RefactorHub:
A Commit Annotator for Refactoring

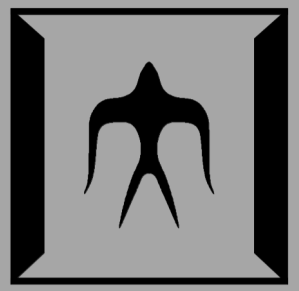
github.com/
salab/**RefactorHub**

Ryo Kuramoto

Motoshi Saeki

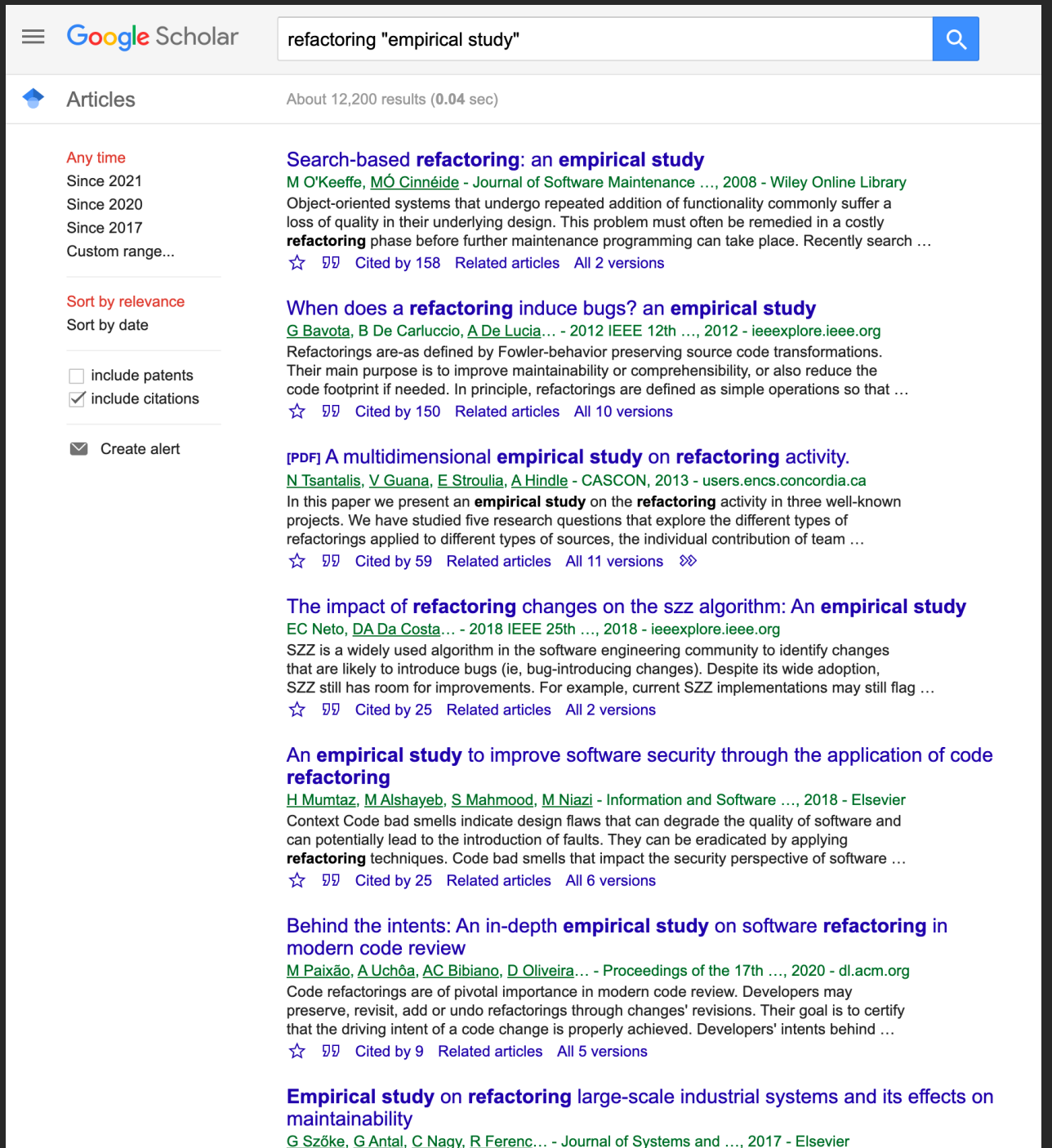
Shinpei Hayashi

Tokyo Tech.
Japan



Empirical studies on refactoring

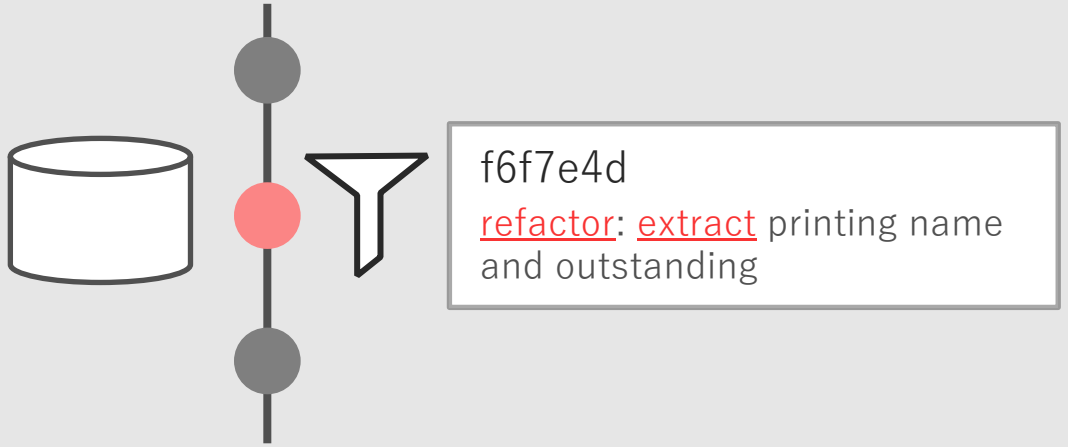
