

ReachHover: Preserving Context in Data-flow Analysis

James Yoo^{a, b}

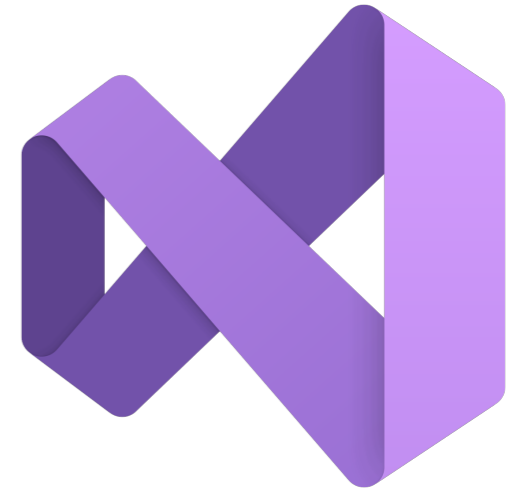
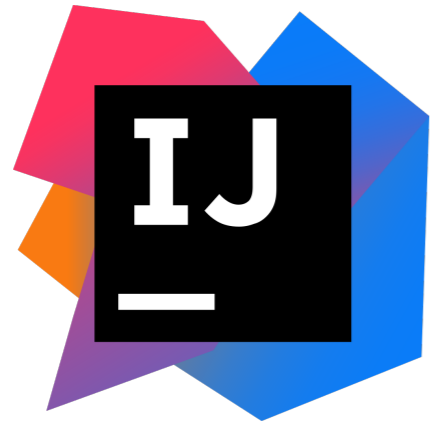
Gail C. Murphy^b

^aUniversity of Washington

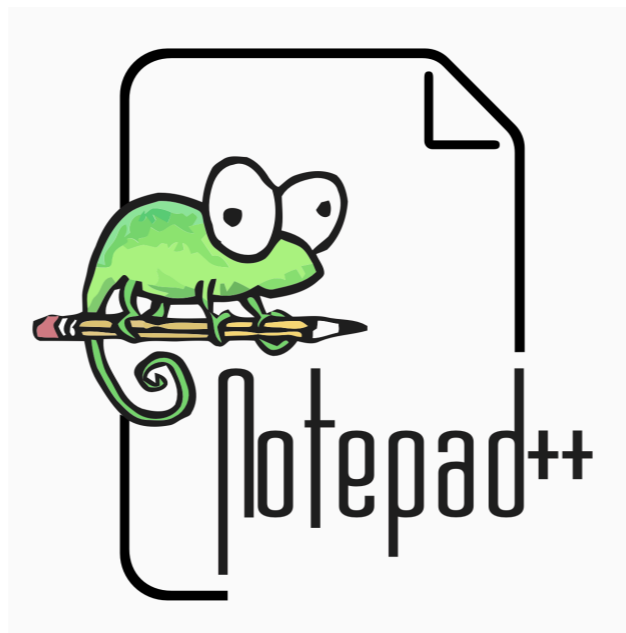
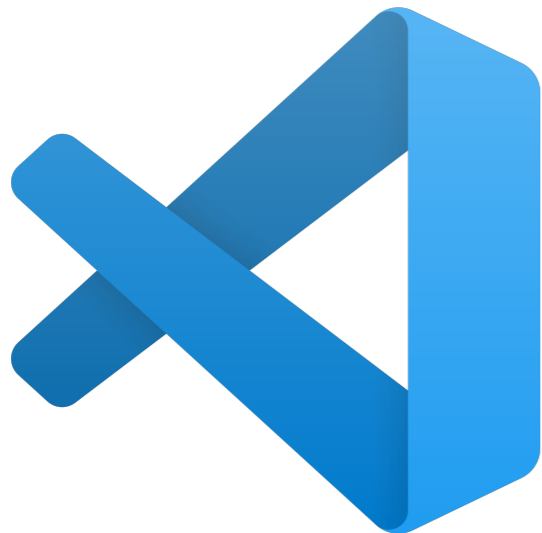


