jminusminus

Class JMethodDeclaration

- java.lang.Object
  - jminusminus.JAST
    - jminusminus.JMethodDeclaration

- All Implemented Interfaces:
  JMember
  Direct Known Subclasses:
  JConstructorDeclaration

```
class JMethodDeclaration
extends JAST
implements JMember
```

The AST node for a method declaration.

- **Field Summary**

Fields

Modifier and Type	Field and Description
protected JBlock	**body** Method body.
protected MethodContext	**context** Built in analyze().
protected String	**descriptor** Computed by preAnalyze().
protected boolean	**isAbstract** Is method abstract.

protected boolean	**isPrivate** Is method private.
protected boolean	**isStatic** Is method static.
protected ArrayList<String>	**mods** Method modifiers.
protected String	**name** Method name.
protected ArrayList<JFormalParameter>	**params** The formal parameters.
protected Type	**returnType** Return type.

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

  compilationUnit, line

- **Constructor Summary**

Constructors

Constructor and Description
JMethodDeclaration(int line, ArrayList<String> mods, String name, Type returnType, ArrayList<JFormalParameter> params, JBlock body) Construct an AST node for a method declaration given the line number, method name, return type, formal parameters, and the method body.

- **Method Summary**

Methods

Modifier and Type	Method and Description
JAST	**analyze**(Context context) Analysis for a method declaration involves (1) creating a new method context (that records the return type; this is used in the analysis of the method body), (2) bumping up the offset (for instance methods), (3) declaring the formal parameters in the method context, and (4) analyzing the method's body.
void	**codegen**(CLEmitter output) Generate code for the method declaration.
void	**partialCodegen**(Context context, CLEmitter partial) Add this method declaration to the partial class.
void	**preAnalyze**(Context context, CLEmitter partial) Declare this method in the parent (class) context.
void	**writeToStdOut**(PrettyPrinter p) Write the information pertaining to this AST to STDOUT.

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

  line

- **Methods inherited from class java.lang.Object**

  `clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- **Field Detail**

  - **mods**

  ```
  protectedArrayList<String> mods
  ```

  Method modifiers.

  - **name**

  ```
  protectedString name
  ```

  Method name.

  - **returnType**

  ```
  protectedType returnType
  ```

  Return type.

  - **params**

  ```
  protectedArrayList<JFormalParameter> params
  ```

  The formal parameters.

  - **body**

  ```
  protectedJBlock body
  ```

  Method body.

  - **context**

  ```
  protectedMethodContext context
  ```

  Built in analyze().

  - **descriptor**

  ```
  protectedString descriptor
  ```

  Computed by preAnalyze().

  - **isAbstract**

  ```
  protectedboolean isAbstract
  ```

  Is method abstract.

  - **isStatic**

  ```
  protectedboolean isStatic
  ```

  Is method static.

  - **isPrivate**

  ```
  protectedboolean isPrivate
  ```

  Is method private.

- ## Constructor Detail

  - ### JMethodDeclaration

```
publicJMethodDeclaration(intline,
                         ArrayList<String>mods,
                         Stringname,
                         TypereturnType,
                         ArrayList<JFormalParameter>params,
                         JBlockbody)
```

Construct an AST node for a method declaration given the line number, method name, return type, formal parameters, and the method body.

Parameters:

`line` - line in which the method declaration occurs in the source file.
`mods` - modifiers.
`name` - method name.
`returnType` - return type.
`params` - the formal parameters.
`body` - method body.

- ## Method Detail

  - ### preAnalyze

```
publicvoidpreAnalyze(Contextcontext,
                     CLEmitterpartial)
```


Declare this method in the parent (class) context.

Specified by:

`preAnalyze` in interface `JMember`

Parameters:

`context` - the parent (class) context.
`partial` - the code emitter (basically an abstraction for producing the partial class).

  - ### analyze

```
publicJASTanalyze(Contextcontext)
```

Analysis for a method declaration involves (1) creating a new method context (that records the return type; this is used in the analysis of the method body), (2) bumping up the offset (for instance methods), (3) declaring the formal parameters in the method context, and (4) analyzing the method's body.

Specified by:

`analyze` in class `JAST`

Parameters:

`context` - context in which names are resolved.

Returns:

the analyzed (and possibly rewritten) AST subtree.

  - ### partialCodegen

```
publicvoidpartialCodegen(Contextcontext,
                         CLEmitterpartial)
```

Add this method declaration to the partial class.

Overrides:

`partialCodegen` in class `JAST`

Parameters:

`context` - the parent (class) context.
`partial` - the code emitter (basically an abstraction for producing the partial class).

- **codegen**

```
publicvoidcodegen(CLEmitteroutput)
```

Generate code for the method declaration.

Specified by:

`codegen` in class `JAST`

Parameters:

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

- **writeToStdOut**

```
publicvoidwriteToStdOut(PrettyPrinterp)
```

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.