Search Index

Java Code Examples for *org.eclipse.jdt.core.dom.ASTNode*

The following are *21* Jave code examples that show how to use the org.eclipse.jdt.core.dom.ASTNode class. These examples are extracted from open source projects. You can click 👍 to vote up the examples you like. Your votes will be used in an intelligent system to get more and better code examples. Thanks!

Example 1

👍 8

From project *besouro*, under directory */src/besouro/measure/*, in source file *JavaStatementMeter.java*

```
public JavaStatementMeter measureJavaFile(IFile file){
  ICompilationUnit cu=(ICompilationUnit)JavaCore.create(file);
  ASTParser parser=ASTParser.newParser(AST.JLS3);
  parser.setSource(cu);
  parser.setResolveBindings(true);
  ASTNode root=parser.createAST(null);
  JavaStatementMeter meter=new JavaStatementMeter();
  root.accept(meter);
  return meter;
}
```

Example 2

👍 8

From project *CIDE*, under directory */CIDE_Export_JDTBase/src/de/ovgu/cide/export/*, in source file *CopiedNaiveASTFlattener.java*

```
/** 
 * Appends the text representation of the given modifier flags, followed by a single space. Used for 3.0 modifiers and annotations.
 * @param ext the list of modifier and annotation nodes (element type: <code>IExtendedModifiers</code>)
 */
protected void printModifiers(List ext){
  for (Iterator it=ext.iterator(); it.hasNext(); ) {
    ASTNode p=(ASTNode)it.next();
    p.accept(this);
    this.buffer.append(" ");
  }
}
```

Example 3

👍 8

From project *CIDE*, under directory */CIDE_Export_JDTBase/src/de/ovgu/cide/export/*, in source file *CopiedNaiveASTFlattener.java*

```
public boolean visit(Javadoc node){
  printIndent();
  this.buffer.append("/** ");
  for (Iterator it=node.tags().iterator(); it.hasNext(); ) {
    ASTNode e=(ASTNode)it.next();
    e.accept(this);
  }
  this.buffer.append("\n */\n");
  return false;
}
```

Example 4

👍 8

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *JakColorChecker.java*

```
@Override protected boolean visitNode(ASTNode node){
  if (!colorManager.getOwnColors(node).isEmpty()) {
    Set<IFeature> nodeColors=getColors(node);
    ASTNode parent=node.getParent();
    if (parent != null && !nodeColors.equals(colorManager.getColors(parent))) {
      UnsupportedColoring er=isSupportedColoring(node);
      if (er != null)       unsupportedColorings.add(er);
    }
  }
  return super.visitNode(node);
}
```

Example 5

👍 8

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *JakFeatureRefactorer.java*

```
protected void addStatements(List<Statement> newStatements,List<Statement> target){
  for (  Statement stmt : newStatements) {
    ASTNode excluded=RefactoringUtils.getSubtreeRuleExceptionNode(stmt,colorManager,derivative);
    if (excluded != null) {
    }
    target.add(stmt);
  }
}
```

Example 6

👍 7

From project *android_sdk*, under directory */eclipse/plugins/com.android.ide.eclipse.adt/src/com/android/ide/eclipse/adt/internal/refactorings/extractstring/*, in source file *ReplaceStringsVisitor.java*

```
@SuppressWarnings("unchecked") @Override public boolean visit(StringLiteral node){
  if (node.getLiteralValue().equals(mOldString)) {
    boolean useGetResource=false;
    useGetResource=examineVariableDeclaration(node) || examineMethodInvocation(node);
    Name qualifierName=mAst.newName(mRQualifier + ".string");
    SimpleName idName=mAst.newSimpleName(mXmlId);
    ASTNode newNode=mAst.newQualifiedName(qualifierName,idName);
    String title="Replace string by ID";
    if (useGetResource) {
      Expression context=methodHasContextArgument(node);
      if (context == null && !isClassDerivedFromContext(node)) {
        context=findContextFieldOrMethod(node);
        if (context == null) {
          context=mAst.newSimpleName("Context");
        }
      }
      MethodInvocation mi2=mAst.newMethodInvocation();
      mi2.setName(mAst.newSimpleName("getString"));
      mi2.setExpression(context);
      mi2.arguments().add(newNode);
      newNode=mi2;
      title="Replace string by Context.getString(R.string...)";
    }
    TextEditGroup editGroup=new TextEditGroup(title);
    mEditGroups.add(editGroup);
    mRewriter.replace(node,newNode,editGroup);
  }
  return super.visit(node);
}
```

