

JavaScript is disabled on your browser.

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

jminusminus

Class JStatementExpression

- [java.lang.Object](#)
- [jminusminus.AST](#)
- [jminusminus.JStatement](#)
- [jminusminus.JStatementExpression](#)

.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
class JStatementExpression
extends JStatement
```

The AST node for an expression that appears as a statement. Only the expressions that have a side-effect are valid statement expressions.

- **Field Summary**

Fields

Modifier and Type

**Field and
Description**

```
(package private) JExpression expr
The expression.
```

- **Fields inherited from class jminusminus.JAST**

`compilationUnit, line`

- **Constructor Summary**

Constructors

Constructor and Description

```
JStatementExpression(int line, JExpression expr)
```

Construct an AST node for a statement expression given its line number, and expression.

- **Method Summary**

Methods	
Modifier and Type	Method and Description
JStatement	analyze (Context context) Analysis involves analyzing the encapsulated expression if indeed it is a statement expression, i.e., one with a side effect.
void	codegen (CLEmitter output) Generating code for the statement expression involves simply generating code for the encapsulated expression.
void	writeToStdOut (PrettyPrinter p) Write the information pertaining to this AST to STDOUT.

- **Methods inherited from class jminusminus.JAST**

line, partialCodegen

- **Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- **Field Detail**

- **expr**

JExpression expr

The expression.

- **Constructor Detail**

- **JStatementExpression**

publicJStatementExpression(intline,
JExpressionexpr)

Construct an AST node for a statement expression given its line number, and expression.

Parameters:

line - line in which the expression occurs in the source file.

expr - the expression.

- **Method Detail**

- **analyze**

publicJStatementanalyze(Contextcontext)

Analysis involves analyzing the encapsulated expression if indeed it is a statement expression, i.e., one with a side effect.

Specified by:

analyze in class JAST

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Parameters:

context - context in which names are resolved.

Returns:

the analyzed (and possibly rewritten) AST subtree.

- **codegen**

```
public void codegen(CLEmitter output)
```

Generating code for the statement expression involves simply generating code for the encapsulated expression.

Specified by:

`codegen` in class `JAST`

Parameters:

output - the code emitter (basically an abstraction for producing the .class file).

- **writeToStdOut**

```
public void writeToStdOut(PrettyPrinter p)
```

Description copied from class: JAST

Write the information pertaining to this AST to STDOUT.

Specified by:

`writeToStdOut` in class `JAST`

Parameters:

p - for pretty printing with indentation.

Assignment Project Exam Help

- [Prev Class](#)
- [Next Class](#)

<https://powcoder.com>

- [Frames](#)
- [No Frames](#)
- [All Classes](#)

Add WeChat powcoder

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)