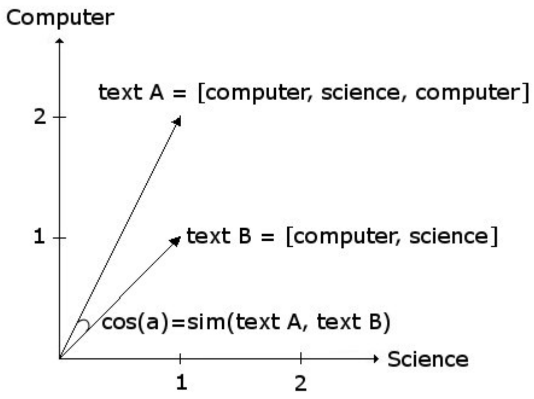
A Case Study for Evaluating Feature Location Techniques

Meghan Revelle

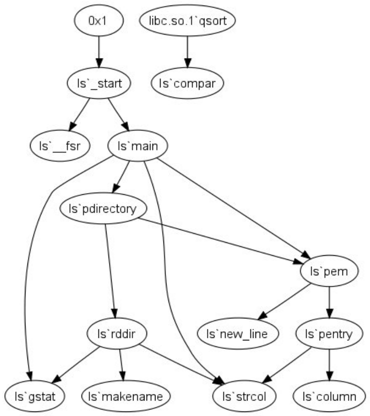
Nov. 20, 2008

How can we locate features?

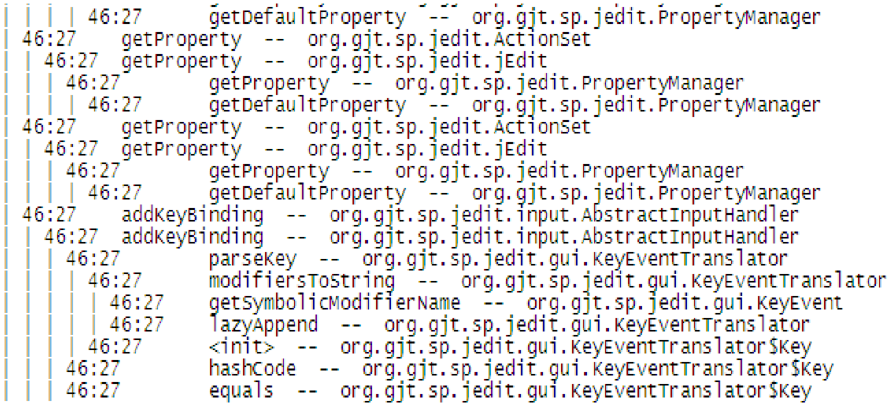
- Textual similarity


Computer
text A = [computer, science, computer]
text B = [computer, science]
cos(a)=sim(text A, text B)
Science
2
1
1
2

- Statically



- Dynamically



```
| | | | 46:27      getDefaultProperty  --  org.gjt.sp.jedit.PropertyManager
| 46:27    getProperty  --  org.gjt.sp.jedit.ActionSet
| | 46:27  getProperty  --  org.gjt.sp.jedit.jEdit
| | | 46:27        getProperty  --  org.gjt.sp.jedit.PropertyManager
| | | | 46:27      getDefaultProperty  --  org.gjt.sp.jedit.PropertyManager
| 46:27    getProperty  --  org.gjt.sp.jedit.ActionSet
| | 46:27  getProperty  --  org.gjt.sp.jedit.jEdit
| | | 46:27        getProperty  --  org.gjt.sp.jedit.PropertyManager
| | | | 46:27      getDefaultProperty  --  org.gjt.sp.jedit.PropertyManager
| 46:27    addKeyBinding  --  org.gjt.sp.jedit.input.AbstractInputHandler
| | 46:27  addKeyBinding  --  org.gjt.sp.jedit.input.AbstractInputHandler
| | | 46:27        parseKey  --  org.gjt.sp.jedit.gui.KeyEventTranslator
| | | | 46:27      modifiersToString  --  org.gjt.sp.jedit.gui.KeyEventTranslator
| | | | | 46:27    getSymbolicModifierName  --  org.gjt.sp.jedit.gui.KeyEvent
| | | | | 46:27    lazyAppend  --  org.gjt.sp.jedit.gui.KeyEventTranslator
| | | | 46:27      <init>  --  org.gjt.sp.jedit.gui.KeyEventTranslator$Key
| | | 46:27        hashCode  --  org.gjt.sp.jedit.gui.KeyEventTranslator$Key
| | | 46:27        equals  --  org.gjt.sp.jedit.gui.KeyEventTranslator$Key
```

