

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 JInstanceOfOp

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

.

Assignment Project Exam Help  
<https://powcoder.com>

Add WeChat powcoder

```
class JInstanceOfOp
extends JExpression
```

The AST node for an instanceof expression, having two arguments: an expression and a reference type.

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

Constructors  
**Constructor and Description**

```
JInstanceOfOp(int line, JExpression expr, Type typeSpec)
Construct an AST node for an instanceof expression given its line number, the
relational expression and reference type.
```

- **Method Summary**

Methods	
Modifier and Type	Method and Description
<code>JInstanceOfOp</code>	<b><code>analyze</code></b> ( <code>Context</code> context) Analysis of an instanceof operation requires analyzing the expression to be tested, resolving the type was are testing for, and determining if the test is legal, or if the answer can be determined at compile time.
<code>void</code>	<b><code>codegen</code></b> ( <code>CLEmitter</code> output) Generate code for the type test.
<code>void</code>	<b><code>codegen</code></b> ( <code>CLEmitter</code> output, <code>String</code> targetLabel, <code>boolean</code> onTrue) Short-circuiting branching for instanceof.
<code>void</code>	<b><code>writeToStdOut</code></b> ( <code>PrettyPrinter</code> p) Write the information pertaining to this AST to STDOUT.

- **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`

**Add WeChat powcoder**

- **Constructor Detail**

- **`JInstanceOfOp`**

```
public JInstanceOfOp(int line,
                    JExpression expr,
                    Type typeSpec)
```

Construct an AST node for an instanceof expression given its line number, the relational expression and reference type.

**Parameters:**

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

`expr` - the expression denoting the value to be tested.

`typeSpec` - the reference type we are testing for.

- **Method Detail**

- **`analyze`**

```
public JInstanceOfOp analyze(Context context)
```

Analysis of an instanceof operation requires analyzing the expression to be tested, resolving the type was are testing for, and determining if the test is legal, or if the answer can be determined at compile time.

**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)
```

Generate code for the type test.

**Specified by:**

`codegen` in class `JAST`

**Parameters:**

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

- `codegen`

```
public void codegen(CLEmitter output,
    String targetLabel,
    boolean onTrue)
```

Short-circuiting branching for instanceof.

**Overrides:**

`codegen` in class `JExpression`

**Parameters:**

`output` - code emitter.

`targetLabel` - the label to which we should branch.

`onTrue` - do we branch on true?

- `writeToStdOut`

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

**Add WeChat powcoder**

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.

- Overview
- Package
- **Class**
- Tree
- Deprecated
- Index
- Help

- **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**