^bUniversity of British Columbia





**Who here has
used an IDE?**



**Do You Ever Struggle To
Maintain Context in Your IDE?**

Files

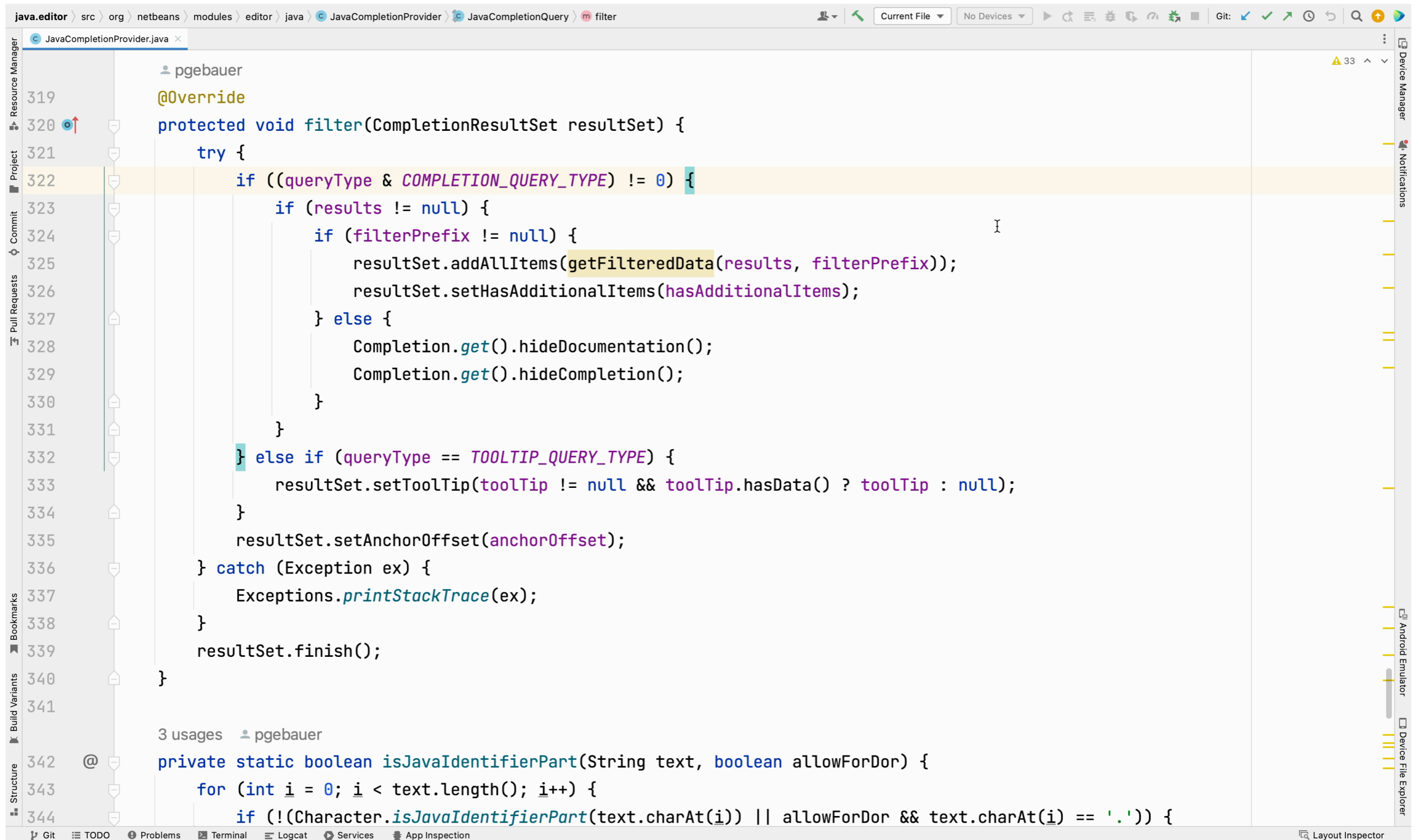
Classes

**Do You Ever Struggle To
Maintain Context in Your IDE?**

Methods

Locations

Preserving Context in IDEs



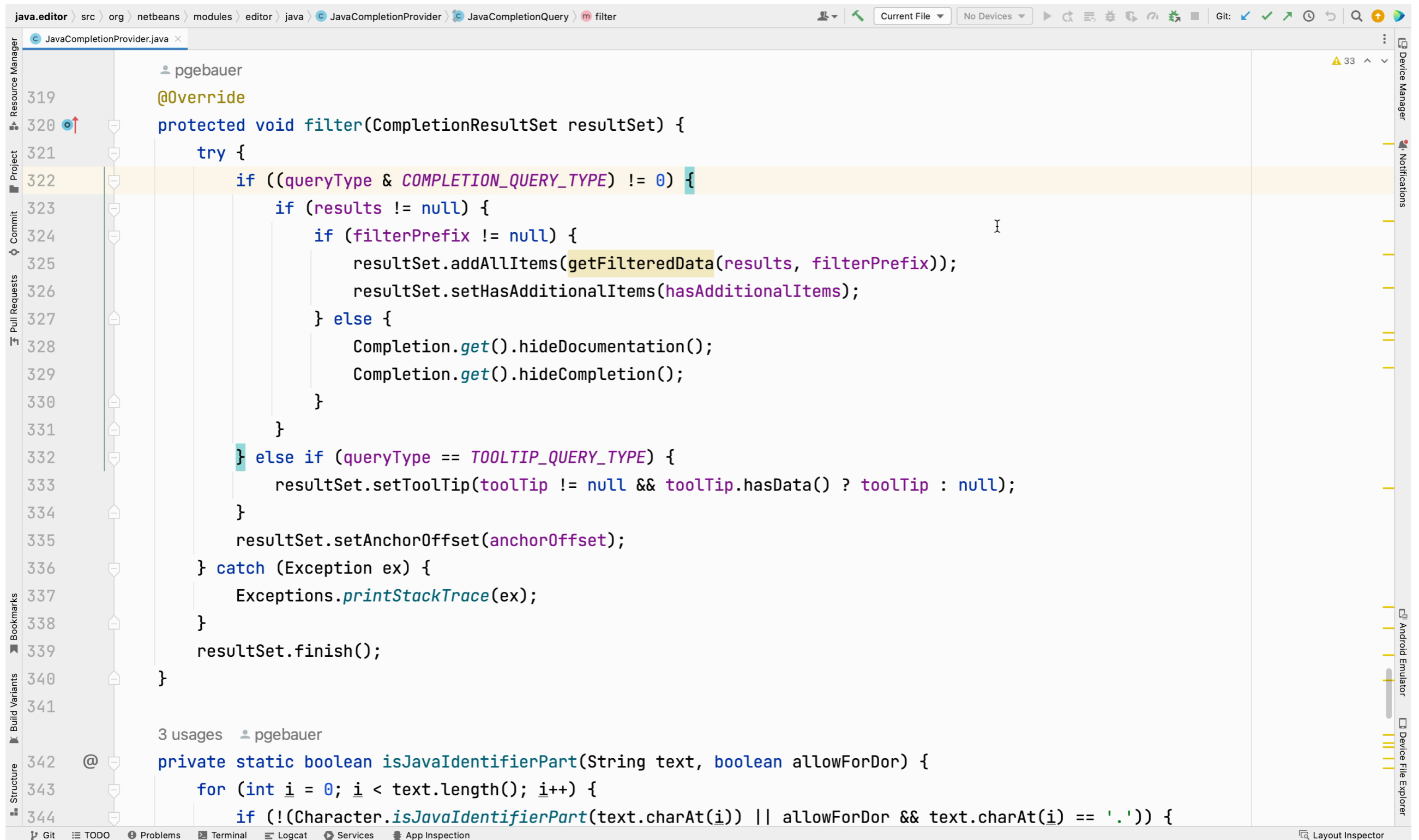
The screenshot shows an IDE window with the following components:

- Top Bar:** Shows the current file path: `java.editor > src > org > netbeans > modules > editor > java > JavaCompletionProvider > JavaCompletionQuery > filter`. It also includes a search bar and various utility icons.
- Left Sidebar:** Contains a vertical toolbar with icons for Resource Manager, Project, Commit, Pull Requests, Bookmarks, Build Variants, and Structure.
- Main Editor:** Displays the source code for `JavaCompletionProvider.java`. The code is as follows:

```
319     @Override
320     protected void filter(CompletionResultSet resultSet) {
321         try {
322             if ((queryType & COMPLETION_QUERY_TYPE) != 0) {
323                 if (results != null) {
324                     if (filterPrefix != null) {
325                         resultSet.addAllItems(getFilteredData(results, filterPrefix));
326                         resultSet.setHasAdditionalItems(hasAdditionalItems);
327                     } else {
328                         Completion.get().hideDocumentation();
329                         Completion.get().hideCompletion();
330                     }
331                 }
332             } else if (queryType == TOOLTIP_QUERY_TYPE) {
333                 resultSet.setTooltip(toolTip != null && toolTip.hasData() ? toolTip : null);
334             }
335             resultSet.setAnchorOffset(anchorOffset);
336         } catch (Exception ex) {
337             Exceptions.printStackTrace(ex);
338         }
339         resultSet.finish();
340     }
341
342     @private static boolean isJavaIdentifierPart(String text, boolean allowForDor) {
343         for (int i = 0; i < text.length(); i++) {
344             if (!(Character.isJavaIdentifierPart(text.charAt(i)) || allowForDor && text.charAt(i) == '.')) {
```

The line `if ((queryType & COMPLETION_QUERY_TYPE) != 0) {` at line 322 is highlighted in yellow. The cursor is positioned at the end of the line.
- Right Sidebar:** Contains icons for Device Manager, Notifications, Android Emulator, and Device File Explorer.
- Bottom Bar:** Includes icons for Git, TODO, Problems, Terminal, Logcat, Services, and App Inspection.

Preserving Context in IDEs



```
java.editor > src > org > netbeans > modules > editor > java > JavaCompletionProvider > JavaCompletionQuery > filter
JavaCompletionProvider.java x
pgebauer
319 @Override
320 protected void filter(CompletionResultSet resultSet) {
321     try {
322         if ((queryType & COMPLETION_QUERY_TYPE) != 0) {
323             if (results != null) {
324                 if (filterPrefix != null) {
325                     resultSet.addAllItems(getFilteredData(results, filterPrefix));
326                     resultSet.setHasAdditionalItems(hasAdditionalItems);
327                 } else {
328                     Completion.get().hideDocumentation();
329                     Completion.get().hideCompletion();
330                 }
331             }
332         } else if (queryType == TOOLTIP_QUERY_TYPE) {
333             resultSet.setTooltip(toolTip != null && toolTip.hasData() ? toolTip : null);
334         }
335         resultSet.setAnchorOffset(anchorOffset);
336     } catch (Exception ex) {
337         Exceptions.printStackTrace(ex);
338     }
339     resultSet.finish();
340 }
341
342 3 usages pgebauer
343 @ private static boolean isJavaIdentifierPart(String text, boolean allowForDor) {
344     for (int i = 0; i < text.length(); i++) {
345         if (!(Character.isJavaIdentifierPart(text.charAt(i)) || allowForDor && text.charAt(i) == '.')) {
```

Preserving Context in IDEs

The image displays an IDE interface with three main panels:

- Structure Panel (Left):** Shows the class hierarchy for `KeyPromoterStatistics`, listing methods such as `getState()`, `loadState()`, `registerAction()`, `registerShortcutUsed()`, `resetStatistic()`, `suppressItem()`, `get()`, `getStatisticItems()`, `getSuppressedItems()`, `isSuppressed()`, `unsuppressItem()`, `suppressItem()`, `STATISTIC`, `SUPPRESS`, `statistics`, `suppressed`, and `myChangeSupport`.
- Code Editor (Center):** Displays the source code for `KeyPromoterStatistics.java`. The code includes annotations like `@Override`, `@Transient`, and `@NotNull`. Key methods shown are `loadState()`, `registerAction()`, and `registerShortcutUsed()`. The code uses `XmlSerializerUtil.copyBean(stats, this);` and `statistics.putIfAbsent(action.getDescription(), new StatisticsItem(action));`.
- Project Panel (Right):** Shows the project structure for `IntelliJ-Key-Promoter-X`. The `src [main]` directory contains the `de.halirutan.keypromoterx` package, which includes the `statistic` sub-package. The `KeyPromoterStatistics` class is highlighted in the project view.
- Analyze Dataflow To (Bottom):** Shows a dataflow analysis for the `parameter action`. The analysis starts at line 87: `statistics.putIfAbsent(action.getDescription(), new StatisticsItem(action));` in `KeyPromoterStatistics.registerAction(KeyPromoterAction)`. It then traces back to line 85: `public void registerAction(KeyPromoterAction action)` in `KeyPromoterStatistics.registerAction(KeyPromoterAction)`, and further back to line 164: `statsService.registerAction(action);` in `KeyPromoter.showTip(KeyPromoterAction, ActionType)`. The analysis continues through `showTip()` and `beforeActionPerformed()` methods, eventually identifying the creation of `kpAction` at line 124: `kpAction = new KeyPromoterAction(action, event, KeyPromoterAction.ActionSource.TOOL_WINDOW_BUTTON);` in `KeyPromoter.beforeActionPerformed(AnAction, AnActionEvent)`.

Preserving Context in IDEs

Preserving Context in IDEs

Context is not preserved across views of content

Preserving Context in IDEs

Context is not preserved across views of content

Views of content are separated and spatially distant

Contributions

- **Formative study:** what questions do programmers frequently ask?
- **ReachHover:** a context-preserving interface for data-flow questions
- **User study:** ReachHover helps developers answer data-flow questions

Contributions

- **Formative study:** what questions do programmers frequently ask?

Programmers care about data-flow

Programmers care about data-flow

**Where does a value
come from?**

Programmers care about data-flow

**Where does a value
come from?**

**How is this object going
to be modified?**

Programmers care about data-flow

**Where does a value
come from?**

**How is this object going
to be modified?**

**Which parts of this object
are accessed?**

Programmers care about data-flow

**Where does a value
come from?**

**How is this object going
to be modified?**

**Which parts of this object
are accessed?**

**How is this value
used throughout the program?**

Programmers care about data-flow

10 difficult questions from the literature

- Refactoring
- Control-flow
- Data-flow
- ...

Our work: Survey of 72 programmers

- How often do you ask yourself this question?

Most frequent: data-flow questions

- “Where does this value come from?”
- “How is this value modified?”
- ...

Contributions

- **Formative study**: what questions do programmers frequently ask?
 - Programmers **frequently** ask **data-flow questions**

Contributions

- **Formative study**: what questions do programmers frequently ask?
 - Programmers **frequently** ask **data-flow questions**
- **ReachHover**: a context-preserving interface for data-flow questions

ReachHover Live Demo

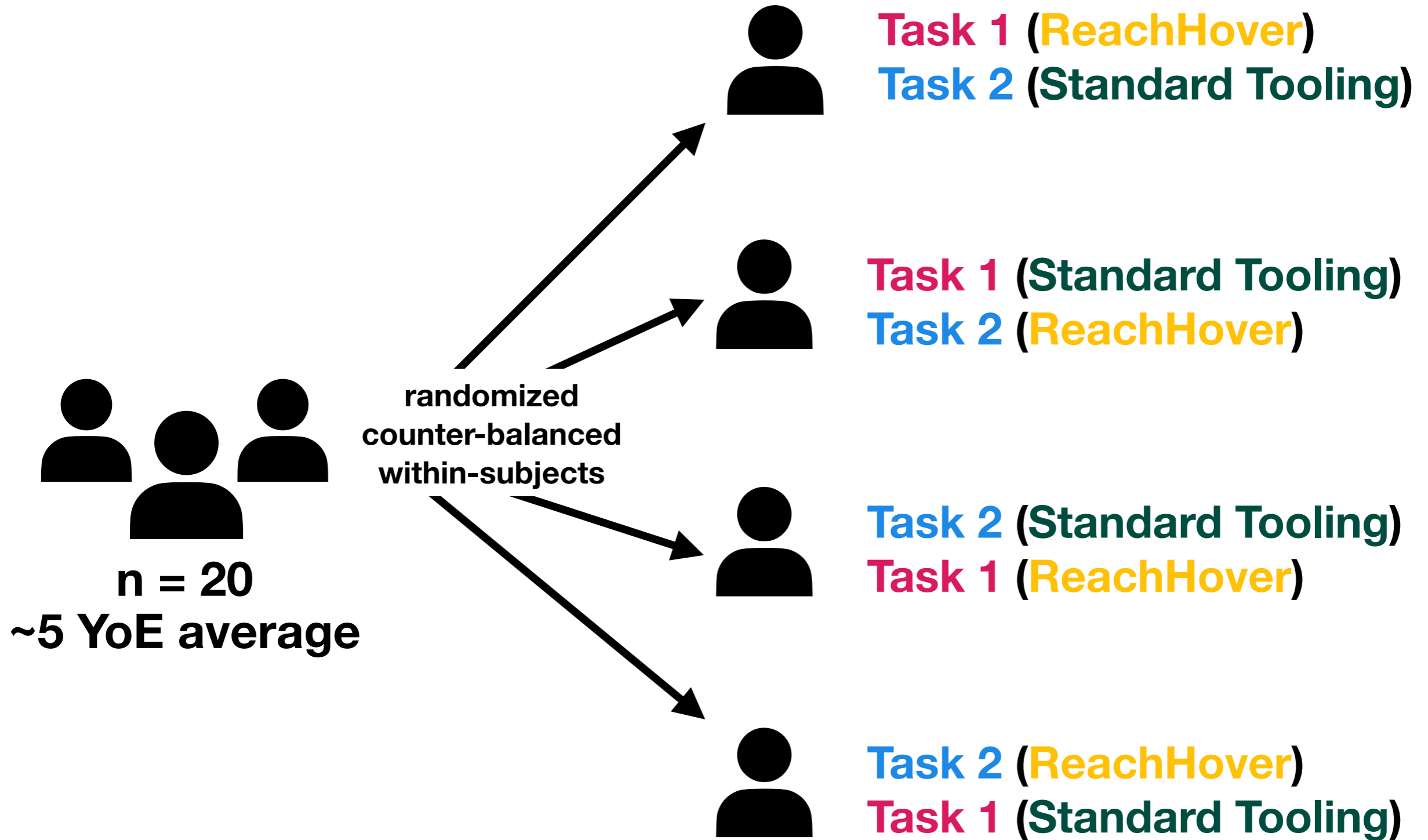
Contributions

- **Formative study**: what questions do programmers frequently ask?
 - Programmers **frequently** ask **data-flow questions**
- **ReachHover**: a context-preserving interface for data-flow questions
 - **Open-source plugin** for the **IntelliJ** Platform