- Combinations of the three
  - IR + static
  - IR + dynamic
  - Static + dynamic
  - IR + dynamic + static
  - …

Evaluation Approaches

- Precision and recall
- Position of first relevant method on ranked list
- Compare with methods in a patch
- Programmer agreement
  - This is where you come into the picture

Example

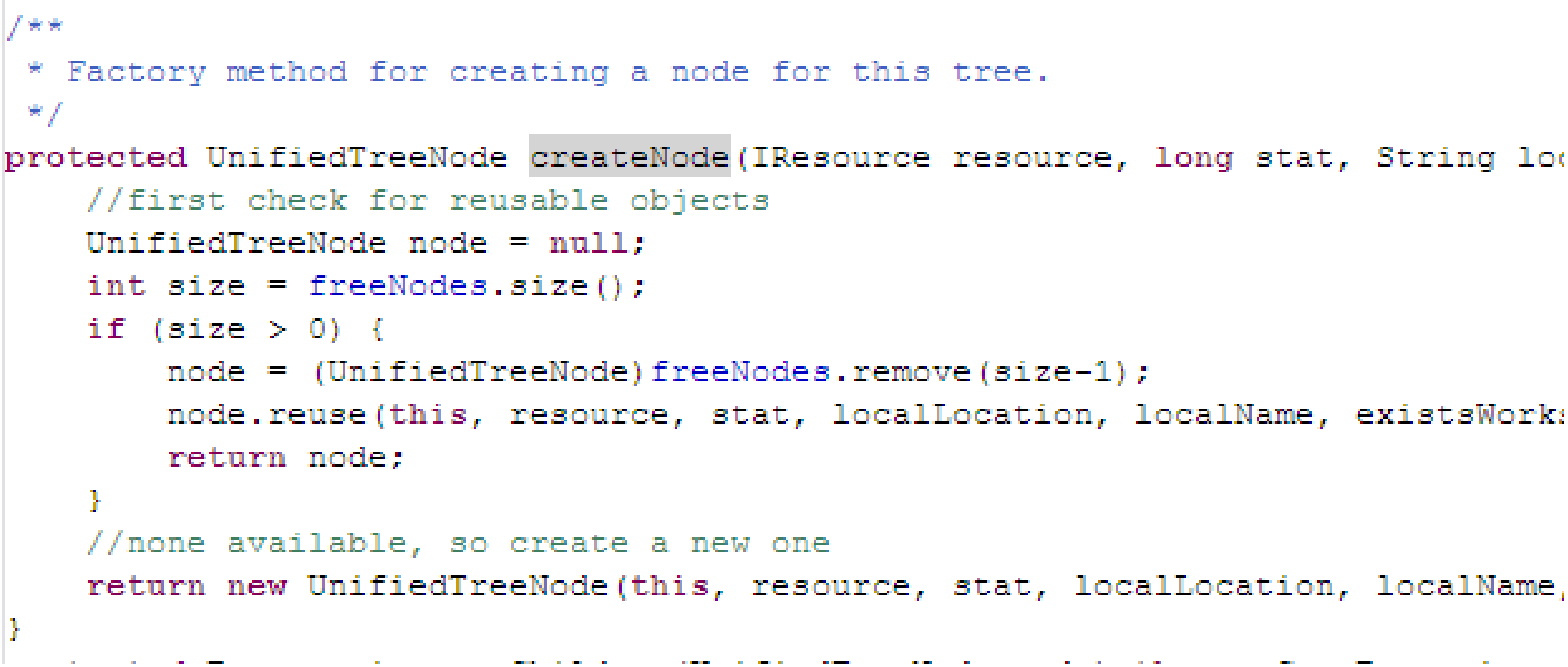
- System
  - Eclipse

- Feature
  - Add files and folders to UnifiedTree

Classifying Methods

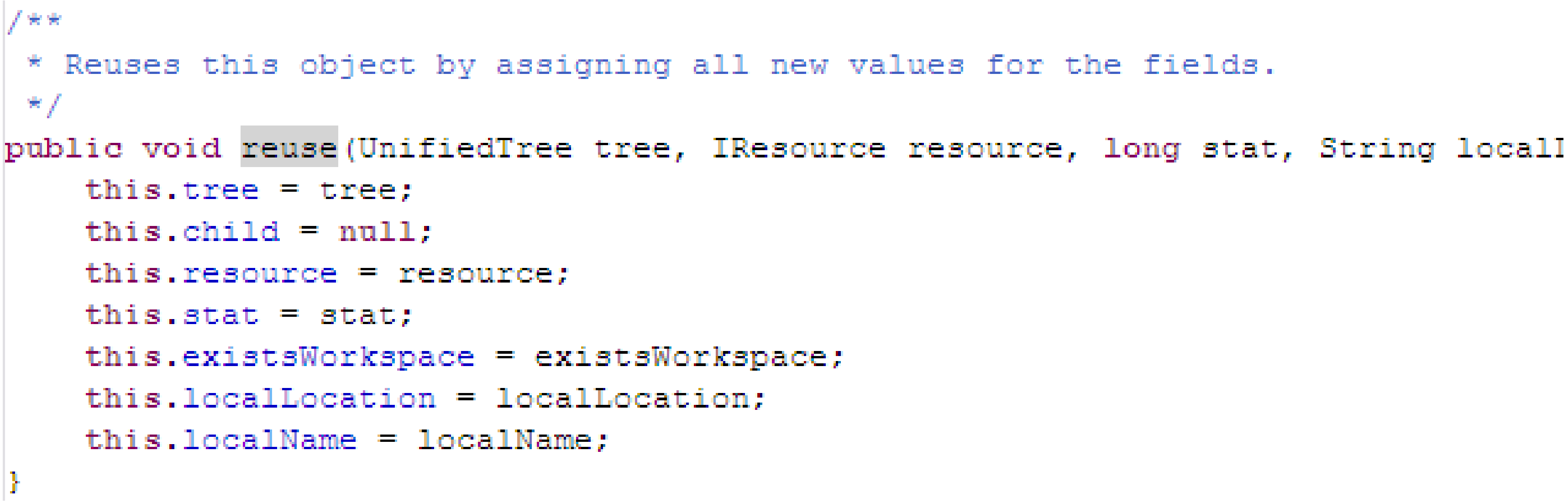
List 1	Relevant	Somewhat Relevant	Not Relevant
org.eclipse.core.internal.localstore.UnifiedTree.addChildrenFromFileSystem	X		
org.eclipse.core.internal.dtree.AbstractDataTreeNode.childAtOrNull			X
org.eclipse.core.internal.localstore.UnifiedTreeNode.reuse		X	
org.eclipse.core.internal.localstore.UnifiedTree.createNode	X		
org.eclipse.core.internal.localstore.UnifiedTreeNode.getLocalLocation		X	
org.eclipse.core.internal.localstore.RefreshLocalVisitor.visit	X		
org.eclipse.core.internal.localstore.RefreshLocalVisitor.synchronizeExistence		X	
org.eclipse.core.internal.localstore.UnifiedTree.addRootToQueue	X		
org.eclipse.core.internal.dtree.DeltaDataTree.getChildNodes			X
org.eclipse.core.internal.localstore.UnifiedTreeNode.existsInFileSystem	X		

Is this method relevant?



