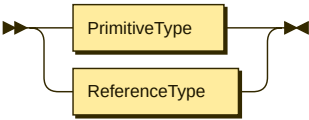


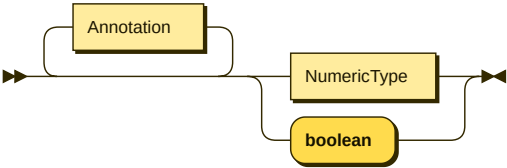
Type:



Type ::= PrimitiveType
| ReferenceType

no references

PrimitiveType:

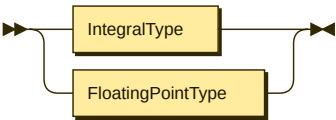


PrimitiveType ::= Annotation* (NumericType | 'boolean')

referenced by:

- [ArrayCreationExpression](#)
- [ArrayType](#)
- [CastExpression](#)
- [Type](#)

NumericType:

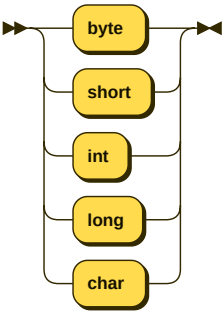


NumericType ::= IntegralType
| FloatingPointType

referenced by:

- [ClassLiteral](#)
- [PrimitiveType](#)
- [UnannPrimitiveType](#)

IntegralType:

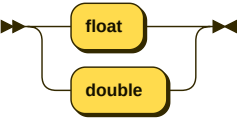


IntegralType ::= 'byte'
| 'short'
| 'int'
| 'long'
| 'char'

referenced by:

- [NumericType](#)

FloatingPointType:

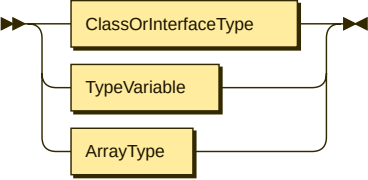


FloatingPointType ::= 'float' | 'double'

referenced by:

- [NumericType](#)

ReferenceType:

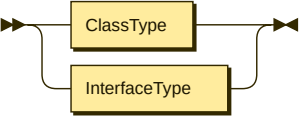


ReferenceType ::= ClassOrInterfaceType | TypeVariable | ArrayType

referenced by:

- [CastExpression](#)
- [EqualityExpression](#)
- [MethodReference](#)
- [Type](#)
- [TypeArgument](#)
- [WildcardBounds](#)

ClassOrInterfaceType:

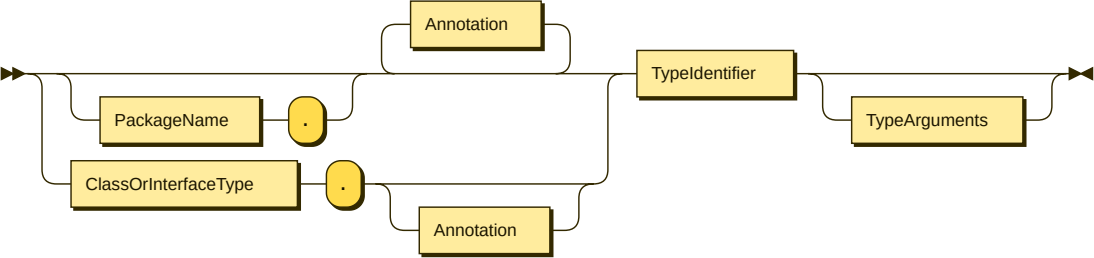


ClassOrInterfaceType ::= ClassType | InterfaceType

referenced by:

- [ArrayCreationExpression](#)
- [ArrayType](#)
- [ClassType](#)
- [ReferenceType](#)
- [TypeBound](#)

ClassType:

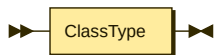


ClassType ::= ((PackageName '.')? Annotation* | ClassOrInterfaceType '.' Annotation?) TypeIdentifier TypeArguments?

referenced by:

- [CatchType](#)

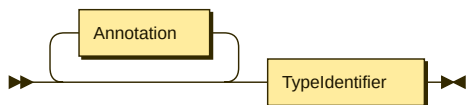
- [ClassOrInterfaceType](#)
- [ExceptionType](#)
- [InterfaceType](#)
- [MethodReference](#)
- [Superclass](#)

