

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 JPlusAssignOp

- [java.lang.Object](#)
- [jminusminus.JAST](#)
- [jminusminus.JStatement](#)
- [jminusminus.JExpression](#)
- [jminusminus.JBinaryExpression](#)
- [jminusminus.JAssignment](#)
- [jminusminus.JPlusAssignOp](#)

.

```
class JPlusAssignOp  
extends JAssignment
```

The AST node for a += expression. A += expression has two operands: a lhs and a rhs.

- **Field Summary**
- **Fields inherited from class [jminusminus.JBinaryExpression](#)**
[lhs](#), [operator](#), [rhs](#)
- **Fields inherited from class [jminusminus.JExpression](#)**
[isStatementExpression](#), [type](#)
- **Fields inherited from class [jminusminus.JAST](#)**
[compilationUnit](#), [line](#)
- **Constructor Summary**

Constructors

Constructor and Description

JPlusAssignOp(int line, JExpression lhs, JExpression rhs)
Construct the AST node for a += expression given its lhs and rhs operands.

- **Method Summary**

Methods

**Modifier and
Type**

Method and Description

JExpression	analyze (Context context) Analyze the lhs and rhs, rewrite rhs as lhs + rhs (string concatenation) if lhs is a String, and set the result type.
void	codegen (CLEmitter output) Code generation for += involves, generating code for loading any necessary l-value onto the stack, for (unless a string concatenation) loading the r-value, for (unless a statement) copying the r-value to its proper place on the stack, and for doing the store.

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

writeToStdOut

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

codegen, isStatementExpression, type

- **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

- **Constructor Detail**

- **JPlusAssignOp**

```
public JPlusAssignOp(int line,
                    JExpression lhs,
                    JExpression rhs)
```

Construct the AST node for a += expression given its lhs and rhs operands.

Parameters:

- line - line in which the assignment expression occurs in the source file.
- lhs - the lhs operand.
- rhs - the rhs operand.

- **Method Detail**

- **analyze**

```
public JExpression analyze(Context context)
```

Analyze the lhs and rhs, rewrite rhs as lhs + rhs (string concatenation) if lhs is a String, and set the result type.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Specified by:

`analyze` in class `JExpression`

Parameters:

`context` - context in which names are resolved.

Returns:

the analyzed (and possibly rewritten) AST subtree.

- **codegen**

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

Code generation for `+=` involves, generating code for loading any necessary l-value onto the stack, for (unless a string concatenation) loading the r-value, for (unless a statement) copying the r-value to its proper place on the stack, and for doing the store.

Specified by:

`codegen` in class `JAST`

Parameters:

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

- **Prev Class**
- **Next Class**

- **Frames**
- **No Frames**

- **All Classes**

- **Summary:**
- **Nested |**
- **Field |**
- **Constr |**
- **Method**

- **Detail:**
- **Field |**
- **Constr |**
- **Method**

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder