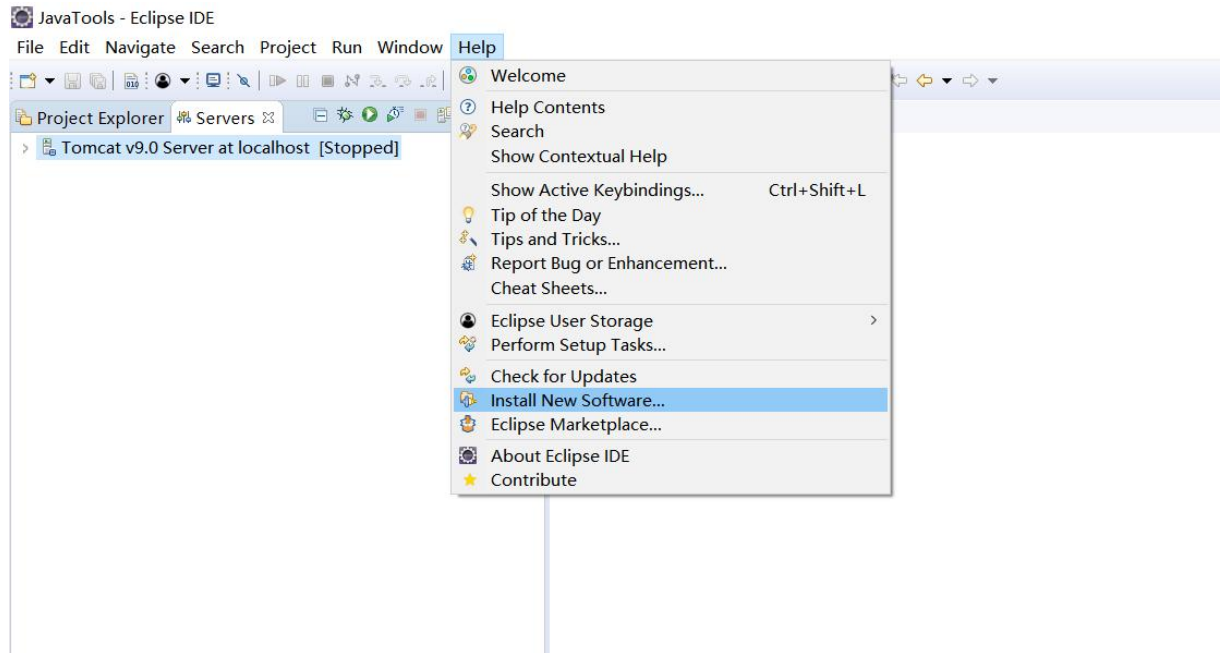


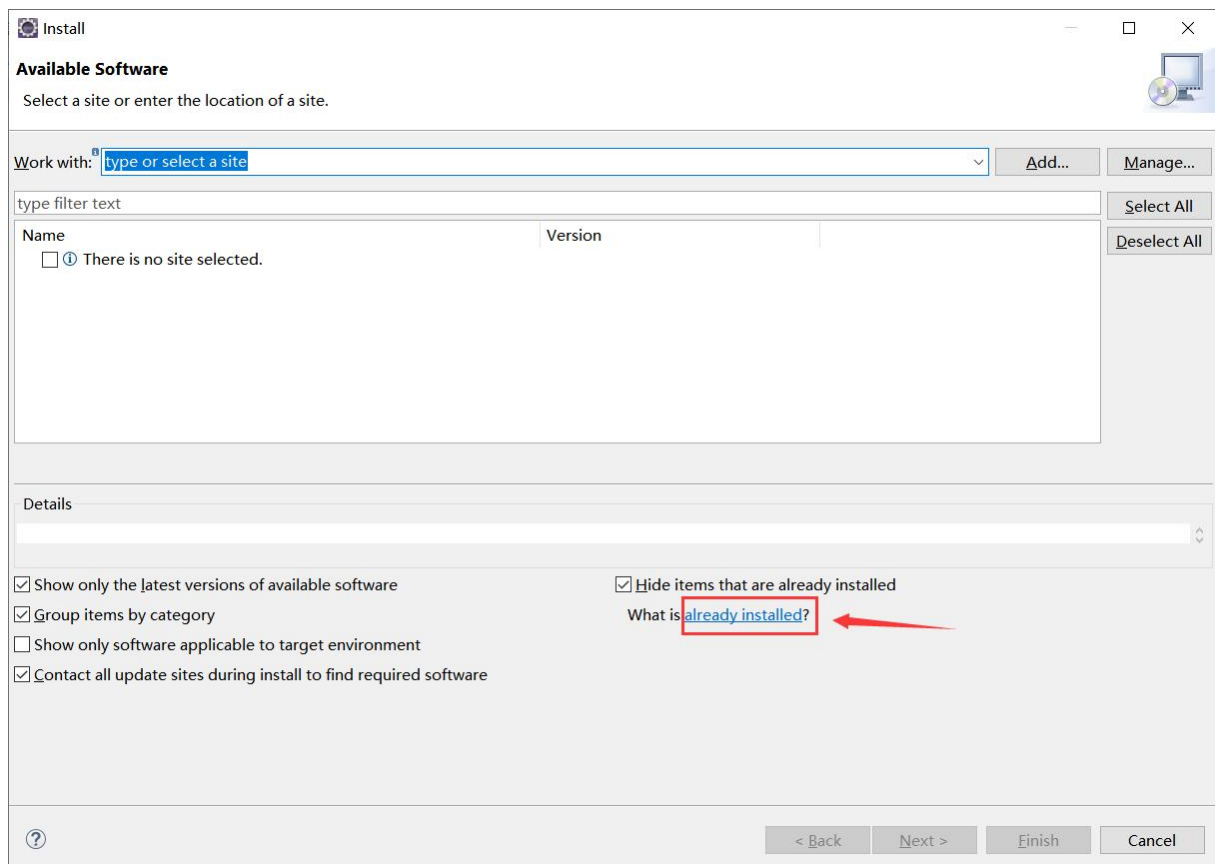
Eclipse's JDT compiler 生成 AST

1. 查看 JDT 是否安装

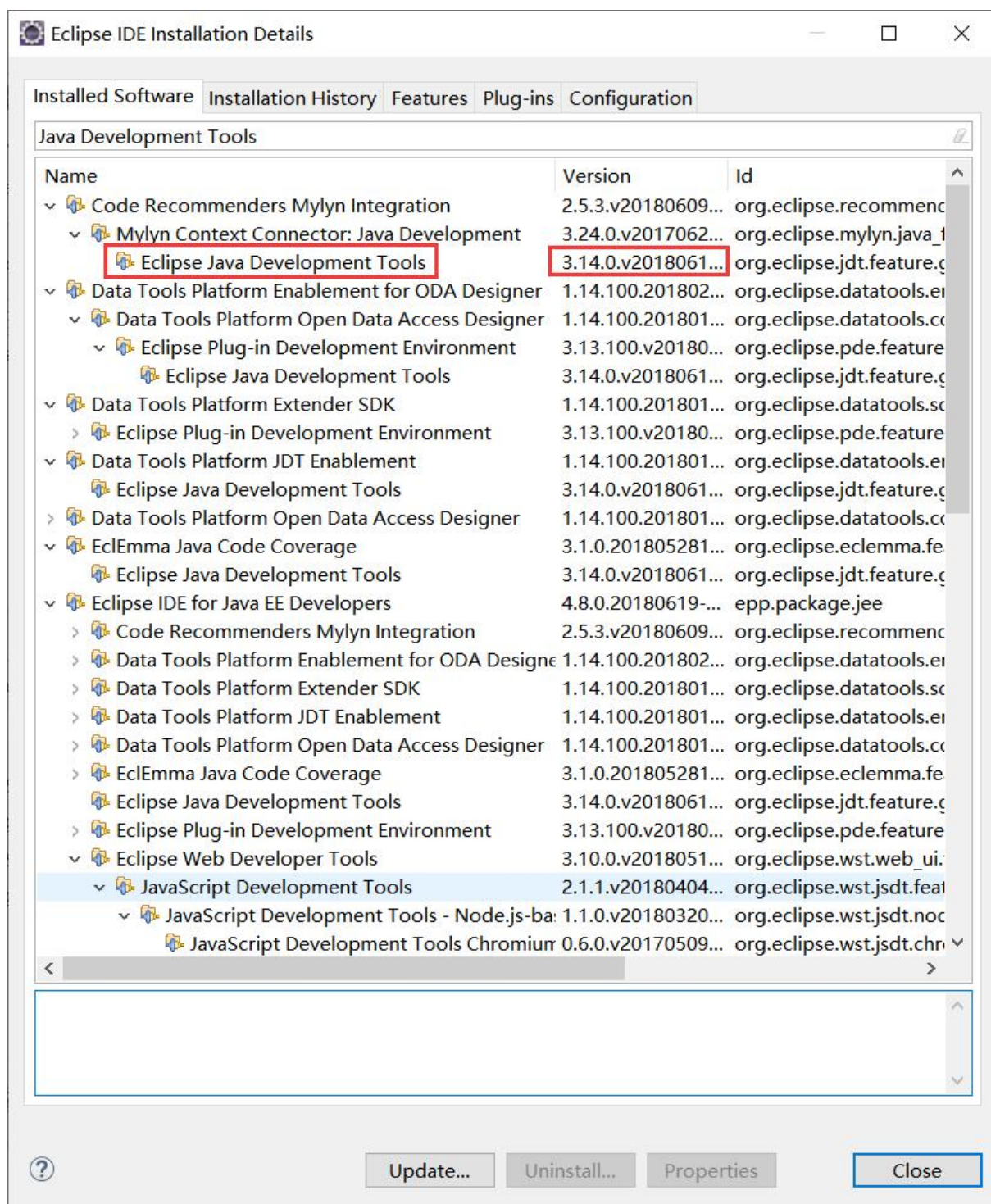
打开 eclipse, 点击【Help】—>【Install New Software】



点击【already installed】



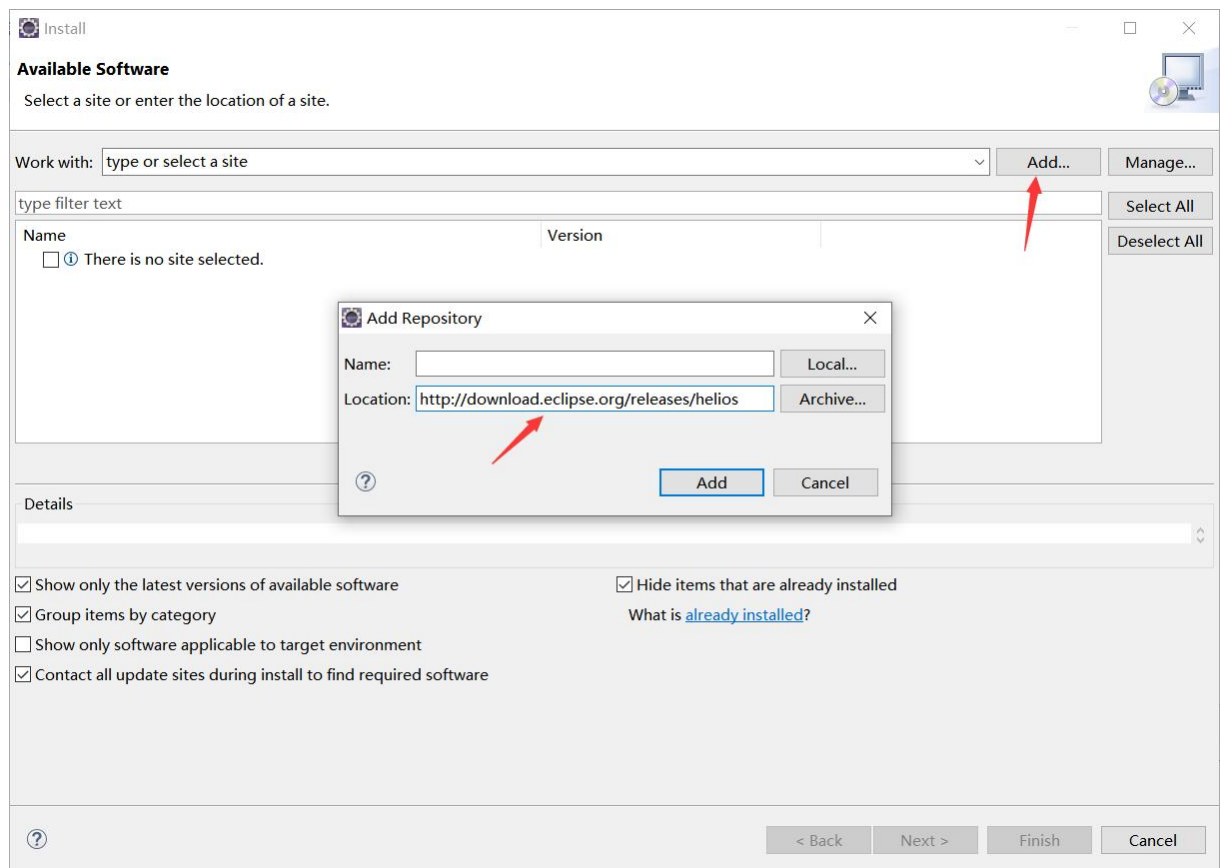
跳出来下面的界面, 搜索“Java Development Tools”, 如果有“Eclipse Java Development Tools”就说明已经安装了 JDT, 这里版本是 3.14.0。点击【Close】。



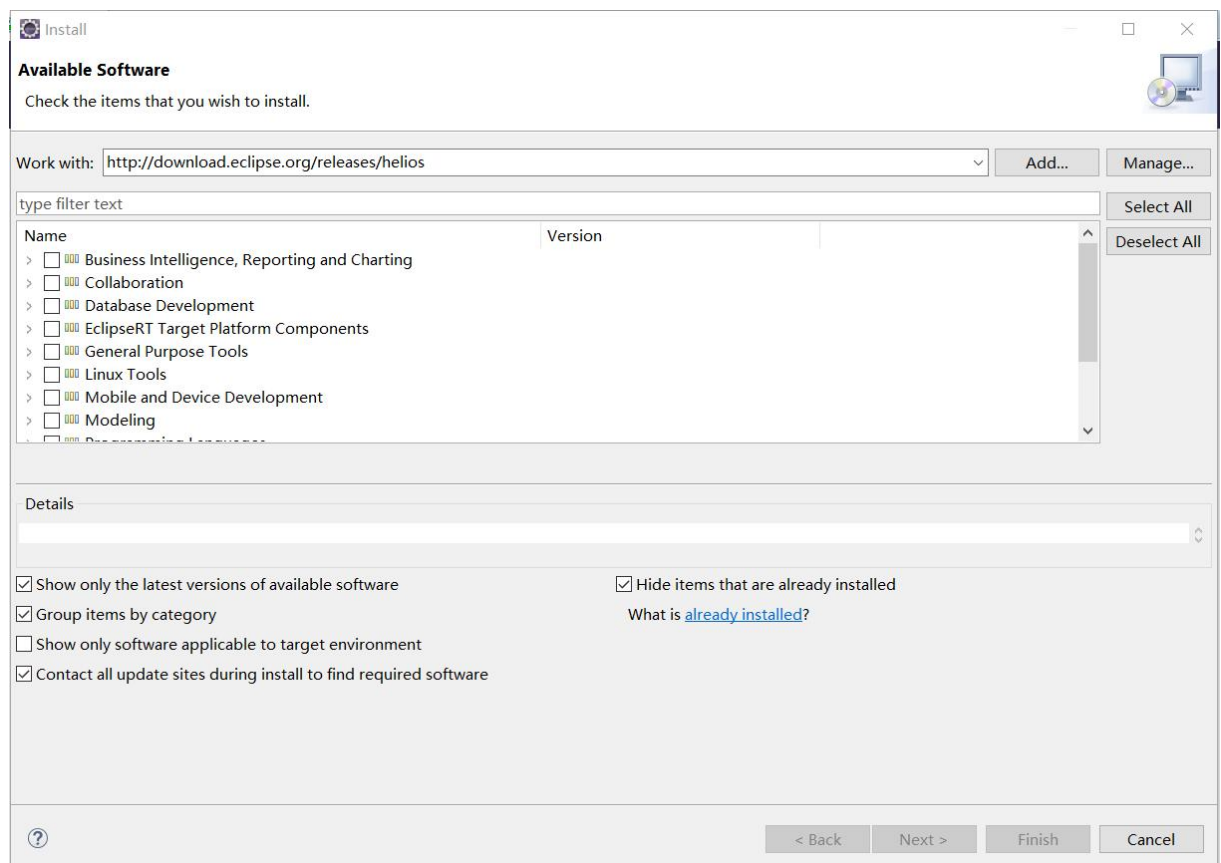
2. 安装 JDT

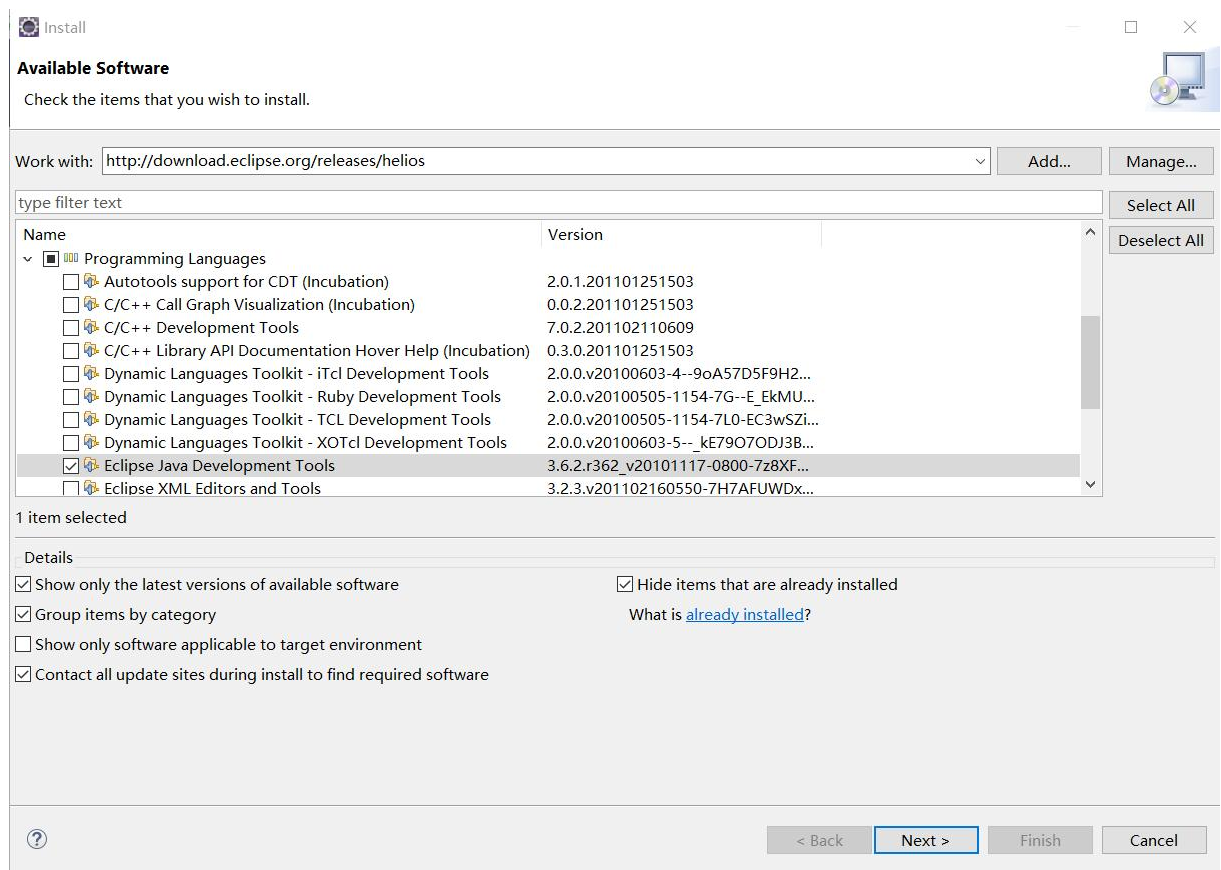
一般安装 eclipse 的时候都安装 JDT 了, 下面安装 JDT 的流程没有试过, 仅供参考。

还是点击【Help】—>【Install New Software】。点击【Add】, 在 Location 文本框输入“<http://download.eclipse.org/releases/helios>”, 再点击【Add】



等待加载一会儿，会跳出来下面的页面，找到“Programming Languages”—>“Eclipse Java Development Tool”，勾选，点【Next】就可以下载了。





