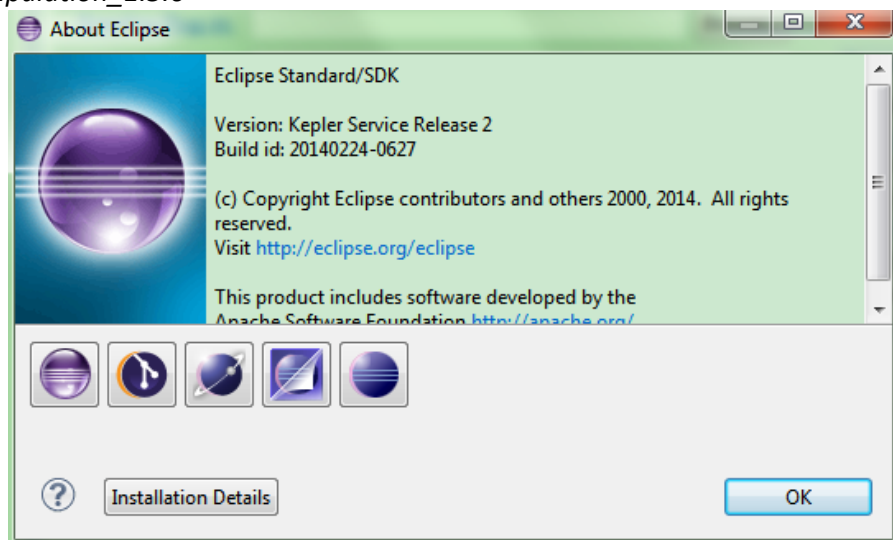# Rename Expander

1. Introduction

Rename Expander identifies renaming opportunities by expanding conducted renamings. Once a rename refactoring is conducted manually or with tool support, the proposed approach recommends to rename closely related software entities whose names are similar to that of the renamed entity. The rationale is that if an engineer makes a mistake in naming a software entity it is likely for her to make the same mistake in naming similar and closely related software entities. The main advantage of the proposed approach is that it does not involve difficult semantic analysis of source code or complex natural language understanding. Another advantage of this approach is that it is less influenced by subjective factors, e.g., experience and preference of software engineers.

2. How to Install

Eclipse: the tool is tested on Kepler Service Release 2. We expect it to work on any Eclipse that relies on *jdt.core.manipulation_1.5.0*

Files needed:

RenameExpand   jdt.core.manipulation   ltk.core.refactoring

org.eclipse.jdt.core.manipulation_1.5.0.v20130605-1748.jar
org.eclipse.ltk.core.refactoring_3.6.100.v20130605-1748.jar
RenameExpand_1.0.0.201502052050.jar

Installation:

1) Copy all of the three jar files into the plugins folder of your Eclipse (recommended)
   e.g., *C:\eclipse\plugins*

2) Install via the update site:
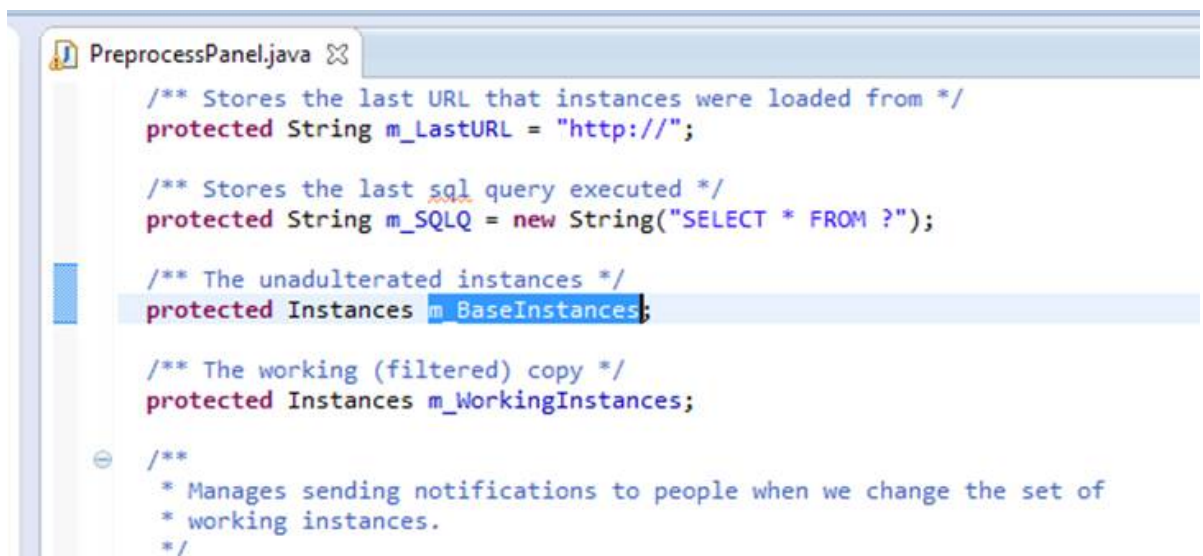   http://www.sei.pku.edu.cn/~liuhui04/tools/rename/RnameExpandUpdateSite

   * The update site cannot replace the original jdt.core.manipulation or ltk.core.refactoring with the customized versions. Consequently, if you install the *Rename Expander* via update site, please manually copy these two jar files into the plugins folder of your Eclipse.

   jdt.core.manipulation   ltk.core.refactoring

3. How to Use

1) Open Java Editor and select a software entity to rename

2) Rename the selected entity (with rename refactoring tool provided by Eclipse)

3) The tool tries to recommend renaming opportunities. If any opportunities are identified, the red mark is presented on the left margin beside the renamed entity.

4) Click on the red mark, and suggestions are presented as follows

```
PreprocessPanel.java ⊠

    /** Stores the last URL that instances were loaded from */
    protected String m_LastURL = "http://";

    /** Stores the last sql query executed */
    protected String m_SQLQ = new String("SELECT * FROM ?");

    /** The unadulterated instances */
    protected Instances m_Instances;
```
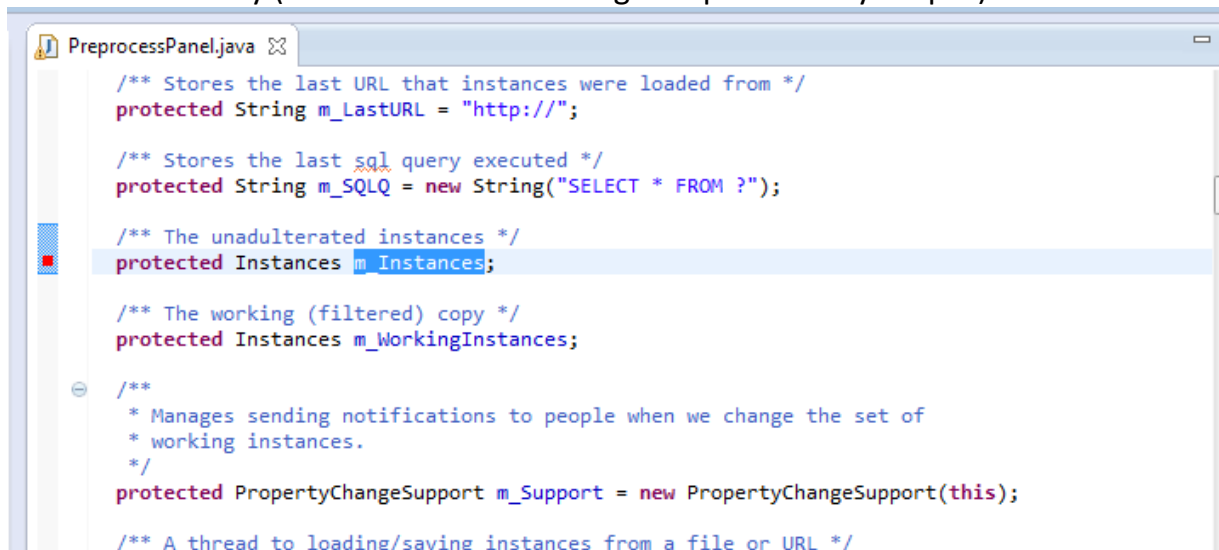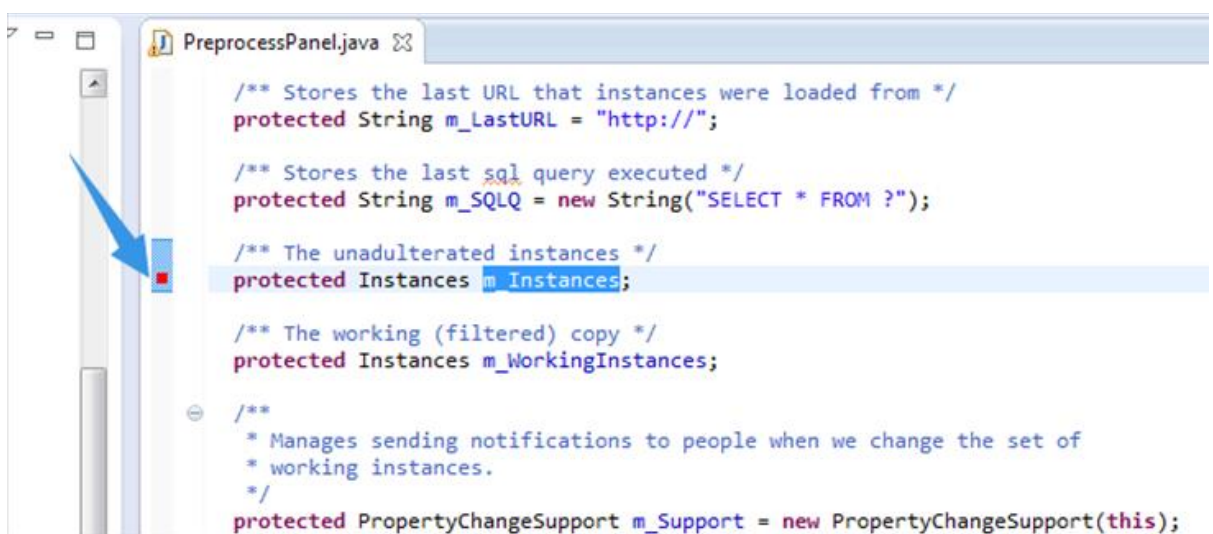
⊘ Would you want to rename method setBaseInstances to setInstances

```
set of

Support(this);
```