Contributions

- **Formative study**: what questions do programmers frequently ask?
 - Programmers **frequently** ask **data-flow questions**
- **ReachHover**: a context-preserving interface for data-flow questions
 - **Open-source plugin** for the **IntelliJ** Platform
- **User study**: ReachHover helps developers answer data-flow questions

Evaluating ReachHover: User Study



Evaluating ReachHover: Experimental Design

- **Task 1:** Questions along a **backward data-flow trace**
 - **Q1** Locations where a method argument is instantiated
 - **Q2** Where is a method argument used?
- **Task 2:** Questions along a **forward data-flow trace**
 - **Q1** Locations where a local variable is checked for null
 - **Q2** Where is a local variable used?

Evaluating ReachHover: Conclusions

- Programmers can accurately answer data-flow-related questions with ReachHover

Task 1: Backward Data-flow

```
public void addBookmark(FileBookmarks fileBookmarks, BookmarkInfo bookmark) {  
    checkModDuringFire();  
    fileBookmarks.add(bookmark);  
    BookmarkChange change = getBookmarkChange(bookmark);  
    change.markAdded();  
    bookmark.getFileBookmarks().getProjectBookmarks().setModified(true);  
    structureChange = true;  
}
```

T1.Q1 In how many locations does a call to the **BookmarkInfo** constructor exist?

- ReachHover: 10/10
- IntelliJ: 6/10

Task 1: Backward Data-flow

```
public void addBookmark(FileBookmarks fileBookmarks, BookmarkInfo bookmark) {  
    checkModDuringFire();  
    fileBookmarks.add(bookmark);  
    BookmarkChange change = getBookmarkChange(bookmark);  
    change.markAdded();  
    bookmark.getFileBookmarks().getProjectBookmarks().setModified(true);  
    structureChange = true;  
}
```

T1.Q1 In how many locations does a call to the **BookmarkInfo** constructor exist?

- ReachHover: 10/10
- IntelliJ: 6/10

Task 1: Backward Data-flow

T1.Q2 Provide the names of the methods where the value of `bookmark` is used

	Correct Answers	Incorrect Answers	Precision
Unmodified IntelliJ	13	8	62%
ReachHover	14	2	88%

Task 2: Forward Data-flow

```
FindReplaceResult findReplaceResult = new FindReplaceResult(new int[]{-1, -1}, replacedString: "");  
findReplaceResult.setErrorMsg(NbBundle.getMessage(DocumentFinder.class, resName: "pattern-error-dialo  
return findReplaceResult;
```

T2.Q1 How many times was **findReplaceResult** explicitly checked for a null value?

- ReachHover: 7/10
- IntelliJ: 7/10

Task 2: Forward Data-flow

```
· FindReplaceResult findReplaceResult = new FindReplaceResult(new int[]{-1, -1}, replacedString: "");  
· findReplaceResult.setErrorMsg(NoBundle.getMessage(DocumentFinder.class, resName: "pattern-error-dialo  
· return findReplaceResult;
```

T2.Q1 How many times was **findReplaceResult** explicitly checked for a null value?

- ReachHover: 7/10
- IntelliJ: 7/10

Task 2: Forward Data-flow

T2.Q2 Provide the names of the methods where the value of `findReplaceResult` is used

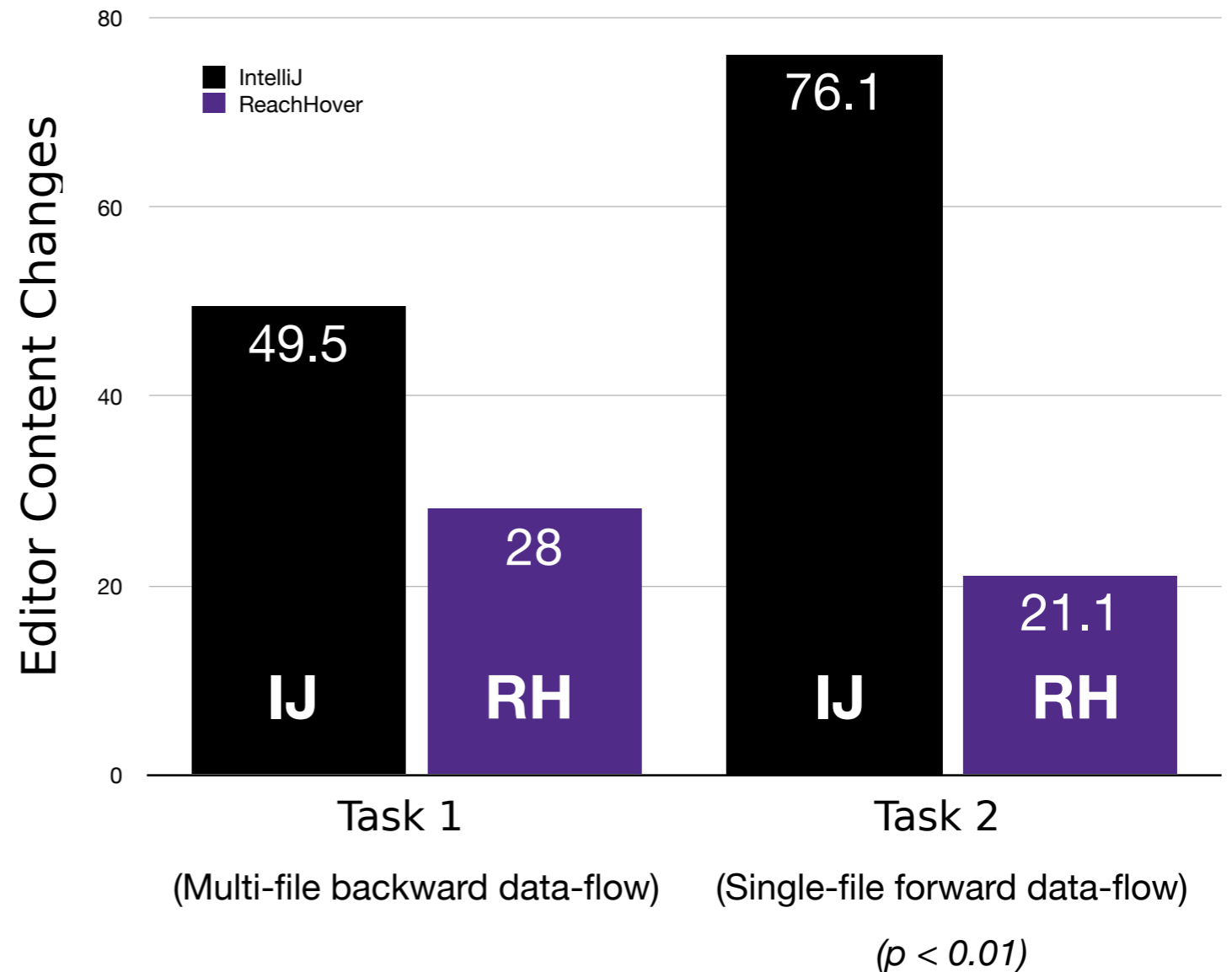
	Correct Answers	Incorrect Answers	Precision
Unmodified IntelliJ	22	10	69%
ReachHover	24	3	80%