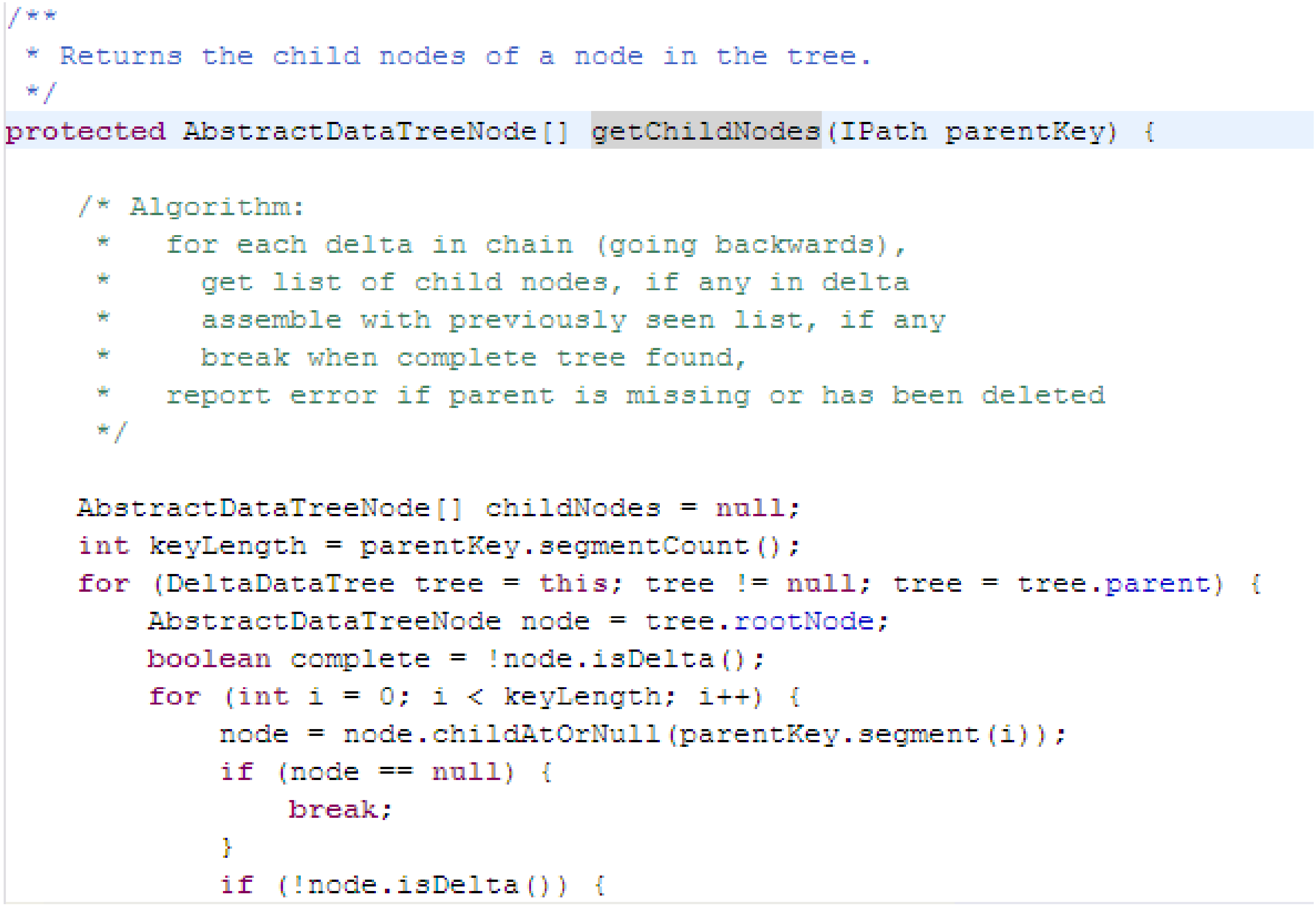
```
/**
 * Factory method for creating a node for this tree.
 */
protected UnifiedTreeNode createNode(IResource resource, long stat, String lo
    //first check for reusable objects
    UnifiedTreeNode node = null;
    int size = freeNodes.size();
    if (size > 0) {
        node = (UnifiedTreeNode)freeNodes.remove(size-1);
        node.reuse(this, resource, stat, localLocation, localName, existsWork
        return node;
    }
    //none available, so create a new one
    return new UnifiedTreeNode(this, resource, stat, localLocation, localName,
}
```

