

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 JSubtractOp

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

.

Add WeChat powcoder

```
class JSubtractOp  
extends JBinaryExpression
```

The AST node for a subtraction (-) 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

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

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

- **Method Summary**

Methods

**Modifier and
Type**

Method and Description

<code>JExpression</code>	<code>analyze</code> (<code>Context</code> context) Analyzing the - operation involves analyzing its operands, checking types, and determining the result type.
<code>void</code>	<code>codegen</code> (<code>CLEmitter</code> output) Generating code for the - operation involves generating code for the two operands, and then the subtraction 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`

- **Constructor Detail**

- **`JSubtractOp`**

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

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

Parameters:

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

`lhs` - the lhs operand.

`rhs` - the rhs operand.

- **Method Detail**

- **`analyze`**

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

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

Specified by:

`analyze` in class `JExpression`

Parameters:

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

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 - operation involves generating code for the two operands, and then the subtraction instruction.

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