Evaluating ReachHover: Conclusions

- Programmers can accurately answer data-flow-related questions with ReachHover
- ReachHover minimizes context-switching actions.

Evaluating ReachHover: Context

- Editor content changes as a proxy for context switching
- Lower average of editor content changes using ReachHover



(Lower is better)

Evaluating ReachHover: Conclusions

- Programmers accurately answer data-flow questions
- ReachHover reduces context-switching
- Users prefer context-preserving interface

Evaluating ReachHover: Qualitative Results

“Felt **less cluttered** and made the task easier”

“*Was **easier to use** (maybe because I **didn't lose the context**)”*

“*I preferred that I could **look at the matches** in the same modal window **without affecting the open file**”*

Related Work: Alternative Interfaces

Code Bubbles

- An infinitely-zoomable 2D canvas

Related Work: Code Bubbles

Code Bubbles - 0.258 @ <http://www.cs.brown.edu/people/spr/bubbles/>

Create Task...

```
org>jit>sp>jedit>buffer>UndoManager>redo(...)
//{{{ redo() method
public int redo()
{
    if(insideCompoundEdit())
        throw new
            InternalError("Unbalanced begin/endCompoundEdit()");

    if(redosFirst == null)
        return -1;
    else
    {
        reviseUndoId();
        undoCount++;

        int caret = redosFirst.redo();
        undosLast = redosFirst;
        if(undosFirst == null)
            undosFirst = undosLast;
        redosFirst = redosFirst.next;
        return caret;
    }
} //}}}
```

```
returns the current edit pane's buffer.
+ @return the current edit pane's buffer.
*/
public Buffer getBuffer()
{
    if(editPane == null)
        return null;
    else
        return editPane.getBuffer();
} //}}}
```

```
/**
 * Returns the fold level of the specified line
 * @param buffer The buffer in question
 * @param lineIndex The line index
 * @param seg A segment the fold handler can
 * text from the buffer, if necessary
 * @return The fold level of the specified line
 * @since jEdit 4.0pre1
 */
public int getFoldLevel(JEditBuffer buffer,
    int lineIndex, Segment seg)
{
    if(lineIndex == 0)
        return 0;
    else
    {
        int foldLevel =
            buffer.getFoldLevel(lineIndex - 1);
        buffer.getLineText(lineIndex - 1,
            seg);
    }
}
```

```
org>jit>sp>jedit>buffer>ContentManager>getLength(...)
//{{{ getLength() method
public final int getLength()
{
    return length;
} //}}}
```

```
org>jit>sp>jedit>buffer>JEditBuffer>createPosition(...)
//{{{ createPosition() method
/**
 * Creates a floating position.
 * @param offset The offset
 */
public Position createPosition(int offset)
{
    try
    {
        readLock();

        if(offset < 0 || offset > contentMgr.getLength())
        {
            throw new
                ArrayIndexOutOfBoundsException(offset);
        }

        return positionMgr.createPosition(offset);
    }
    finally
    {
        readUnlock();
    }
} //}}}
```

```
org>jit>sp>jedit>buffer>IndentFoldHandler>getFoldLevel(...)
//{{{ getFoldLevel() method
/**
 * Returns the fold level of the specified line. For a whitespace-only
 * line, returns the fold level of the next non-whitespace line, or
 * the level of the previous line if no non-whitespace line follows or if
 * the level of the previous line is higher.
 * @param buffer The buffer in question
 * @param lineIndex The line index
 * @param seg A segment the fold handler can use to obtain any
 * text from the buffer, if necessary
 * @return The fold level of the specified line
 * @since jEdit 4.0pre1
 */
public int getFoldLevel(JEditBuffer buffer, int lineIndex, Segment seg)
{
    int tabSize = buffer.getTabSize();
    // Look for first non-whitespace line starting at lineIndex
    int prevLevel = 0;
    for (int index = lineIndex; index < buffer.getLineCount(); index++)
    {
        buffer.getLineText(index, seg);
        int whitespace = getLeadingWhitespaceWidth(seg, tabSize);
    }
}
```

```
org>jit>sp>jedit>textarea>TextArea>getSelection(...)
//{{{ getSelection() methods
/**
 * Returns the current selection.
 * @since jEdit 3.2pre1
 */
public Selection[] getSelection()
{
    return selectionManager.getSelection();
}
```

```
org>jit>sp>jedit>textarea>TextArea>getSelectionAt(...)
//{{{ getSelectionAtOffset() method
/**
 * Returns the selection containing the
 * if there is no selection at that offset
 * @param offset The offset
 * @since jEdit 3.2pre1
 */
public Selection getSelectionAtOffset(int offset)
{
    return selectionManager.getSelectionAtOffset(offset);
} //}}}
```