Example 7

👍 7

From project *android_sdk*, under directory */eclipse/plugins/com.android.ide.eclipse.adt/src/com/android/ide/eclipse/adt/internal/refactorings/extractstring/*, in source file *ReplaceStringsVisitor.java*

```
/** 
 * Examines if the StringLiteral is part of of an assignment to a string, e.g. String foo = id. The parent fragment is of syntax "var = expr" or "var|
 */
private boolean examineVariableDeclaration(StringLiteral node){
  VariableDeclarationFragment fragment=findParentClass(node,VariableDeclarationFragment.class);
  if (fragment != null) {
    ASTNode parent=fragment.getParent();
    Type type=null;
    if (parent instanceof VariableDeclarationStatement) {
      type=((VariableDeclarationStatement)parent).getType();
    }
 else     if (parent instanceof VariableDeclarationExpression) {
      type=((VariableDeclarationExpression)parent).getType();
    }
    if (type instanceof SimpleType) {
      return isJavaString(type.resolveBinding());
    }
  }
  return false;
}
```

Example 8

👍 7

From project *CIDE*, under directory */CIDE_Export_JDTBase/src/de/ovgu/cide/export/*, in source file *CopiedNaiveASTFlattener.java*

```
public boolean visit(TagElement node){
  if (node.isNested()) {
    this.buffer.append("{");
  }
 else {
    this.buffer.append("\n * ");
  }
  boolean previousRequiresWhiteSpace=false;
  if (node.getTagName() != null) {
    this.buffer.append(node.getTagName());
    previousRequiresWhiteSpace=true;
  }
  boolean previousRequiresNewLine=false;
  for (Iterator it=node.fragments().iterator(); it.hasNext(); ) {
    ASTNode e=(ASTNode)it.next();
    boolean currentIncludesWhiteSpace=(e instanceof TextElement);
    if (previousRequiresNewLine && currentIncludesWhiteSpace) {
      this.buffer.append("\n * ");
    }
    previousRequiresNewLine=currentIncludesWhiteSpace;
    if (previousRequiresWhiteSpace && !currentIncludesWhiteSpace) {
      this.buffer.append(" ");
    }
    e.accept(this);
    previousRequiresWhiteSpace=!currentIncludesWhiteSpace && !(e instanceof TagElement);
  }
  if (node.isNested()) {
    this.buffer.append("}");
  }
  return false;
}
```

Example 9

👍 7

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *FieldMethodVisibilityLifter.java*

```
private boolean visitName(Name node){
  IBinding binding=node.resolveBinding();
  if (binding == null)   return true;
  ASTNode declaration=compUnit.findDeclaringNode(binding);
  if (declaration instanceof MethodDeclaration) {
    makeProtected((MethodDeclaration)declaration);
  }
 else   if (declaration instanceof VariableDeclarationFragment) {
    VariableDeclarationFragment variableFragment=(VariableDeclarationFragment)declaration;
    if (variableFragment.getParent() instanceof FieldDeclaration) {
      FieldDeclaration fieldDecl=(FieldDeclaration)variableFragment.getParent();
      makeProtected(fieldDecl);
    }
  }
  return false;
}
```

Example 10

👍 7

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *JakHookMethodHelper.java*

```
private List<Name> calcThrownExceptions(MethodDeclaration method,List<Statement> statements2){
  ArrayList<Name> result=new ArrayList<Name>();
  result.addAll(method.thrownExceptions());
  ASTNode node=statements2.get(0);
  while (node != null && node != method) {
    if (node.getLocationInParent() == TryStatement.BODY_PROPERTY) {
      TryStatement tryStmt=(TryStatement)node.getParent();
      for (      CatchClause cc : (List<CatchClause>)tryStmt.catchClauses()) {
        Type exceptionType=cc.getException().getType();
        if (!containsException(result,exceptionType.toString()))         if (!exceptionType.toString().startsWith("Return"))         result.add(metho
      }
    }
    node=node.getParent();
  }
  return result;
}
```

Example 11

👍 5

From project *android_sdk*, under directory */eclipse/plugins/com.android.ide.eclipse.adt/src/com/android/ide/eclipse/adt/internal/refactorings/extractstring/*, in source file *ReplaceStringsVisitor.java*

```
/** 
 * Walks up the node hierarchy and returns the first ASTNode of the requested class. Only look at parents. Implementation note: this is a generic meth
 */
@SuppressWarnings("unchecked") private <T extends ASTNode>T findParentClass(ASTNode node,Class<T> clazz){
  for (node=node.getParent(); node != null; node=node.getParent()) {
    if (node.getClass().equals(clazz)) {
      return (T)node;
    }
  }
  return null;
}
```

Example 12

👍 5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/ast/*, in source file *JakClassRefinement.java*

```
public void initializeFromType(TypeDeclaration type){
  modifiers.clear();
  modifiers.addAll(ASTNode.copySubtrees(ast,type.modifiers()));
  isInterface=type.isInterface();
  superInterfaceTypes.clear();
  superType=null;
  typeParameters.clear();
  typeParameters.addAll(ASTNode.copySubtrees(ast,type.typeParameters()));
  name=(SimpleName)ASTNode.copySubtree(ast,type.getName());
}
```