We need refactoring data



How to collect refactoring data

Using detection tools
(e.g.. RefactoringMiner, RefDiff)

- ✓ Accuracy
- ✗ Type coverage
- ✓ Detail information



Using self-affirmed commits
(commit message filtering)

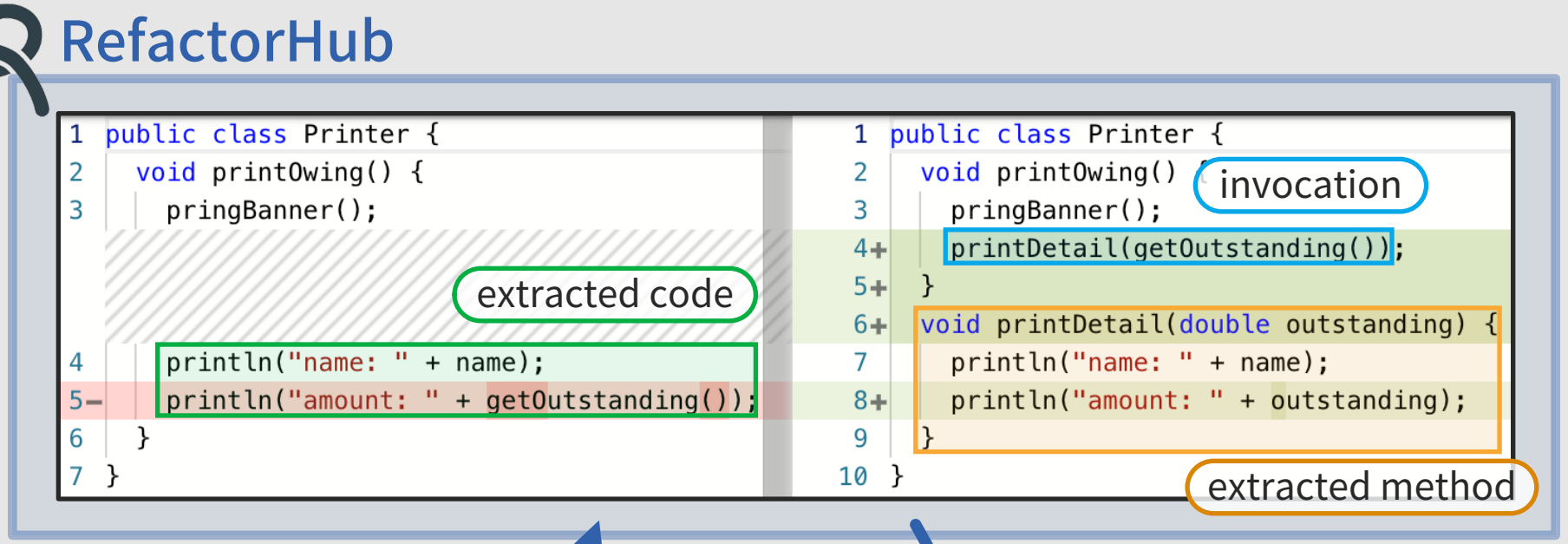
- ✗ Accuracy
- ✓ Type coverage
- ✗ Detail information