Is this method relevant?



```
/**
 * Reuses this object by assigning all new values for the fields.
 */
public void reuse(UnifiedTree tree, IResource resource, long stat, String localL
    this.tree = tree;
    this.child = null;
    this.resource = resource;
    this.stat = stat;
    this.existsWorkspace = existsWorkspace;
    this.localLocation = localLocation;
    this.localName = localName;
}
```

Is this method relevant?



```
/**
 * Returns the child nodes of a node in the tree.
 */
protected AbstractDataTreeNode[] getChildNodes(IPath parentKey) {

    /* Algorithm:
     *   for each delta in chain (going backwards),
     *     get list of child nodes, if any in delta
     *     assemble with previously seen list, if any
     *     break when complete tree found,
     *   report error if parent is missing or has been deleted
     */

    AbstractDataTreeNode[] childNodes = null;
    int keyLength = parentKey.segmentCount();
    for (DeltaDataTree tree = this; tree != null; tree = tree.parent) {
        AbstractDataTreeNode node = tree.rootNode;
        boolean complete = !node.isDelta();
        for (int i = 0; i < keyLength; i++) {
            node = node.childAtOrNull(parentKey.segment(i));
            if (node == null) {
                break;
            }
            if (!node.isDelta()) {
```

We need your help

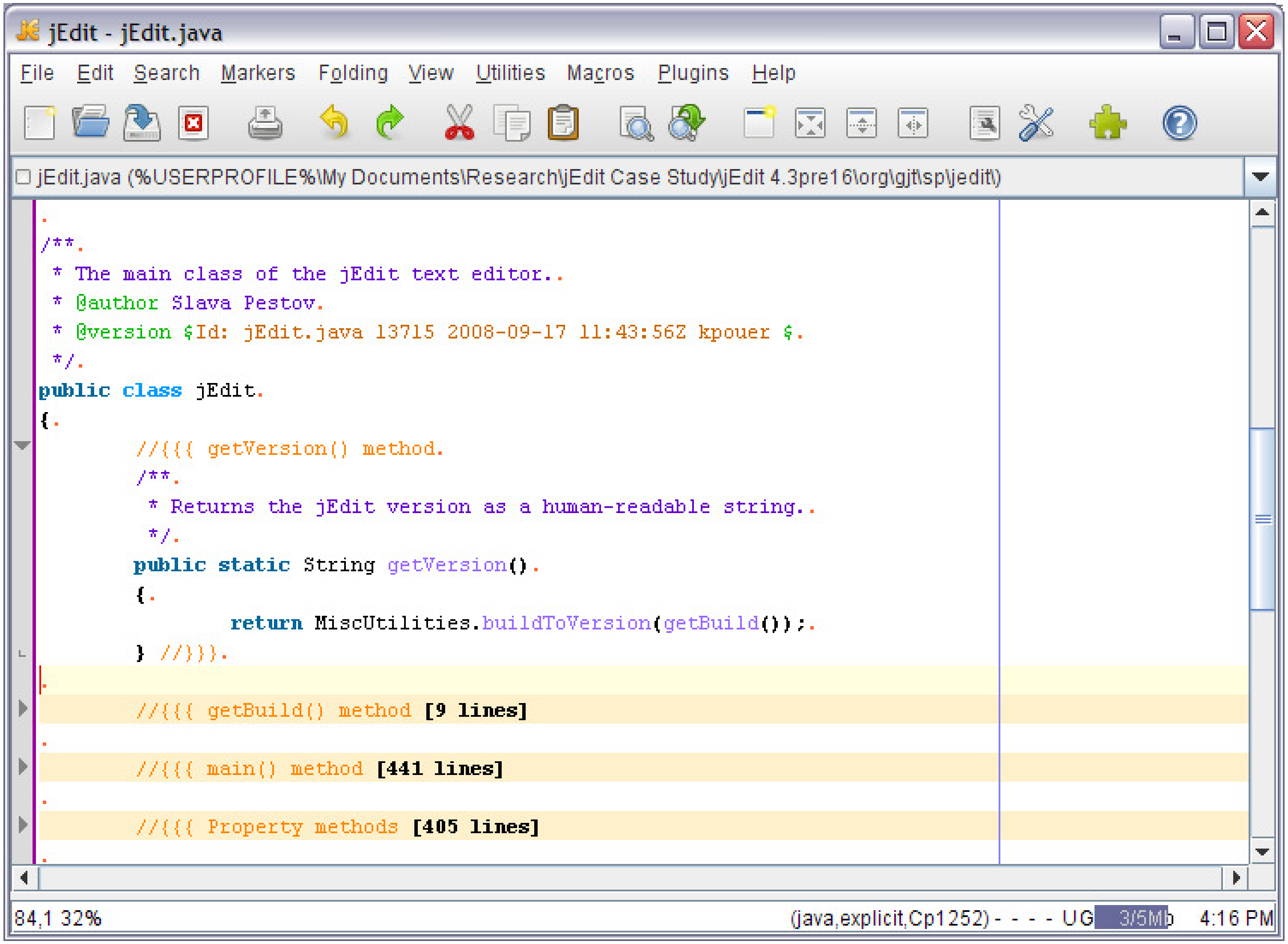
- Evaluate our feature location technique against others