jedit

- Abbrevs
- AbstractOptionPane
- ActionContext
- ActionListHandler
- ActionSet
- Autosave
- BeanShell
- BeanShellAction
- BeanShellFacade
- Buffer
- BufferHistory
- Debug
- EBComponent
- EBMessage
- EBPlugin
- EditAction
- EditBus
- EditPane
- EditPlugin
- EditServer
- GUIUtilities
- IPropertyManager
- JARClassLoader
- JEditAbstractEditAction
- JEditActionContext
- JEditActionSet
- JEditBeanShellAction
- JEditKillRing
- JEditMode
- JEditRegisterSaver
- JEditRegistersListener
- Macros
- Marker
- MiscUtilities
- Mode
- ModeCatalogHandler
- OperatingSystem
- OptionGroup
- OptionPane
- PerspectiveManager
- PluginJAR

Related Work: Alternative Interfaces

Code Bubbles

- An infinitely-zoomable 2D canvas

Drastic re-design of a standard IDE

CodeRibbon

- A tiled “ribbon” of code views

Related Work: Code Ribbon

The screenshot displays an IDE interface with a project explorer on the left and three code editors. The project explorer shows the structure of the 'IntelliJ-Key-Promoter-X' project, including source files like 'KeyPromoterToolWindowPanel.form', 'KeyPromoterConfiguration.java', and 'KeyPromoterSettings.java'.

The top editor shows the XML code for 'KeyPromoterToolWindowPanel.form', including a grid layout and a splitpane component.

```
36 </component>
37 <hspacer id="beba2">
38 <constraints>
39 <grid row="0" column="1" row-span="1" col-span="1" vsize-policy="1" hsize-policy="1" />
40 </constraints>
41 </hspace>
42 </children>
43 </grid>
44 <splitpane id="a25e0" binding="splitPane">
45 <constraints>
46 <grid row="0" column="0" row-span="2" col-span="1" vsize-policy="7" hsize-policy="7" />
47 <preferred-size width="200" height="200" />
48 </grid>
49 </constraints>
50 <properties>
51 <orientation value="0" />
52 </properties>
53 <border type="empty">
54 <color color="-386270" />
55 </border>
56 <children>
57 <scrollpane id="40379">
58 <constraints>
59 <splitpane position="left" />
60 </constraints>

```

The top-right editor shows the Java code for 'KeyPromoterConfiguration.java', including a copyright notice and package declarations.

```
1 /*
2  * Copyright (c) 2019 Patrick Scheibe, Dmitry Kashin, Athiele.
3  *
4  * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the
5  * following conditions are met:
6  * 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the
7  * accompanying License Agreement must be reproduced in full.
8  * 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the
9  * accompanying License Agreement in the documentation and/or other materials provided with the distribution.
10 *
11 * 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or
12 * promote products derived from this software without specific prior written permission.
13 *
14 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED
15 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
16 * PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY
17 * DIRECT, INDIRECT, INCIDENTAL, DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
18 * SERVICES; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, TORT
19 * OR OTHERWISE ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
20 * DAMAGE.
21 */
22
23 package de.halirutan.keypromoterx;
24
25 import com.intellij.openapi.application.ApplicationManager;

```

The bottom editor shows the Java code for 'KeyPromoterSettings.java', including public fields and comments.

```
56 public boolean editorPopupEnabled = true;
57 /**
58  * Whether popup enabled or disabled on all buttons with mnemonics clicks.
59  */
60 public boolean allButtonsEnabled = true;
61
62 /**
63  * If the user clicks a button, usually each time a notification is shown (setting 1). If the user clicks a
64  * second click will trigger a notification, and so on.
65  */
66 public int showTipsClickCount = 1;
67
68 public int proposeToCreateShortcutCount = 3;
69
70 /**
71  * Whether to show notification when the IDE is in Presentation Mode
72  */
73 public boolean disabledInPresentationMode = false;
74
75 /**
76  * Whether to show notification when the IDE is in Distraction Free Mode
77  */
78 public boolean disabledInDistractionFreeMode = false;
79
80