RefactorHub:

A Commit Annotator for Refactoring

Annotation Approach

RefactorHub



Input (Commit)

f6f7e4d

refactor: extract printing name and outstanding

✗ Accuracy
✗ Detail information

Output (JSON)

```
{
 "type": "ExtractMethod",
 ~
 "extracted method": {
  "type": "MethodDeclaration",
  ~
  "path": "src/main/Printer.java",
  "range": {
   "startLine": 6, "startColumn": 2,
   "endLine": 9, "endColumn": 3
   ~
```

✓ Accuracy
✓ Detail information

Refactoring info.

Save Discard

Extract Method

Description
Extract Method package clearCallbacksAndListener() : void extracted from private checkCurrentDimens() : void in class com.bumptech.glide.request.target.ViewTarget.SizeDeterminer

bumptech / glide / 0d4b279

Clear callbacks from ViewTargets in onLoadCleared.

Created by MOE: http://code.google.com/p/moe-java
MOE_MIGRATED_REVID=95763399

judds committed on 2015/6/12 4:11:07

Before Elements

extracted code

target method

element name

ElementType

multiple

Refactoring parameters (before refactoring)

Lock

```
int currentHeight = getViewHeightOrParam();
if (!isSizeValid(currentWidth) || !isSizeValid(currentHeight)) {
  return;
}

notifyCbs(currentWidth, currentHeight);
// Keep a reference to the layout listener and remove it here
// rather than having the observer remove itself because the observer
// we add the listener to will be almost immediately merged into
// another observer and will therefore never be alive. If we instead
// keep a reference to the listener and remove it here, we get the
// current view tree observer and should succeed.
ViewTreeObserver observer = view.getViewTreeObserver();
if (observer.isAlive()) {
  observer.removeOnPreDrawListener(layoutListener);
}
layoutListener = null;
}

public void getSize(SizeReadyCallback cb) {
  int currentWidth = getViewWidthOrParam();
  int currentHeight = getViewHeightOrParam();
  if (isSizeValid(currentWidth) && isSizeValid(currentHeight)) {
    cb.onSizeReady(currentWidth, currentHeight);
  } else {
    // We want to notify callbacks in the order they were added and we only expect one
    // callbacks to
    // be added a time, so a List is a reasonable choice.
    if (!cbs.contains(cb)) {
      cbs.add(cb);
    }
    if (layoutListener == null) {
      final ViewTreeObserver observer = view.getViewTreeObserver();
```

Refactoring parameters (after refactoring)

Lock

```
int currentHeight = getViewHeightOrParam();
if (!isSizeValid(currentWidth) || !isSizeValid(currentHeight)) {
  return;
}

notifyCbs(currentWidth, currentHeight);
clearCallbacksAndListener();

}

void getSize(SizeReadyCallback cb) {
  int currentWidth = getViewWidthOrParam();
  int currentHeight = getViewHeightOrParam();
  if (isSizeValid(currentWidth) && isSizeValid(currentHeight)) {
    cb.onSizeReady(currentWidth, currentHeight);
  } else {
    // We want to notify callbacks in the order they were added and we only expect one
    // callbacks to
    // be added a time, so a List is a reasonable choice.
    if (!cbs.contains(cb)) {
      cbs.add(cb);
    }
    if (layoutListener == null) {
      final ViewTreeObserver observer = view.getViewTreeObserver();
```