- Classify methods from 10 lists

- System
  - jEdit

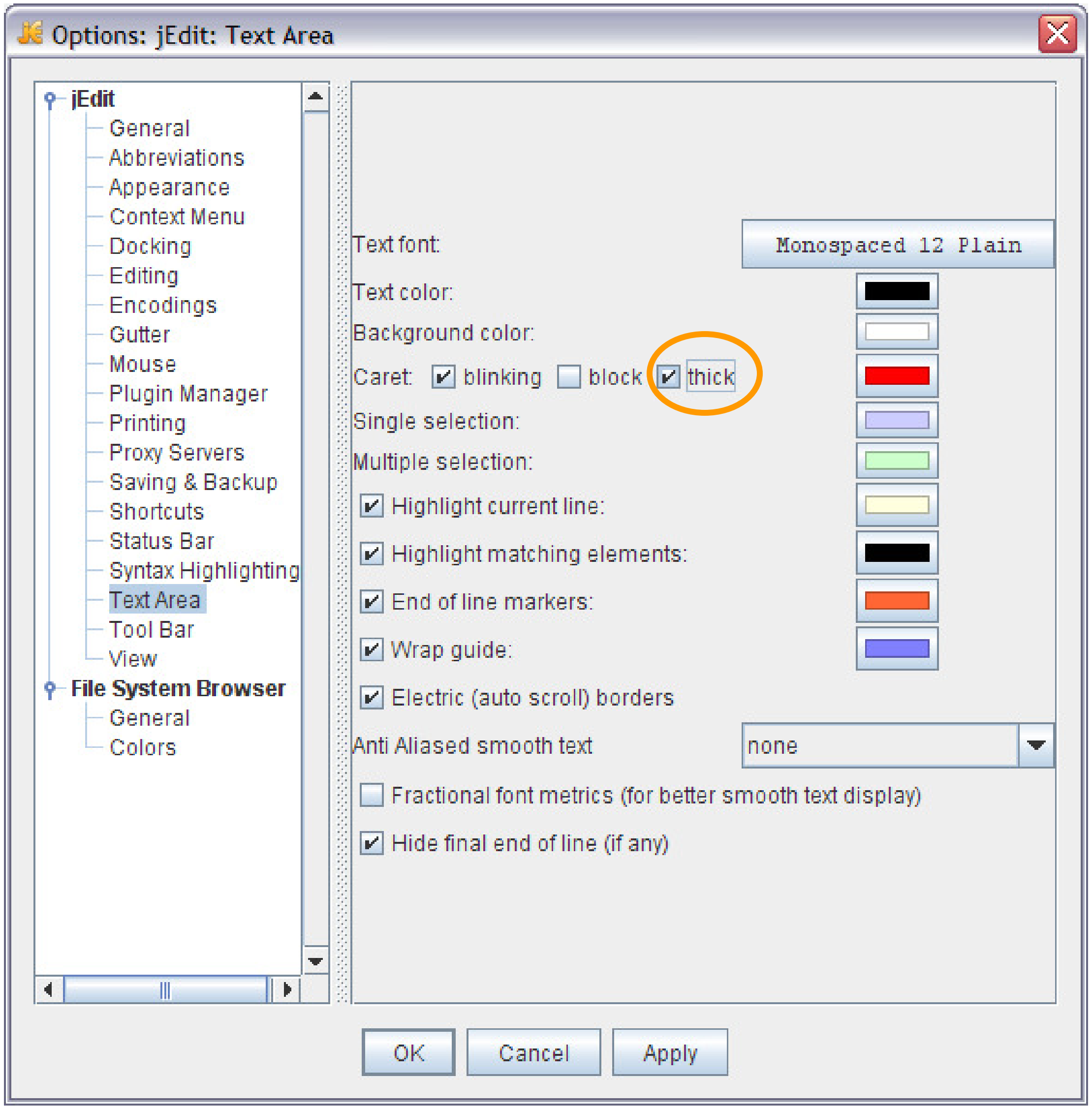
- Feature
  - Configurable thick cursor option