```

The bottom-right editor shows the XML code for 'KeyPromoterToolWindowPanel.form', including a form layout and a button component.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uideigner/form/" version="1" bind-to-class="de.halirutan.keypromoterx.KeyPromoterToolWindowPanel" />
3 <grid id="27dc6" binding="panel" layout-manager="GridLayoutManager" row-count="3" column-count="1" />
4 <margin top="5" left="5" bottom="5" right="5" />
5 <constraints>
6 <xy x="20" y="20" width="422" height="246" />
7 </constraints>
8 <properties />
9 <border type="none" />
10 <children>
11 <grid id="2b6d2" layout-manager="GridLayoutManager" row-count="2" column-count="2" />
12 <margin top="0" left="0" bottom="0" right="0" />
13 <constraints>
14 <grid row="2" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" />
15 </constraints>
16 <properties />
17 <border type="none" title="Actions" />
18 <children>
19 <component id="48222" class="javax.swing.JButton" binding="resetStatisticsButton" />
20 <constraints>
21 <grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" />
22 </constraints>
23 <properties>
24 <label value="Reset Statistics" />
25 <text value="&Reset Statistics" />

```


Related Work: Code Ribbon

The screenshot displays an IDE interface with a project tree on the left and three code editors. The project tree shows the structure of the 'IntelliJ-Key-Promoter-X' project, including source files like 'KeyPromoterConfiguration.java' and 'KeyPromoterSettings.java'. The top editor shows the XML code for 'KeyPromoterToolWindowPanel.form', detailing the layout of a window panel with a grid and a split pane. The middle editor shows the Java code for 'KeyPromoterSettings.java', defining various boolean and integer settings. The bottom editor shows the XML code for 'KeyPromoterToolWindowPanel.form', detailing the layout of a tool window panel with a grid and a button.

```
36 </component>
37 <hspacer id="beba2">
38 <constraints>
39 <grid row="0" column="1" row-span="1" col-span="1" vsize-policy="1" hsize-policy="1" />
40 </constraints>
41 </hspacer>
42 </children>
43 </grid>
44 <splitpane id="a25e0" binding="splitPane">
45 <constraints>
46 <grid row="0" column="0" row-span="2" col-span="1" vsize-policy="7" hsize-policy="7" />
47 <preferred-size width="200" height="200" />
48 </grid>
49 </constraints>
50 <properties>
51 <orientation value="0" />
52 </properties>
53 <border type="empty">
54 <color color="-386270" />
55 </border>
56 <children>
57 <scrollpane id="40379">
58 <constraints>
59 <splitpane position="left" />
60 </constraints>
61 </scrollpane>
62 </children>
63 </splitpane>
64 </constraints>
65 </component>
66 </form>
67 </xml>
```

```
1 /*
2  * Copyright (c) 2019 Patrick Scheibe, Dmitry Kashin, Athiele.
3  *
4  * Redistribution and use in source and binary forms, with or without modification, are permitted
5  * provided that the following conditions are met:
6  * 1. Redistributions of source code must retain the above copyright notice, this list of
7  *    conditions and the following disclaimer.
8  * 2. Redistributions in binary form must reproduce the above copyright notice, this list
9  *    of conditions and the following disclaimer in the documentation and/or other materials
10 *    provided with the distribution.
11 * 3. Neither the name of the copyright holder nor the names of its contributors may be used
12 *    to endorse or promote products derived from this software without specific prior written
13 *    permission.
14 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS
15 *    OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
16 *    MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL
17 *    THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
18 *    OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
19 *    GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
20 *    CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
21 *    (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
22 *    SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
23 */
24
25 package de.halirutan.keypromoterx;
26
27 import com.intellij.openapi.application.ApplicationManager;
```

```
56 public boolean editorPopupEnabled = true;
57 /**
58  * Whether popup enabled or disabled on all buttons with mnemonics clicks.
59  */
60 public boolean allButtonsEnabled = true;
61
62 /**
63  * If the user clicks a button, usually each time a notification is shown (setting 1). If
64  * second click will trigger a notification, and so on.
65  */
66 public int showTipsClickCount = 1;
67
68 public int proposeToCreateShortcutCount = 3;
69
70 /**
71  * Whether to show notification when the IDE is in Presentation Mode
72  */
73 public boolean disabledInPresentationMode = false;
74
75 /**
76  * Whether to show notification when the IDE is in Distraction Free Mode
77  */
78 public boolean disabledInDistractionFreeMode = false;
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uidesigner/form/" version="1" bind-to-class="de.halirutan.keypromoterx.KeyPromoterToolWindowPanel" />
3 <grid id="27dc6" binding="panel" layout-manager="GridLayoutManager" row-count="3" column-count="1" />
4 <margin top="5" left="5" bottom="5" right="5" />
5 <constraints>
6 <xy x="20" y="20" width="422" height="246" />
7 </constraints>
8 <properties />
9 <border type="none" />
10 <children>
11 <grid id="2b6d2" layout-manager="GridLayoutManager" row-count="2" column-count="2" />
12 <margin top="0" left="0" bottom="0" right="0" />
13 <constraints>
14 <grid row="2" column="0" row-span="1" col-span="1" vsize-policy="3" hsize-policy="3" />
15 </constraints>
16 <properties />
17 <border type="none" title="Actions" />
18 <children>
19 <component id="48222" class="javax.swing.JButton" binding="resetStatisticsButton" />
20 <constraints>
21 <grid row="0" column="0" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" />
22 </constraints>
23 <properties>
24 <label value="Reset Statistics" />
25 <text value="&Reset Statistics" />
```

Preserving Context in IDEs

Context is not preserved across views of content

Views of content are separated and spatially distant

Contributions

- **Formative study**: what questions do programmers frequently ask?
 - Programmers **frequently** ask **data-flow questions**

Contributions

- **Formative study**: what questions do programmers frequently ask?
 - Programmers **frequently** ask **data-flow questions**
- **ReachHover**: a context-preserving interface for data-flow questions
 - **Open-source plugin** for the **IntelliJ** Platform

Contributions

- **Formative study**: what questions do programmers frequently ask?
 - Programmers **frequently** ask **data-flow questions**
- **ReachHover**: a context-preserving interface for data-flow questions
 - **Open-source plugin** for the **IntelliJ** Platform
- **User study**: ReachHover helps developers answer data-flow questions

ReachHover
source code