Code difference

After Elements

extracted code

invocation

target method

element name

ElementType

multiple

Modified files

...m/bumptech/glide/request/target/ViewTarget.java

...ptech/glide/request/target/ImageViewTarget.java

...m/bumptech/glide/request/target/ViewTarget.java

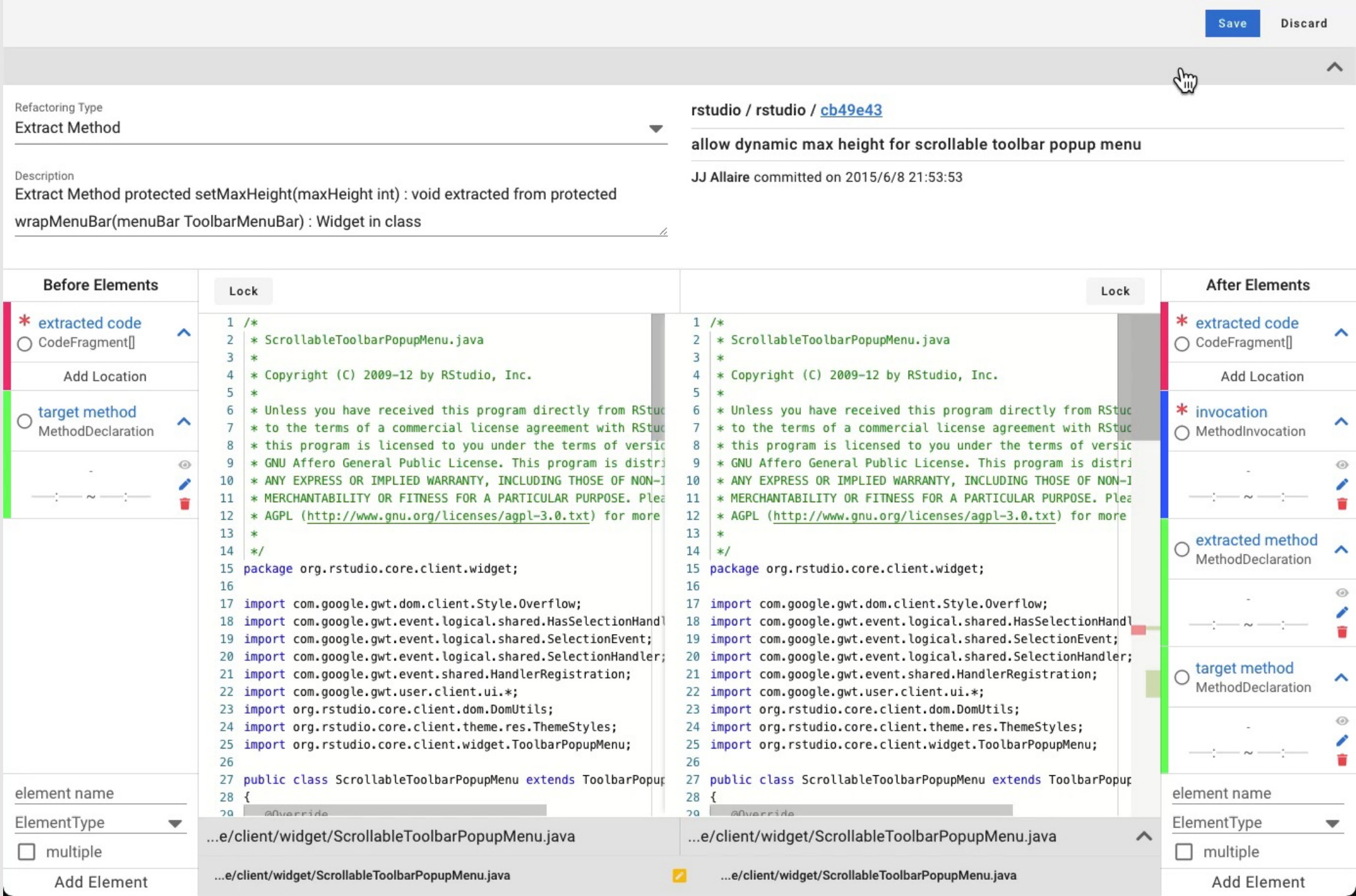
Annotating Extract Method

Before:

- extracted code
- target method

After:

- extracted code
- invocation
- extracted method
- target method



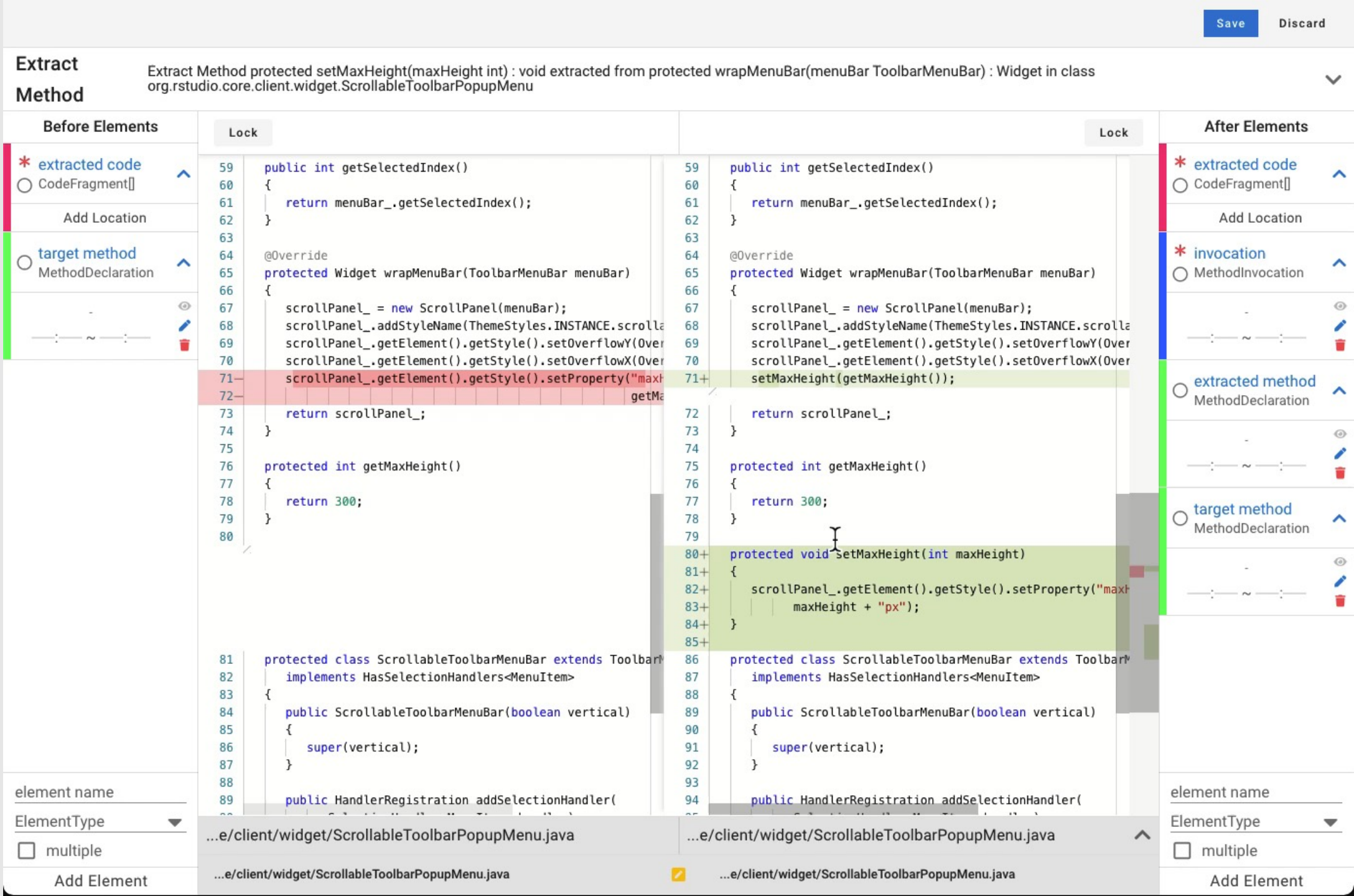
Annotating Extract Method

Before:

- extracted code
- target method

After:

- extracted code
- invocation
- extracted method
- target method



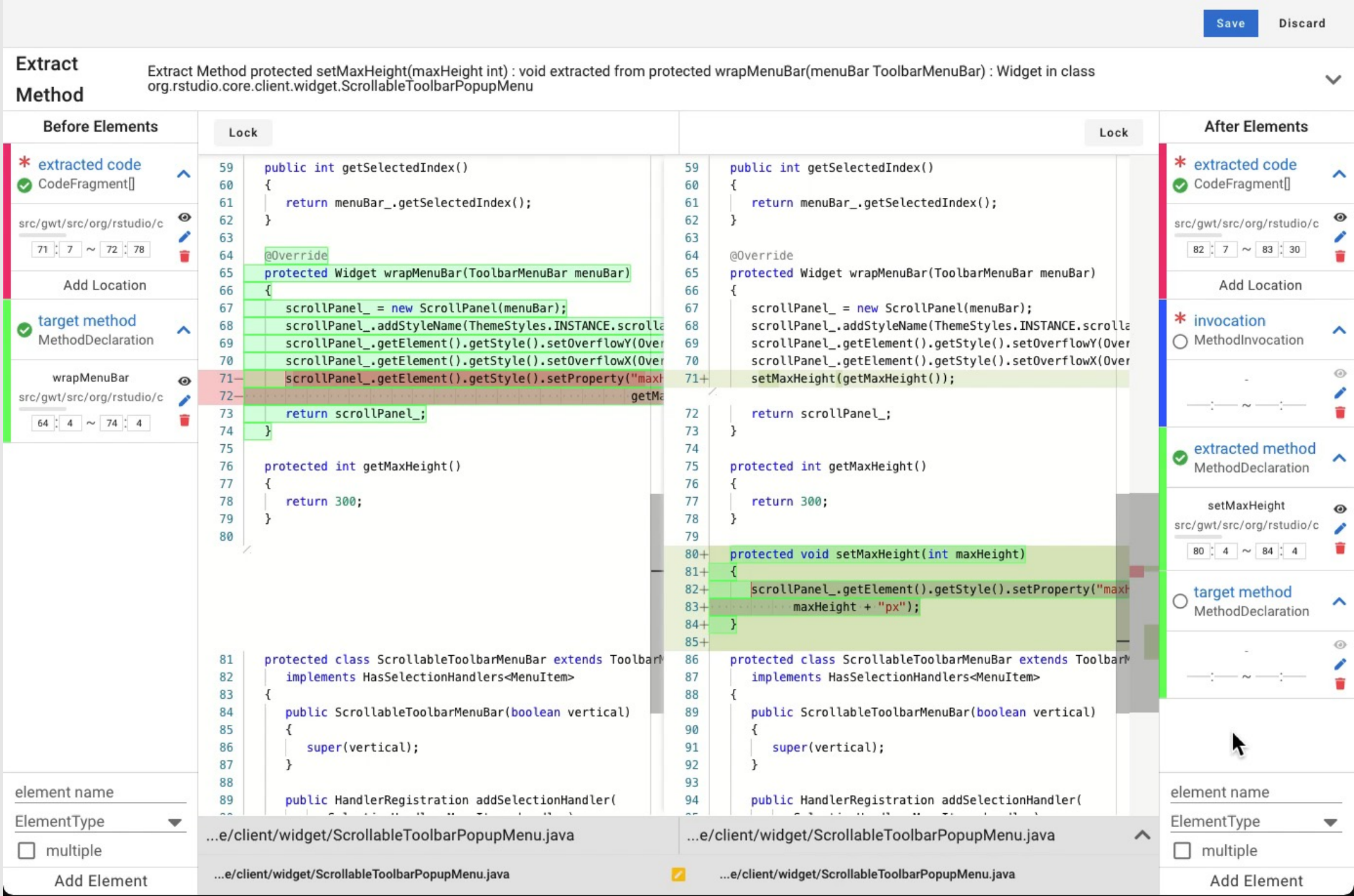
Annotating Extract Method

Before:

- extracted code
- target method

After:

- extracted code
- invocation
- extracted method
- target method



RefactorHub:
A Commit Annotator for Refactoring

github.com/
salab/**RefactorHub**

Ryo Kuramoto

Motoshi Saeki

Shinpei Hayashi

Tokyo Tech.
Japan