static {

6) Double click the suggestion, and the cursor goes to the suggested entity that needs to be renamed


```
PreprocessPanel.java ⊠

    /**
     * Tells the panel to use a new base set of instances.
     *
     * @param inst a set of Instances
     */
    public void setBaseInstances(Instances inst) {

        m_Instances = inst;
        try {
            Runnable r = new Runnable() {
            public void run() {
                m_BaseInstPanel.setInstances(m_Instances);
                m_AttPanel.setInstances(m_Instances);
                m_AttSummaryPanel.setInstances(m_Instances);
                m_Log.logMessage("Base relation is now "
                        + m_Instances.relationName()
                        + " (" + m_Instances.numInstances()
                        + " instances)");
                m_Log.statusMessage("OK");

                // clear most recently applied filters
                m_FiltersCopy = null;
                setWorkingInstances(m_Instances);
                m_ApplyBut.setEnabled(true);
```

7) If you decide to rename it, the tool will try to recommend new renamings based on it.


```
PreprocessPanel.java ⊠

    /**
     * Tells the panel to use a new base set of instances.
     *
     * @param inst a set of Instances
     */
    public void setInstances(Instances inst) {

        m_Instances = inst;
        try {
            Runnable r = new Runnable() {
            public void run() {
                m_BaseInstPanel.setInstances(m_Instances);
                m_AttPanel.setInstances(m_Instances);
                m_AttSummaryPanel.setInstances(m_Instances);
                m_Log.logMessage("Base relation is now "
                        + m_Instances.relationName()
                        + " (" + m_Instances.numInstances()
```

*The Java document used through in this introduction is Weka 3-3-4