jminusminus

# Class JNegateOp

- java.lang.Object
  - jminusminus.JAST
    - jminusminus.JStatement
      - jminusminus.JExpression
        - jminusminus.JUnaryExpression
          - jminusminus.JNegateOp



.

```
class JNegateOp
extends JUnaryExpression
```

The AST node for a unary negation (-) expression.

- **Field Summary**
- **Fields inherited from class jminusminus.JUnaryExpression**
  `arg`
- **Fields inherited from class jminusminus.JExpression**
  `isStatementExpression, type`
- **Fields inherited from class jminusminus.JAST**
  `compilationUnit, line`
- **Constructor Summary**

Constructors

**Constructor and Description**

```
JNegateOp(int line, JExpression arg)
```

Construct an AST node for a negation expression given its line number, and the operand.

- **Method Summary**

Methods

| Modifier and Type | Method and Description |
|---|---|
| JExpression | **analyze**(Context context)<br>Analyzing the negation operation involves analyzing its operand, checking its type and determining the result type. |
| void | **codegen**(CLEmitter output)<br>Generating code for the negation operation involves generating code for the operand, and then the negation instruction. |

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

  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**

  - **JNegateOp**

```
publicJNegateOp(intline,
        JExpressionarg)
```

Construct an AST node for a negation expression given its line number, and the operand.

**Parameters:**

`line` - line in which the negation expression occurs in the source file.
`arg` - the operand.

- **Method Detail**

  - **analyze**

```
publicJExpressionanalyze(Contextcontext)
```

Analyzing the negation operation involves analyzing its operand, checking its type 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 negation operation involves generating code for the operand, and then the negation instruction.

**Specified by:**

`codegen` in class `JAST`

**Parameters:**

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