jEdit in action…


jEdit - jEdit.java
File Edit Search Markers Folding View Utilities Macros Plugins Help
jEdit.java (%USERPROFILE%\My Documents\Research\jEdit Case Study\jEdit 4.3pre16\org\gjt\sp\jedit\)
84,1 32%
(java,explicit,Cp1252) - - - - UG 3/5Mb 4:16 PM

```
.
/**.
 * The main class of the jEdit text editor..
 * @author Slava Pestov.
 * @version $Id: jEdit.java 13715 2008-09-17 11:43:56Z kpouer $.
 */.
public class jEdit.
{.
        //{{{ getVersion() method.
        /**.
         * Returns the jEdit version as a human-readable string..
         */.
        public static String getVersion().
        {.
                return MiscUtilities.buildToVersion(getBuild());.
        } //}}}.
.
        //{{{ getBuild() method [9 lines]
.
        //{{{ main() method [441 lines]
.
        //{{{ Property methods [405 lines]
.
```

Thick Caret Option



Text font: Monospaced 12 Plain

Text color: ⬛

Background color: ⬜

Caret: ☑ blinking ☐ block ☑ thick 🟥

Single selection: 🟪

Multiple selection: 🟩

☑ Highlight current line: 🟨

☑ Highlight matching elements: ⬛

☑ End of line markers: 🟧

☑ Wrap guide: 🟦

