

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 JEqualOp

- [java.lang.Object](#)
- [jminusminus.JAST](#)
- [jminusminus.JStatement](#)
- [jminusminus.JExpression](#)
- [jminusminus.JBinaryExpression](#)
- [jminusminus.JBooleanBinaryExpression](#)
- [jminusminus.JEqualOp](#)

.

```
class JEqualOp
extends JBooleanBinaryExpression
```

The AST node for an equality (==) expression. Implements short-circuiting branching.

- **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

JEEqualOp(int line, JExpression lhs, JExpression rhs)
Construct an AST node for an equality expression.

- **Method Summary**

Methods

**Modifier and
Type**

Method and Description

JExpression	analyze (Context context) Analyzing an equality expression means analyzing its operands and checking that the types match.
void	codegen (CLEmitter output, String targetLabel, boolean onTrue) Branching code generation for == operation.

- **Methods inherited from**

class jminusminus.JBooleanBinaryExpression

codegen

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

writeToStdOut

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

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**

- **JEEqualOp**

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

Construct an AST node for an equality expression.

Parameters:

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

lhs - lhs operand.

rhs - rhs operand.

- **Method Detail**

- **analyze**

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

Analyzing an equality expression means analyzing its operands and

checking that the types match.

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,  
                  String targetLabel,  
                  boolean onTrue)
```

Branching code generation for `==` operation.

Overrides:

`codegen` in class `JExpression`

Parameters:

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

`targetLabel` - target for generated branch instruction.

`onTrue` - should we branch on true?

- [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