- Program Creek
 - o <u>R</u>
 - Research
 - Machine Learning
 - Contact
- Simple Java
- Java 8
- Coding Interview
- Java
 - Java Basics
 - Java Object Oriented Concepts
 - Java Collections & Generics
 - Java File I/O
 - Java Database
 - o Java Multi-Threading
 - Java XML Parsing
 - Advanced Topics
- DP Stories
- API Examples
- Web Frameworks
 - o Struts 2
 - o GWT
 - o Spring
- Eclipse
 - Eclipse Architecture
 - o RCP
 - o JDT
- <u>Python</u>

Get Internal Comments By Using Eclipse JDT ASTParser



You may try to use Eclipse JDT ASTParser to parse Java source files, but cannot get the list of internal comments in the .java file. There is a method called getCommentList from the CompilationUnit class. But if you use it, you find that it returns a list of empty comments.

How to get internal comments by using Eclipse JDT ASTParser?

The answer is that comments return by the getCommentList method of CompilationUnit does not have the comment body. The comments are not visited. One way to visit the comments is calling accept method for each comment in the Comment List.

```
for (Comment comment : (List<Comment>) cu.getCommentList()) {
   comment.accept(new CommentVisitor(cu, sourceString));
}
```

Below is the jar files required to run the program. This is prerequisite condition, problem will happen if this is not configured right. Here is download link: lib.zip.

🛨 👼 org.eclipse.equinox.preferences_3.2.301.R35x_v20091117.jar - FindMethod/lib

A complete code example:

```
package ASTParser;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.List;
import org.eclipse.jdt.core.dom.AST;
import org.eclipse.jdt.core.dom.ASTParser;
import org.eclipse.jdt.core.dom.ASTVisitor;
import org.eclipse.jdt.core.dom.BlockComment;
import org.eclipse.jdt.core.dom.Comment;
import org.eclipse.jdt.core.dom.CompilationUnit;
import org.eclipse.jdt.core.dom.LineComment;
public class ParseJavaFile {
        public static void main(String[] args) throws IOException {
                String str = "/workspace/UltimateTest/src/ASTParser/a.java";
                String converted = readFileToString(str);
                parse(converted);
        }
        // use ASTParse to parse string
        public static void parse(final String str) {
                ASTParser parser = ASTParser.newParser(AST.JLS3);
                parser.setSource(str.toCharArray());
                parser.setKind(ASTParser.K COMPILATION UNIT);
                final CompilationUnit cu = (CompilationUnit) parser.createAST(null);
                for (Comment comment : (List<Comment>) cu.getCommentList()) {
                        comment.accept(new CommentVisitor(cu, str));
                }
        }
        // read file content into a string
        public static String readFileToString(String filePath) throws IOException {
                StringBuilder fileData = new StringBuilder(1000);
                BufferedReader reader = new BufferedReader(new FileReader(filePath));
                char[] buf = new char[10];
                int numRead = 0;
                while ((numRead = reader.read(buf)) != -1) {
                        String readData = String.valueOf(buf, 0, numRead);
                        fileData.append(readData);
                        buf = new char[1024];
                }
                reader.close();
                return fileData.toString();
```

```
}
}
//comment visitor
class CommentVisitor extends ASTVisitor {
        CompilationUnit cu;
        String source;
        public CommentVisitor(CompilationUnit cu, String source) {
                super();
                this.cu = cu;
                this.source = source;
        }
        public boolean visit(LineComment node) {
                int start = node.getStartPosition();
                int end = start + node.getLength();
                String comment = source.substring(start, end);
                System.out.println(comment);
                return true;
        }
        public boolean visit(BlockComment node) {
                int start = node.getStartPosition();
                int end = start + node.getLength();
                String comment = source.substring(start, end);
                System.out.println(comment);
                return true;
        }
}
```

a.java file

Output:



You May Also Like ...

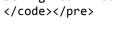
- 1. Eclipse JDT tutorial: Parse a Java method by using ASTParser
- 2. A complete standalone example of ASTParser
- 3. Use ASTParser to Parse a Java Project
- 4. Use JDT ASTParser to Parse Single .java files

Category >> <u>Java Development Tools (JDT)</u>

If you want to post syntax highlighted code and let me or someone else review it, please put the code inside <code> and </code> tags.

For example:

<code> String foo = "bar";







0 Comments



Program Creek

Sort by Best ▼



Start the discussion...

Be the first to comment.





Add Disgus to your site



Privacy







Hampton Inn Franklin / Milford



96.90 USD
Book Now! Best Rate
Guaranteed
Hilton.com

Copyright © 2008 - 2015 programcreek.com