public void foo(int k) {
>                 {
>                         f();
>                         switch (k) {
>                         case 3:
>                                 f();
>                                 int x=f();
>                                 break;
>                         default:
>                                 int y=f();
>                         }
>                 }
>         }
> ```
>
> The test works fine if I choose the top f() call. If I choose the one in case 3: it does not change the following statement declaring x for some reason.

Hi, I re-checked the code and fixed this problem and attached Test-151 and Test-152 to validate.

chixiaye requested a review from **jjohnstn** 2 months ago

jjohnstn approved these changes on Mar 24

View reviewed changes

jjohnstn requested changes on Mar 24

View reviewed changes

...i/core refactoring/org/eclipse/jdt/internal/corext/refactoring/util/ChangedValueChecker.java Outdated

```
+                          }
+                          threadPool.shutdown();
+                          threadPool.awaitTermination(5, TimeUnit.SECONDS);
+                          while (!threadPool.isTerminated() && fConflict == true)
```

jjohnstn on Mar 24

Last comment. You only want to loop if fConflict is false which means that nothing has found a conflict yet. Once a conflict is found, there is no need to do any more processing or waiting for termination because nothing more will be found. That is why I suggested checking if fConflict is true before even asking for termination or shutdown since the finally clause will shutdownNow() regardless.

chixiaye on Mar 25

I think I understand what you mean. I modified the code to only call `shutdown` when `fConflict` is false, and the detection would be interrupted if `fConlict` is true.

jjohnstn on Mar 25

Yes, change looks good.

Reply...

chixiaye force-pushed the `issue432_ChangedValue` branch from **b6d7f5e** to **0b94333** Compare
2 months ago

chixiaye and others added 12 commits 2 months ago

update Copyright	2443932
Improve the Safety of Extract Local Variable refactorings by identify… …	0294ee6
Discard two test cases	fc261e7
Update ExtractTempTests1d7.java	b1ead8b
Update ExtractTempTests1d8.java	78b0fb7
Update ExtractTempTests1d8.java	cfeea97
revision about test cases and coding issues	8da0166
solve the casuse of failed tests	4d1d6f5
replace tab with space	e095a04
change test-47	7247bbb
reformat	7449095
restore tests	ba87335

chixiaye and others added 8 commits 2 months ago

restore the original strategy and add a new test	dc3ce7e
fix problems for AssistQuickFixTest and AdvancedQuickAssistTest	6338ee2
replace thread list with threadPool	57db006
enhanced closing threadpool's part	03b5f04
fixed new variable scope problem while replacing	c85d9d5
Minor corrections.	2c97b91
Some revisions based on comments	ea32ac5
fix bug while extracting in switchstmts	✗ 569efc1

chixiaye force-pushed the `issue432_ChangedValue` branch from **0b94333** to **569efc1** Compare
2 months ago

chixiaye added 2 commits 2 months ago

`Optimize the process of closing the thread pool` ✗ `ca76881`

`fix typo error` ● `43cf391`

chixiaye requested a review from **jjohnstn** 2 months ago

jjohnstn approved these changes on Mar 25

View reviewed changes

jjohnstn merged commit **`c0e779e`** into `eclipse-jdt:master` on Mar 25

4 checks passed

View details

jjohnstn commented on Mar 25

Thanks **@chixiaye** for all the changes. Merged.

👍 1

jjohnstn added the enhancement label on Mar 25

jjohnstn added this to the **4.28 M1** milestone on Mar 25

Reviewers

jjohnstn ✓

Assignees

No one assigned

Labels

enhancement

Projects

None yet

Milestone

4.28 M1

Development

Successfully merging this pull request may close these issues.

None yet

3 participants