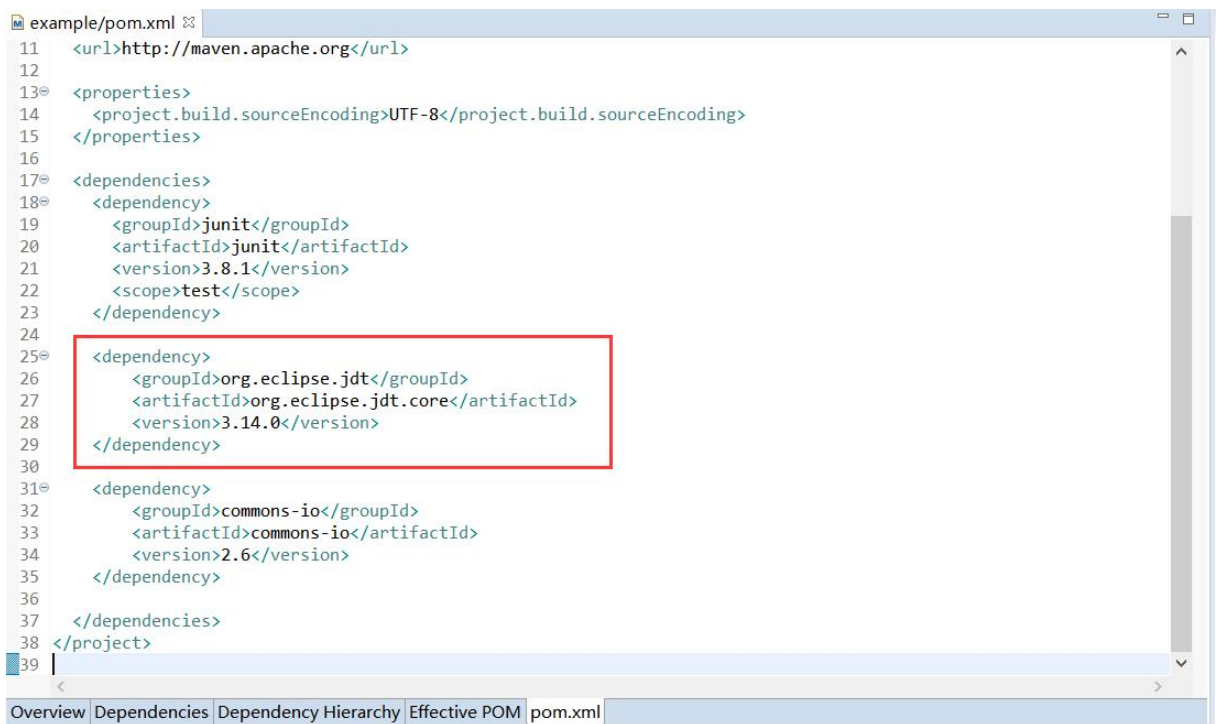
3. 使用 AST

新建一个 Maven Project，在 pom.xml 文件添加下面的依赖（版本号可能要对应修改一下，步骤 1 里面可以查看版本）。

```

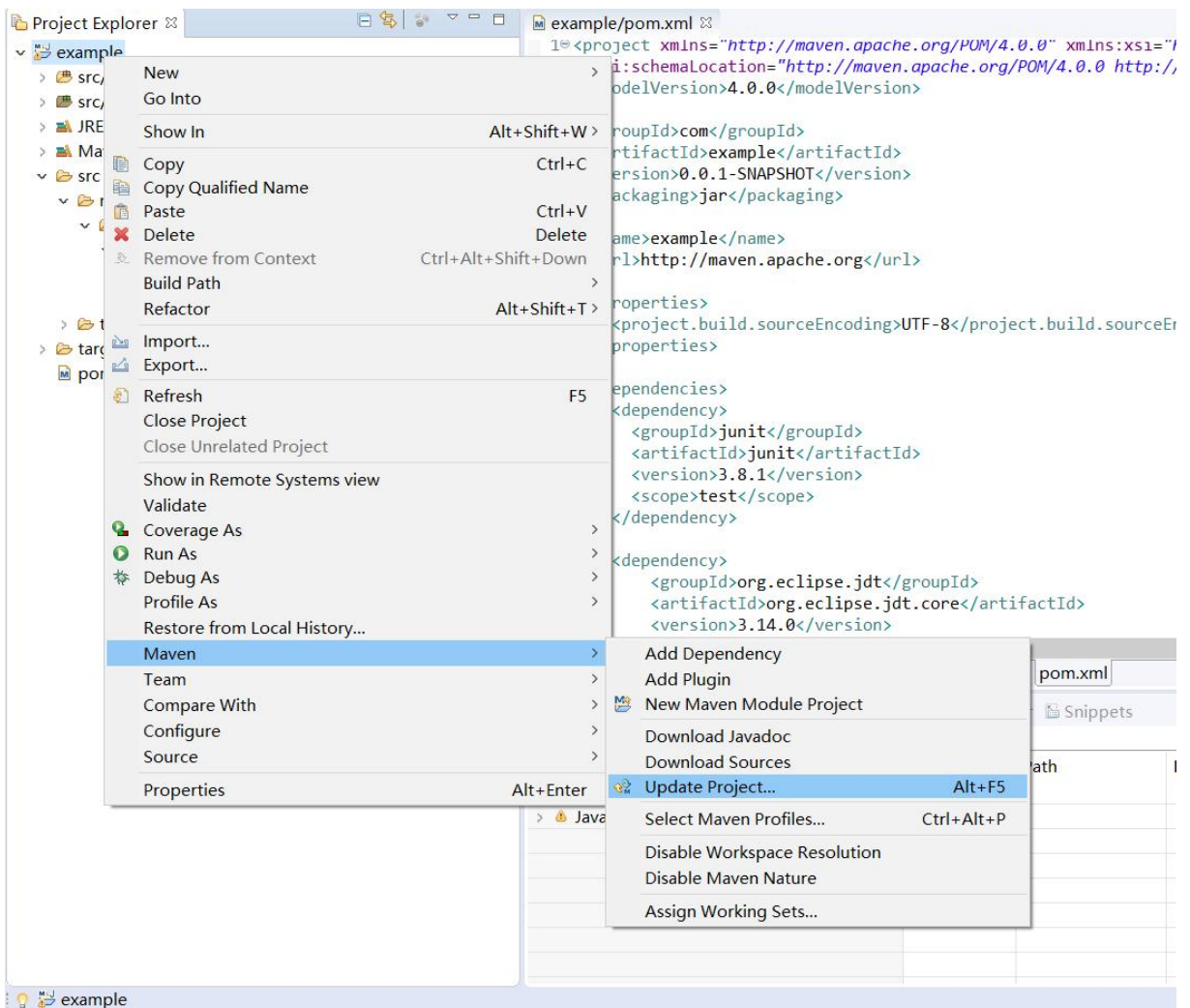
<dependency>
  <groupId>org.eclipse.jdt</groupId>
  <artifactId>org.eclipse.jdt.core</artifactId>
  <version>3.14.0</version>
</dependency>

```

```
11 <url>http://maven.apache.org</url>
12
13 <properties>
14   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15 </properties>
16
17 <dependencies>
18   <dependency>
19     <groupId>junit</groupId>
20     <artifactId>junit</artifactId>
21     <version>3.8.1</version>
22     <scope>test</scope>
23   </dependency>
24
25   <dependency>
26     <groupId>org.eclipse.jdt</groupId>
27     <artifactId>org.eclipse.jdt.core</artifactId>
28     <version>3.14.0</version>
29   </dependency>
30
31   <dependency>
32     <groupId>commons-io</groupId>
33     <artifactId>commons-io</artifactId>
34     <version>2.6</version>
35   </dependency>
36 </dependencies>
37 </project>
38
39
```

然后右键 project 的文件夹，点击【Maven】—>【Update Project】



加载完成后就可以使用 JDT 的 AST 工具了。下面的代码可以试一下 JDT 的 AST 工具能不能正常使用。

```
String source = String.join("\n",
    "public class HelloWorld {",
    "    public static void main(String[] args) {",
    "        System.out.println(\"Hello, World\");",
    "    }",
    "}");
```

```
ASTParser parser = ASTParser.newParser(AST.JLS8);
parser.setSource(source.toCharArray());
parser.setKind(ASTParser.K_COMPILATION_UNIT);
```

```
CompilationUnit result = (CompilationUnit) parser.createAST(new
NullProgressMonitor());
```

```
List types = result.types();
TypeDeclaration typeDec = (TypeDeclaration) types.get(0);
MethodDeclaration methodDec[] = typeDec.getMethods();
System.out.println("Method:");
for (MethodDeclaration method : methodDec)
{
    //get method name
    SimpleName methodName=method.getName();
    System.out.println("method name:"+methodName);

    //get method parameters
```

```

List param=method.parameters();

System.out.println("method parameters:"+param);


//get method return type
Type returnType=method.getReturnType2();

System.out.println("method return type:"+returnType);
}

System.out.print("finish\n");

public static void main( String[] args )
{
    String source = String.join("\n",
        "public class HelloWorld {",
        "    public static void main(String[] args) {",
        "        System.out.println(\"Hello, World\");",
        "    }",
        "}");

    ASTParser parser = ASTParser.newParser(AST.JLS8);
    parser.setSource(source.toCharArray());
    parser.setKind(ASTParser.K_COMPILATION_UNIT);

    CompilationUnit result = (CompilationUnit) parser.createAST(new NullProgressMonitor());

    List types = result.types();
    TypeDeclaration typeDec = (TypeDeclaration) types.get(0);

    MethodDeclaration methodDec[] = typeDec.getMethods();
    System.out.println("Method:");
    for (MethodDeclaration method : methodDec)
    {
        //get method name
        SimpleName methodName=method.getName();
        System.out.println("method name:"+methodName);

        //get method parameters
        List param=method.parameters();
        System.out.println("method parameters:"+param);

        //get method return type
        Type returnType=method.getReturnType2();
        System.out.println("method return type:"+returnType);

        //get method body
    }
    System.out.println("finish");
}

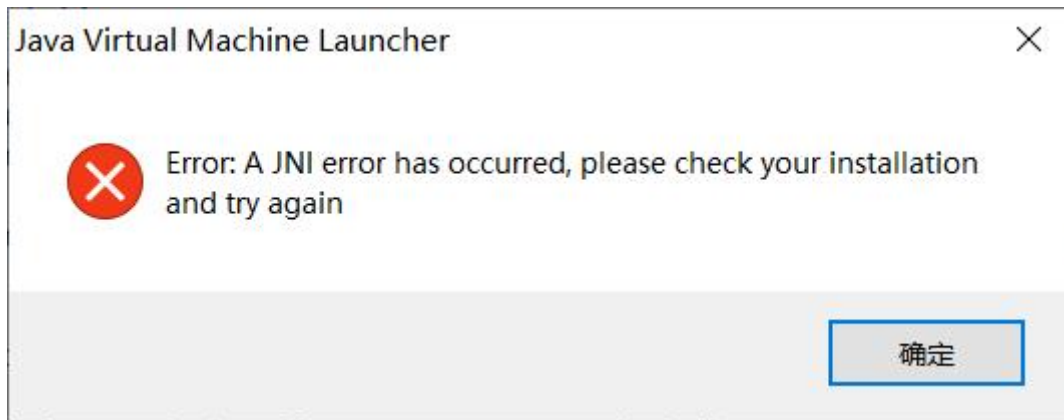
```

4. 遇到的问题

如果运行的时候出现下面的错误，报错


“…… has been compiled by a more recent version of the Java Runtime……”，说明 JDK 版本过低，需要升级 JDK（我升级到了 JDK11），然后更改项目使用的 JDK

(<http://t.zoukankan.com/no8g-p-13415652.html>)。



5. 节点的 label


在 <https://help.eclipse.org/latest/index.jsp> 搜索 ASTNode，选择 org.eclipse.jdt.core.dom 文档打开，在 Class Summary 部分找到 ASTNode 对应的文档打开。




Search: Go Scope: All topics

Search Results


3 matches in All topics: [Change scope](#)

 user

MoDisco Documentation > MoDisco User Guide > Technology > Java Metamodel The Java metamodel is the reflection of the Java language, as defined in version 3 of "Java Language Specification" from Sun Microsystems ("JLS3" corresponds to JDK 5). The Java metamo

 Adopting JDT Luna (4.4) mechanisms and APIs

JDT Plug-in Developer Guide > Plug-in Migration Guide > C This section describes changes that are required if you are trying to change your 4.3 plug-in to adopt the 4.4 mechanisms and APIs. AST.JLS8 for Java 8 constructs 1. AST.JL...

 org.eclipse.jdt.core.dom (Eclipse JDT API Specification)

JDT Plug-in Developer Guide > Reference > API Reference JavaScript is disabled on your browser. Skip navigation links Overview Package Class Use Tree Deprecated Index Help Eclipse JDT 2022-06 (4.24) All Classes SEARCH: JavaScrip...

ArrayAccess

OVERVIEW **PACKAGE** CLASS USE TREE

Array type

AssertStatement

Assignment

Assignment.Operator

AST

ASTMatcher

ASTNode

ASTParser

ASTRequestor

ASTVisitor

Block

往下找到 Field Detail 部分，列举了所有节点的 label。

Field Detail

ANONYMOUS_CLASS_DECLARATION

```
public static final int ANONYMOUS_CLASS_DECLARATION
```

Node type constant indicating a node of type AnonymousClassDeclaration.

See Also:

[AnonymousClassDeclaration](#), [Constant Field Values](#)

ARRAY_ACCESS

```
public static final int ARRAY_ACCESS
```

Node type constant indicating a node of type ArrayAccess.

See Also:

[ArrayAccess](#), [Constant Field Values](#)