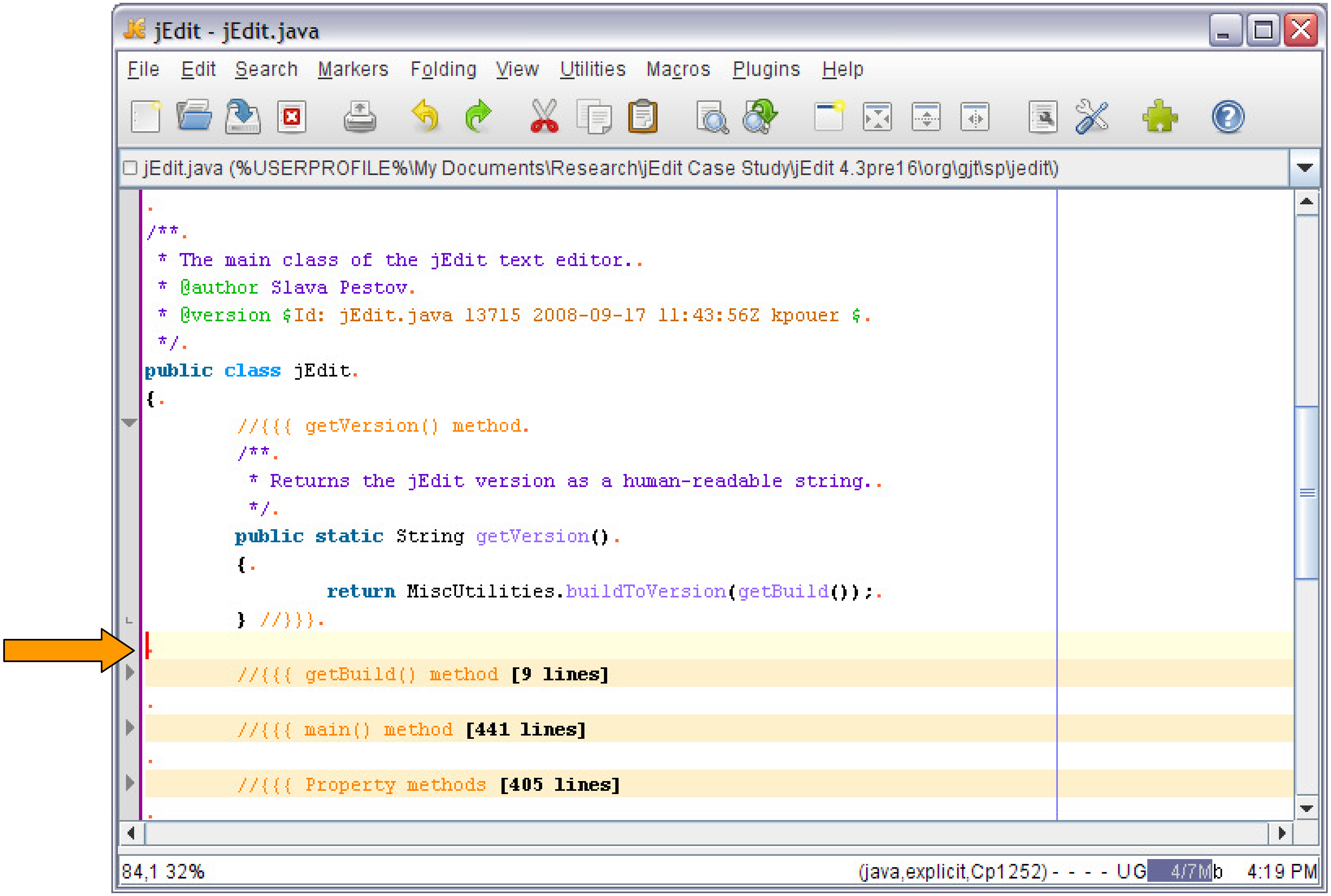
☑ Electric (auto scroll) borders

Anti Aliased smooth text none

☐ Fractional font metrics (for better smooth text display)

☑ Hide final end of line (if any)

Can you spot the difference?



jEdit - jEdit.java
File Edit Search Markers Folding View Utilities Macros Plugins Help
jEdit.java (%USERPROFILE%\My Documents\Research\jEdit Case Study\jEdit 4.3pre16\org\gjt\sp\jedit\)

```
.
/**.
 * The main class of the jEdit text editor..
 * @author Slava Pestov.
 * @version $Id: jEdit.java 13715 2008-09-17 11:43:56Z kpouer $.
 */.
public class jEdit.
{.
	//{{{ getVersion() method.
	/**.
	 * Returns the jEdit version as a human-readable string..
	 */.
	public static String getVersion().
	{.
		return MiscUtilities.buildToVersion(getBuild());.
	} //}}}.

	//{{{ getBuild() method [9 lines]
.
	//{{{ main() method [441 lines]
.
	//{{{ Property methods [405 lines]
.
```

84,1 32%
(java,explicit,Cp1252) - - - - UG 4/7Mb 4:19 PM

In Summary

- Task
  - Determine relevance of methods to a feature
- Time commitment
  - 1-2 hours
- Incentive
  - Extra points
  - Cookies and brownies