Example 13

👍 5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/ast/*, in source file *JakClassRefinement.java*

```
JakClassRefinement clone(AST target){
  JakClassRefinement result=new JakClassRefinement(parent,target,options);
  result.javadoc=(this.javadoc);
  result.name=ast.newSimpleName(name.getIdentifier());
  result.isInterface=isInterface;
  result.modifiers.addAll(ASTNode.copySubtrees(target,modifiers));
  result.typeParameters().addAll(ASTNode.copySubtrees(target,typeParameters()));
  result.superInterfaceTypes().addAll(ASTNode.copySubtrees(target,superInterfaceTypes()));
  result.superType=(Type)ASTNode.copySubtree(target,result.superType);
  result.refinements.addAll(ASTNode.copySubtrees(target,refinements));
  result.introductions.addAll(ASTNode.copySubtrees(target,introductions));
  result.fields.addAll(ASTNode.copySubtrees(target,fields));
  return result;
}
```

Example 14

5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/ast/*, in source file *JakClassRefinement.java*

```
public void accept(ASTVisitor visitor){
  if (visitor instanceof IJakASTVisitor) {
    boolean visitChildren=((IJakASTVisitor)visitor).visit(this);
    if (visitChildren) {
      for (      ASTNode n : modifiers)       n.accept(visitor);
      name.accept(visitor);
      for (      ASTNode n : typeParameters)       n.accept(visitor);
      for (      ASTNode n : superInterfaceTypes)       n.accept(visitor);
      for (      ASTNode n : fields)       n.accept(visitor);
      for (      ASTNode n : introductions)       n.accept(visitor);
      for (      ASTNode n : refinements)       n.accept(visitor);
      if (!options.getMethodObjectsInStaticToplevelClass())       for (      JakClassRefinement n : innerClassRefinements.values())       n.accept(vis
    }
    ((IJakASTVisitor)visitor).endVisit(this);
  }
}
```

Example 15

5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *ExportJavaFileJob.java*

```
private void createFilesForInnerClasses(List<BodyDeclaration> bodyDeclarations,PackageDeclaration packageDecl,List<ImportDeclaration> imports,IContair
  List<TypeDeclaration> innerClasses=extractInnerClasses(bodyDeclarations);
  for (  TypeDeclaration innerClass : innerClasses) {
    CompilationUnit container=innerClass.getAST().newCompilationUnit();
    if (packageDecl != null)     container.setPackage((PackageDeclaration)ASTNode.copySubtree(innerClass.getAST(),packageDecl));
    container.imports().addAll(ASTNode.copySubtrees(innerClass.getAST(),imports));
    container.types().add(innerClass);
    MethodObjectHelper.stripMethodObjectAnnotation(innerClass);
    MethodObjectHelper.stripStaticModifier(innerClass.modifiers());
    createFileForBase(innerClass,container,targetDirectory,targetLayer);
  }
}
```

Example 16

5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *JakColorChecker.java*

```
private UnsupportedColoring isSupportedColoring(ASTNode node){
  if (node instanceof CompilationUnit)   return null;
  if (isImportStatement(node))   return null;
  if (isMovableType(node))   return null;
  if (isExtendsOrImplements(node))   return null;
  if (isRefinableMethodOrField(node))   return null;
  if (isSubtreeRuleException(node))   return new UnsupportedColoring(node,"Multiple exceptions from subtree rule in one node");
  if (isAllStatements(node))   return null;
  if (isAroundAdvice(node))   return null;
  UnsupportedColoring r=isStatementExtractableWithHook(node);
  if (r != dummy)   return r;
  if (isStatementOutsideBlock(node))   return new UnsupportedColoring(node,"Colored statement must be placed in block");
  return new UnsupportedColoring(node,"Export not supported (too fine granularity?)");
}
```

Example 17

5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *JakColorChecker.java*

```
private boolean isExtendsOrImplements(ASTNode node){
  if (isMovableType(node.getParent())) {
    if (node.getLocationInParent() == TypeDeclaration.SUPER_INTERFACE_TYPES_PROPERTY) {
      return true;
    }
    if (node.getLocationInParent() == TypeDeclaration.SUPERCLASS_TYPE_PROPERTY) {
      return true;
    }
  }
  return false;
}
```

Example 18

5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *JakFeatureRefactorer.java*

```
private boolean refactorAllStatements(MethodDeclaration method){
  if (!RefactoringUtils.canRefactorAllStatements(method,colorManager,derivative))   return false;
  AST ast=method.getAST();
  MethodDeclaration refinement=(MethodDeclaration)ASTNode.copySubtree(ast,method);
  List<Statement> baseBody=method.getBody().statements();
  List<Statement> refinementBody=refinement.getBody().statements();
  Statement exception=RefactoringUtils.findSubtreeRuleException(baseBody,colorManager,derivative);
  if (exception != null) {
    Statement superCall=createSuperCall(method,ast,false,null);
    replaceStatement(exception,superCall);
  }
  List<Statement> baseStatements=new ArrayList<Statement>(baseBody);
  baseBody.clear();
  refinementBody.clear();
  refinementBody.addAll(baseStatements);
  if (exception != null) {
    RefactoringUtils.addStatementOrBlockContent(exception,baseBody);
  }
 else {
    Statement superCall=createSuperCall(method,ast,true,null);
    if (superCall != null)     refinementBody.add(superCall);
  }
  targetCompUnit.getRefinement().addRefinementForMethod(method,refinement);
  return true;
}
```

Example 19

5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *JakFeatureRefactorer.java*

```
private void extractAroundMethodRefinement(MethodDeclaration method,List<Statement> refactorFirst,List<Statement> refactorLast){
  AST ast=method.getAST();
  MethodDeclaration copiedMethod=(MethodDeclaration)ASTNode.copySubtree(ast,method);
  Statement superCall=createSuperCall(method,ast,refactorLast.isEmpty(),null);
  List<Statement> targetBody=copiedMethod.getBody().statements();
  targetBody.clear();
  targetBody.addAll(refactorFirst);
  if (superCall != null)   targetBody.add(superCall);
  targetBody.addAll(refactorLast);
  if (!refactorLast.isEmpty() && !RefactoringUtils.isVoid(method.getReturnType2())) {
    targetBody.add(createReturnStatement(ast));
  }
  targetCompUnit.getRefinement().addRefinementForMethod(method,copiedMethod);
}
```

Example 20

5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *MethodObjectHelper.java*

```
/** 
 * creates a constructor with the parameter of the methods arguments. all parameters are assigned to fields with the same name that are created as wel
 * @param methodObjectClass
 * @param method
 * @param isStatic
 */
private static void createConstructor(TypeDeclaration moClass,MethodDeclaration method,boolean isStatic){
  if (method.parameters().isEmpty() && !isStatic)   return;
  AST ast=moClass.getAST();
  MethodDeclaration constructorDecl=ast.newMethodDeclaration();
  constructorDecl.setConstructor(true);
  constructorDecl.setName(ast.newSimpleName(moClass.getName().getIdentifier()));
  constructorDecl.setBody(ast.newBlock());
  List<SingleVariableDeclaration> parameterList=new ArrayList<SingleVariableDeclaration>(method.parameters());
  if (isStatic) {
    SingleVariableDeclaration _thisVar=ast.newSingleVariableDeclaration();
    _thisVar.setName(ast.newSimpleName("_this"));
    TypeDeclaration mainClass=RefactoringUtils.getContainingType(method);
    _thisVar.setType(ast.newSimpleType(ast.newSimpleName(mainClass.getName().getIdentifier())));
    parameterList.add(0,_thisVar);
  }
  for (  SingleVariableDeclaration p : parameterList) {
    String varName=p.getName().getIdentifier();
    constructorDecl.parameters().add(ASTNode.copySubtree(ast,p));
    addField(moClass,p.getType(),varName,null);
    FieldAccess fieldAccess=ast.newFieldAccess();
    fieldAccess.setExpression(ast.newThisExpression());
    fieldAccess.setName(ast.newSimpleName(varName));
    Assignment assignment=ast.newAssignment();
    assignment.setLeftHandSide(fieldAccess);
    assignment.setRightHandSide(ast.newSimpleName(varName));
    constructorDecl.getBody().statements().add(ast.newExpressionStatement(assignment));
  }
  moClass.bodyDeclarations().add(0,constructorDecl);
}
```

Example 21

👍 5

From project *CIDE*, under directory */CIDE_Export_Physical/src/de/ovgu/cide/export/physical/ahead/*, in source file *MethodObjectHelper.java*

```
/** 
 * @param methodObjectClass
 * @param type original ASTNode is sufficient (is copied internally)
 * @param name
 * @param colors null if not used
 */
private static void addField(TypeDeclaration methodObjectClass,Type type,String name,Set<IFeature> colors){
  AST ast=type.getAST();
  VariableDeclarationFragment fragment=ast.newVariableDeclarationFragment();
  fragment.setName(ast.newSimpleName(name));
  FieldDeclaration field=ast.newFieldDeclaration(fragment);
  field.setType((Type)ASTNode.copySubtree(ast,type));
  Modifier modifier=ast.newModifier(Modifier.ModifierKeyword.PROTECTED_KEYWORD);
  field.modifiers().add(modifier);
  methodObjectClass.bodyDeclarations().add(field);
}
```

The examples above are mined from open source projects. Each example has a reference to its resource, but the http link may not be provided due to the evoluation of the project.