JavaScript is disabled on your browser.

jminusminus

Class JMultiplyOp

- java.lang.Object
  - jminusminus.JAST
    - jminusminus.JStatement
      - jminusminus.JExpression
        - jminusminus.JBinaryExpression
          - jminusminus.JMultiplyOp

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

.

```
class JMultiplyOp
extends JBinaryExpression
```

The AST node for a multiplication (*) expression.

- **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

```
JMultiplyOp(int line, JExpression lhs, JExpression rhs)
```

Construct an AST for a multiplication expression given its line number, and the lhs and rhs operands.

- **Method Summary**

Methods

Modifier and Type	Method and Description
JExpression	**analyze**(Context context) Analyzing the * operation involves analyzing its operands, checking types, and determining the result type.
void	**codegen**(CLEmitter output) Generating code for the * operation involves generating code for the two operands, and then the multiplication instruction.

- **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

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

- **Constructor Detail**

  - **JMultiplyOp**

```
publicJMultiplyOp(intline,
                  JExpressionlhs,
                  JExpressionrhs)
```

Construct an AST for a multiplication expression given its line number, and the lhs and rhs operands.

Parameters:

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

- **Method Detail**

  - **analyze**

```
publicJExpressionanalyze(Contextcontext)
```

Analyzing the * operation involves analyzing its operands, checking types, and determining the result type.

Specified by:

analyze in class JExpression

Parameters:

`context` - context in which names are resolved.

Returns:

the analyzed (and possibly rewritten) AST subtree.

- **codegen**

```
publicvoidcodegen(CLEmitteroutput)
```

Generating code for the * operation involves generating code for the two operands, and then the multiplication instruction.

Specified by:

`codegen` in class `JAST`

Parameters:

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

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder