JavaScript is disabled on your browser.

jminusminus

Class JVariableDeclarator

- java.lang.Object
  - jminusminus.JAST
    - jminusminus.JVariableDeclarator

```
class JVariableDeclarator
extends JAST
```

The AST node for a variable declarator, which declares a name, its type and (possibly) provides an initialization.

- **Field Summary**

	Fields
Modifier and Type	**Field and Description**
boolean	**isInitialized**

- **Fields inherited from class jminusminus.JAST**

  compilationUnit, line

- **Constructor Summary**

Constructors
Constructor and Description
JVariableDeclarator(int line, String name, Type type, JExpression initializer)

Construct an AST node for a variable declarator given the line number, variable name, its type, and the initializer.

- **Method Summary**

| | Methods | |
| **Modifier and Type** | **Method and Description** | |

JVariableDeclarator	**analyze**(Context context) No analysis is done here.
void	**codegen**(CLEmitter output) No code generation is done here.
JExpression	**initializer**() Return the variable initializer.
String	**name**() Return the variable name.
void	**setInitializer**(JExpression initial) Set the variable initializer.
void	**setType**(Type type) Set the declarator's type.
Type	**type**() Return the variable type.
void	**writeToStdOut**(PrettyPrinter p) Write the information pertaining to this AST to STDOUT.

- **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

- **Field Detail**

  - **isInitialized**

  publicboolean isInitialized

- **Constructor Detail**

  - **JVariableDeclarator**

  publicJVariableDeclarator(intline,
                   Stringname,
                   Typetype,
                   JExpressioninitializer)

  Construct an AST node for a variable declarator given the line number, variable name, its type, and the initializer.

  Parameters:

  line - line in which the variable occurs in the source file.

name - variable name.
`type` - variable type.
`initializer` - initializer.

- **Method Detail**

  - **name**

  `public` `String` `name()`

  Return the variable name.
  Returns:
     the variable name.

  - **type**

  `public` `Type` `type()`

  Return the variable type.
  Returns:
     the variable type.

  - **setType**

  `public` `void` `setType(` `Type` `type)`

  Set the declarator's type.
  Parameters:
     `type` - the new type

  - **initializer**

  `public` `Expression` `initializer()`

  Return the variable initializer.
  Returns:
     the variable initializer.

  - **setInitializer**

  `public` `void` `setInitializer(` `JExpression` `initial)`

  Set the variable initializer.
  Parameters:
     `initial` - initializer.

  - **analyze**

  `public` `JVariableDeclarator` `analyze(` `Context` `context)`

  No analysis is done here.
  Specified by:
     `analyze` in class `JAST`
  Parameters:
     `context` - context in which names are resolved.
  Returns:
     the analyzed (and possibly rewritten) AST subtree.

  - **codegen**

  `public` `void` `codegen(` `CLEmitter` `output)`

  No code generation is done here.

Specified by:
    `codegen` in class `JAST`
Parameters:
    `output` - the code emitter (basically an abstraction for producing the .class file).

- **writeToStdOut**

`public void writeToStdOut(PrettyPrinter p)`

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.