jminusminus

Class JLogicalAndOp

- java.lang.Object
  - jminusminus.JAST
    - jminusminus.JStatement
      - jminusminus.JExpression
        - jminusminus.JBinaryExpression
          - jminusminus.JBooleanBinaryExpression
            - jminusminus.JLogicalAndOp

.

```
class JLogicalAndOp
extends JBooleanBinaryExpression
```

The AST node for a logical AND (&&) 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

JLogicalAndOp(int line, JExpression lhs, JExpression rhs)
Construct an AST node for a logical AND expression given its line number, and lhs and rhs operands.

- **Method Summary**

Methods

Modifier and Type	Method and Description
JExpression	**analyze**(Context context) Analyzing a logical AND expression involves analyzing its operands and insuring they are boolean; the result type is of course boolean.
void	**codegen**(CLEmitter output, String targetLabel, boolean onTrue) The semantics of j-- require that we implement short-circuiting branching in implementing the logical AND.

- **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**

  - **JLogicalAndOp**

```
publicJLogicalAndOp(intline,
                    JExpressionlhs,
                    JExpressionrhs)
```

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

Parameters:

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

- **Method Detail**

  - **analyze**

```
publicJExpressionanalyze(Contextcontext)
```

Analyzing a logical AND expression involves analyzing its operands and insuring they are boolean; the result type is of course boolean.

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,
                  StringtargetLabel,
                  booleanonTrue)
```

The semantics of j-- require that we implement short-circuiting branching in implementing the logical AND.

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?