InterfaceType:

InterfaceType
 ::= ClassType

referenced by:

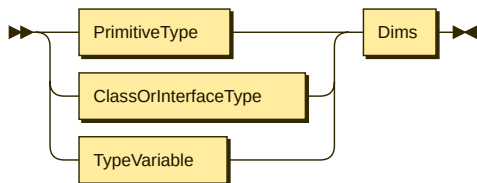
- [ClassOrInterfaceType](#)
- [InterfaceTypeList](#)
- [TypeBound](#)

TypeVariable:

TypeVariable
 ::= Annotation* TypeIdentifier

referenced by:

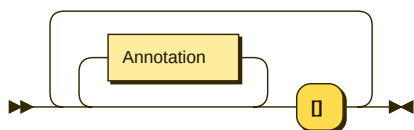
- [ArrayType](#)
- [ExceptionType](#)
- [ReferenceType](#)
- [TypeBound](#)

ArrayType:

ArrayType
 ::= (PrimitiveType | ClassOrInterfaceType | TypeVariable) Dims

referenced by:

- [MethodReference](#)
- [ReferenceType](#)

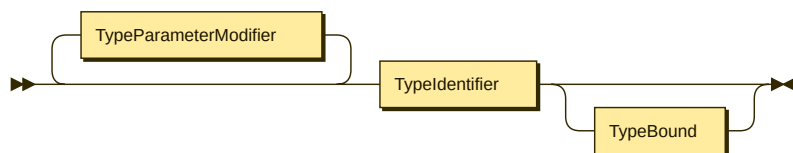
Dims:

Dims
 ::= (Annotation* '[]')+

referenced by:

- [AnnotationTypeElementDeclaration](#)
- [ArrayCreationExpression](#)
- [ArrayType](#)
- [MethodDeclarator](#)
- [UnannArrayType](#)
- [VariableDeclaratorId](#)

TypeParameter:

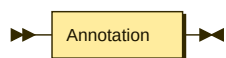


TypeParameter
 ::= TypeParameterModifier* TypeIdentifier TypeBound?

referenced by:

- [TypeParameterList](#)

TypeParameterModifier:

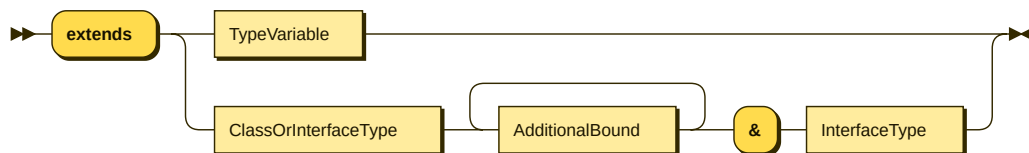


TypeParameterModifier
 ::= Annotation

referenced by:

- [TypeParameter](#)

TypeBound:

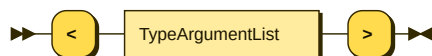


TypeBound
 ::= 'extends' (TypeVariable | ClassOrInterfaceType AdditionalBound+ '&' InterfaceType)

referenced by:

- [TypeParameter](#)

TypeArguments:

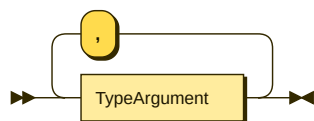


TypeArguments
 ::= '<' TypeArgumentList '>'

referenced by:

- [ClassType](#)
- [ExplicitConstructorInvocation](#)
- [MethodInvocation](#)
- [MethodReference](#)
- [TypeArgumentsOrDiamond](#)
- [UnannClassType](#)
- [UnqualifiedClassInstanceCreationExpression](#)

TypeArgumentList:

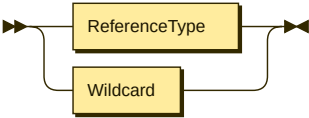


TypeArgumentList
 ::= TypeArgument (',' TypeArgument)*

referenced by:

- [TypeArguments](#)

TypeArgument:

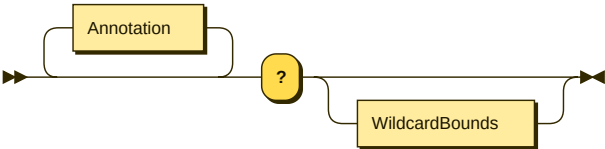


TypeArgument ::= ReferenceType
 | Wildcard

referenced by:

- [TypeArgumentList](#)

Wildcard:

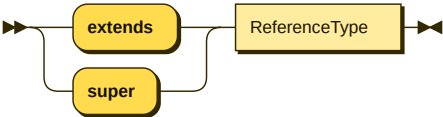


Wildcard ::= Annotation* '?' WildcardBounds?

referenced by:

- [TypeArgument](#)

WildcardBounds:

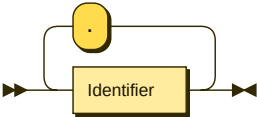


WildcardBounds ::= ('extends' | 'super') ReferenceType

referenced by:

- [Wildcard](#)

ModuleName:

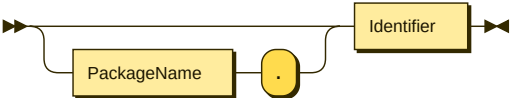


ModuleName ::= Identifier ('.' Identifier)*

referenced by:

- [ModuleDirective](#)

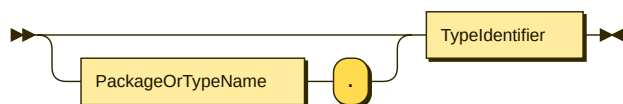
PackkageName:



PackkageName ::= (PackageName '.')? Identifier

no references

TypeName:

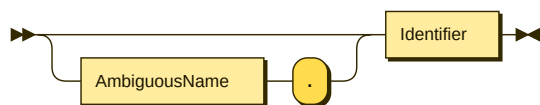


`TypeName ::= (PackageOrTypeName '.')? TypeIdentifier`

referenced by:

- [ClassLiteral](#)
- [FieldAccess](#)
- [MarkerAnnotation](#)
- [MethodInvocation](#)
- [MethodReference](#)
- [ModuleDirective](#)
- [NormalAnnotation](#)
- [PrimaryNoNewArray](#)
- [SingleElementAnnotation](#)
- [SingleStaticImportDeclaration](#)
- [SingleTypeImportDeclaration](#)
- [StaticImportOnDemandDeclaration](#)

ExpressionName:

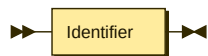


`ExpressionName
::= (AmbiguousName '.')? Identifier`

referenced by:

- [ArrayAccess](#)
- [ClassInstanceCreationExpression](#)
- [ExplicitConstructorInvocation](#)
- [LeftHandSide](#)
- [MethodInvocation](#)
- [MethodReference](#)

MethodName:

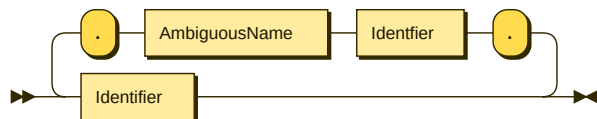


`MethodName
::= Identifier`

referenced by:

- [MethodInvocation](#)

PackageOrTypeName:

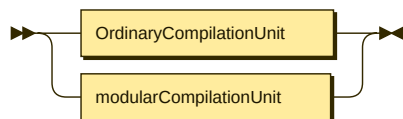


`PackageOrTypeName
::= Identifier ('.' Identifier AmbiguousName '.' Identifier)*`

referenced by:

- [TypeImportOnDemandDeclaration](#)
- [TypeName](#)

CompilationUnit:

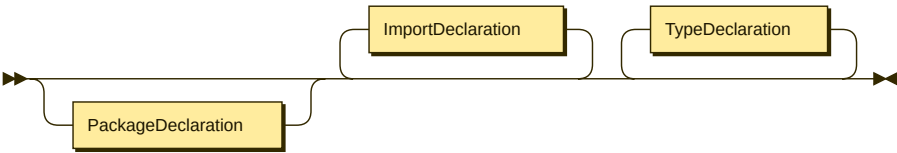


`CompilationUnit`

::= OrdinaryCompilationUnit
 | modularCompilationUnit

no references

OrdinaryCompilationUnit:

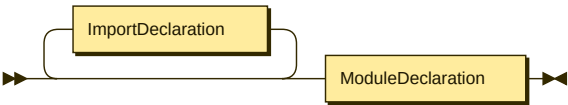


OrdinaryCompilationUnit ::= PackageDeclaration? ImportDeclaration* TypeDeclaration*

referenced by:

- [CompilationUnit](#)

modularCompilationUnit:

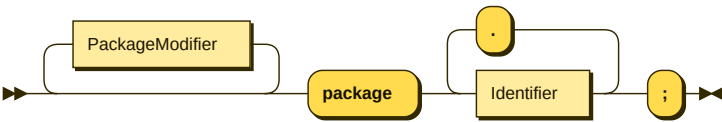


modularCompilationUnit ::= ImportDeclaration* ModuleDeclaration

referenced by:

- [CompilationUnit](#)

PackageDeclaration:

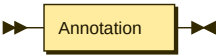


PackageDeclaration ::= PackageModifier* 'package' Identifier ('.' Identifier)* ';'

referenced by:

- [OrdinaryCompilationUnit](#)

PackageModifier:

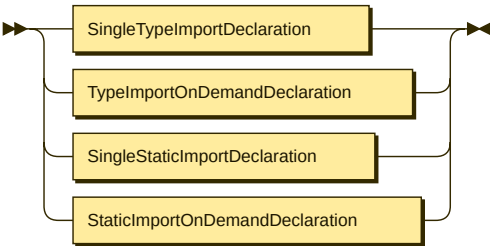


PackageModifier ::= Annotation

referenced by:

- [PackageDeclaration](#)

ImportDeclaration:



```

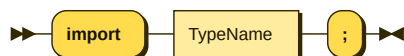
ImportDeclaration
  ::= SingleTypeImportDeclaration
     | TypeImportOnDemandDeclaration
     | SingleStaticImportDeclaration
     | StaticImportOnDemandDeclaration

```

referenced by:

- [OrdinaryCompilationUnit](#)
- [modularCompilationUnit](#)

SingleTypeImportDeclaration:



```

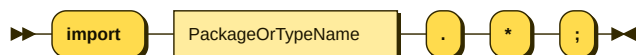
SingleTypeImportDeclaration
  ::= 'import' TypeName ';'

```

referenced by:

- [ImportDeclaration](#)

TypeImportOnDemandDeclaration:



```

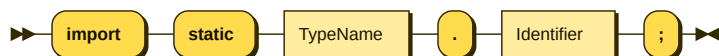
TypeImportOnDemandDeclaration
  ::= 'import' PackageOrTypeName '.' '*' ';'

```

referenced by:

- [ImportDeclaration](#)

SingleStaticImportDeclaration:



```

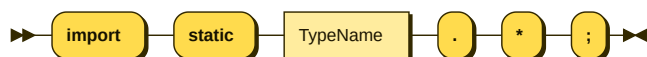
SingleStaticImportDeclaration
  ::= 'import' 'static' TypeName '.' Identifier ';'

```

referenced by:

- [ImportDeclaration](#)

StaticImportOnDemandDeclaration:



```

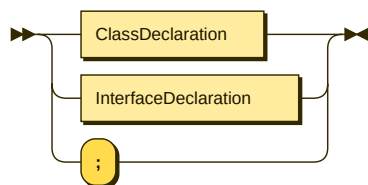
StaticImportOnDemandDeclaration
  ::= 'import' 'static' TypeName '.' '*' ';'

```

referenced by:

- [ImportDeclaration](#)

TypeDeclaration:



```

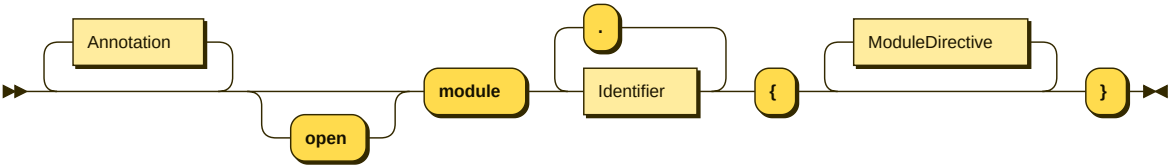
TypeDeclaration
  ::= ClassDeclaration
     | InterfaceDeclaration
     | ';'

```


referenced by:

- [OrdinaryCompilationUnit](#)

ModuleDeclaration:

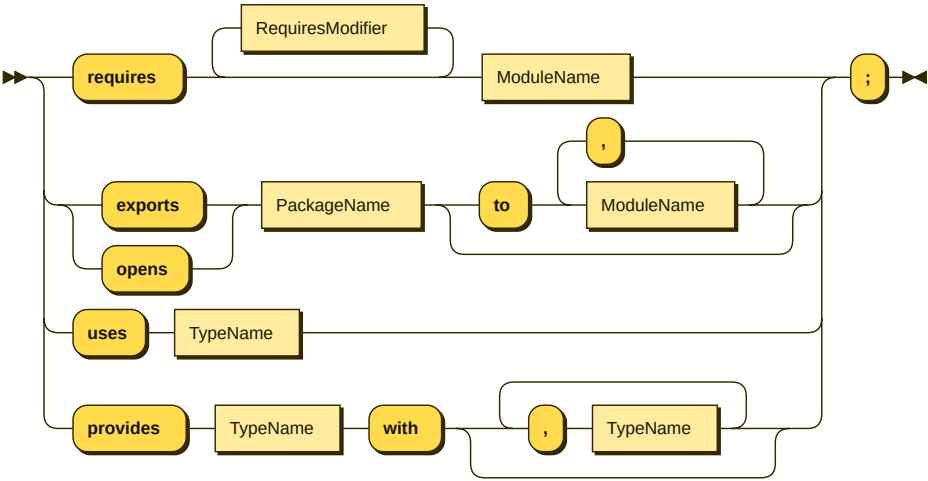


```
ModuleDeclaration ::= Annotation* 'open'? 'module' Identifier ( '.' Identifier )* '{' ModuleDirective* '}'
```

referenced by:

- [modularCompilationUnit](#)

ModuleDirective:

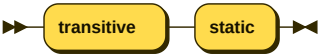


```
ModuleDirective ::= ( 'requires' RequiresModifier* ModuleName | ( 'exports' | 'opens' ) PackageName ( 'to' ModuleName ( ',' ModuleName )* )? | 'uses' TypeName | 'provides' TypeName 'with' ( ',' TypeName )* ) ';' ;
```

referenced by:

- [ModuleDeclaration](#)

RequiresModifier:

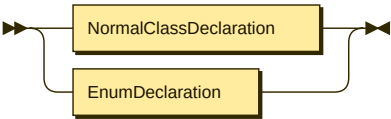


```
RequiresModifier ::= 'transitive' 'static'
```

referenced by:

- [ModuleDirective](#)

ClassDeclaration:

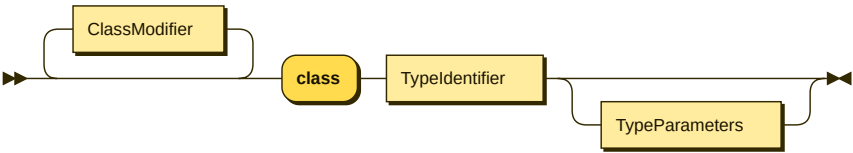


```
ClassDeclaration ::= NormalClassDeclaration | EnumDeclaration
```

referenced by:

- [AnnotationTypeBody](#)
- [BlockStatement](#)
- [ClassMemberDeclaration](#)
- [InterfaceMemberDeclaration](#)
- [TypeDeclaration](#)

NormalClassDeclaration:

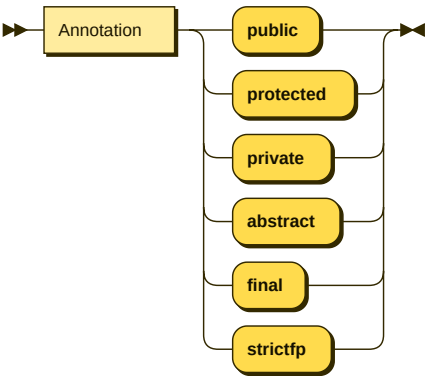


NormalClassDeclaration ::= ClassModifier* 'class' TypeIdentifier TypeParameters?

referenced by:

- [ClassDeclaration](#)

ClassModifier:



ClassModifier ::= Annotation ('public' | 'protected' | 'private' | 'abstract' | 'final' | 'strictfp')

referenced by:

- [EnumDeclaration](#)
- [NormalClassDeclaration](#)

TypeParameters:

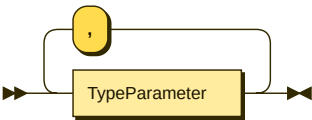


TypeParameters ::= '<' TypeParameterList '>'

referenced by:

- [ConstructorDeclarator](#)
- [MethodHeader](#)
- [NormalClassDeclaration](#)

TypeParameterList:

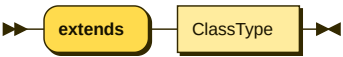


TypeParameterList ::= TypeParameter (',' TypeParameter)*

referenced by:

- [TypeParameters](#)

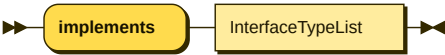
Superclass:



Superclass ::= 'extends' ClassType

no references

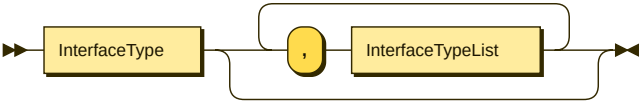
Superinterfaces:



Superinterfaces ::= 'implements' InterfaceTypeList

no references

InterfaceTypeList:

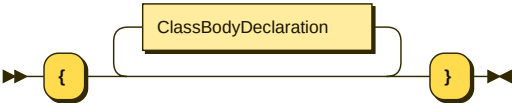


InterfaceTypeList ::= InterfaceType (',' InterfaceTypeList)*

referenced by:

- [ExtendsInterfaces](#)
- [InterfaceTypeList](#)
- [Superinterfaces](#)

ClassBody:

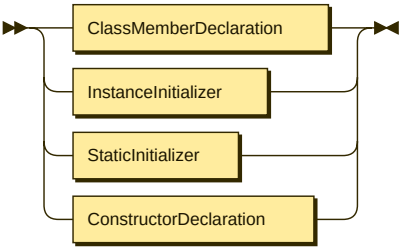


ClassBody ::= '{' ClassBodyDeclaration* '}'

referenced by:

- [EnumConstant](#)
- [UnqualifiedClassInstanceCreationExpression](#)

ClassBodyDeclaration:

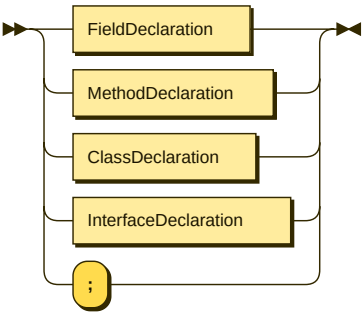


ClassBodyDeclaration ::= ClassMemberDeclaration | InstanceInitializer | StaticInitializer | ConstructorDeclaration

referenced by:

- [ClassBody](#)
- [EnumBodyDeclarations](#)

ClassMemberDeclaration:

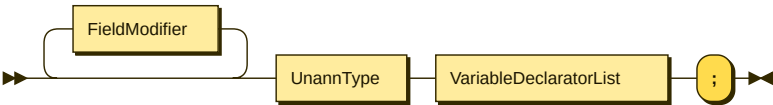


```
ClassMemberDeclaration
  ::= FieldDeclaration
  | MethodDeclaration
  | ClassDeclaration
  | InterfaceDeclaration
  | ';'
  ;
```

referenced by:

- [ClassBodyDeclaration](#)

FieldDeclaration:

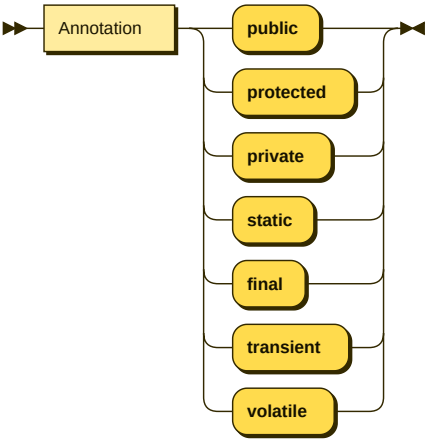


```
FieldDeclaration
  ::= FieldModifier* UnannType VariableDeclaratorList ';'
  ;
```

referenced by:

- [ClassMemberDeclaration](#)

FieldModifier:

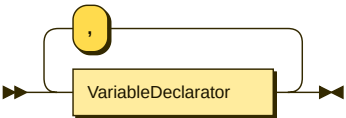


```
FieldModifier
  ::= Annotation ( 'public' | 'protected' | 'private' | 'static' | 'final' | 'transient' | 'volatile' )
  ;
```

referenced by:

- [FieldDeclaration](#)

VariableDeclaratorList:

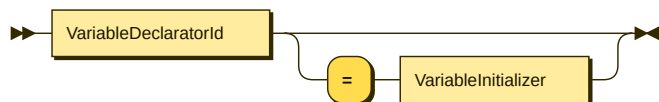


```
VariableDeclaratorList
    ::= VariableDeclarator ( ',' VariableDeclarator )*
```

referenced by:

- [ConstantDeclaration](#)
- [FieldDeclaration](#)
- [LocalVariableDeclaration](#)

VariableDeclarator:

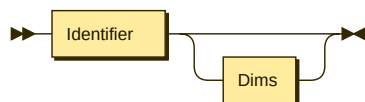


```
VariableDeclarator
    ::= VariableDeclaratorId ( '=' VariableInitializer )?
```

referenced by:

- [VariableDeclaratorList](#)

VariableDeclaratorId:

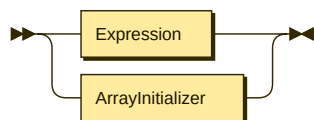


```
VariableDeclaratorId
    ::= Identifier Dims?
```

referenced by:

- [CatchFormalParameter](#)
- [EnhancedForStatement](#)
- [EnhancedForStatementNoShortIf](#)
- [FormalParameter](#)
- [LambdaParameter](#)
- [VariableDeclarator](#)

VariableInitializer:

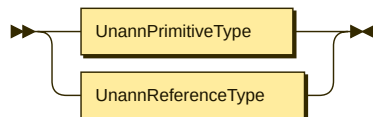


```
VariableInitializer
    ::= Expression
    | ArrayInitializer
```

referenced by:

- [VariableDeclarator](#)
- [VariableInitializerList](#)

UnannType:



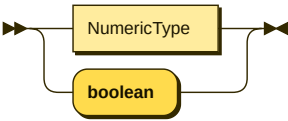
```
UnannType
    ::= UnannPrimitiveType
    | UnannReferenceType
```

referenced by:

- [AnnotationTypeElementDeclaration](#)
- [ConstantDeclaration](#)
- [FieldDeclaration](#)
- [FormalParameter](#)
- [LambdaParameterType](#)
- [LocalVariableType](#)

- [ReceiverParameter](#)
- [Result](#)
- [VariableArityParameter](#)

UnannPrimitiveType:

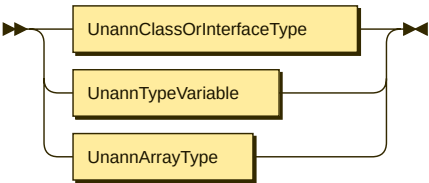


UnannPrimitiveType
 ::= NumericType
 | 'boolean'

referenced by:

- [UnannArrayType](#)
- [UnannType](#)

UnannReferenceType:

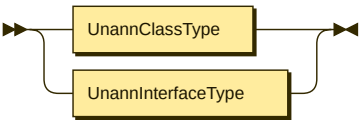


UnannReferenceType
 ::= UnannClassOrInterfaceType
 | UnannTypeVariable
 | UnannArrayType

referenced by:

- [UnannType](#)

UnannClassOrInterfaceType:

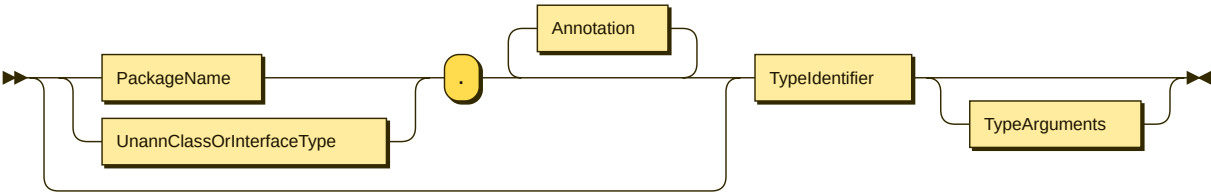


UnannClassOrInterfaceType
 ::= UnannClassType
 | UnannInterfaceType

referenced by:

- [UnannArrayType](#)
- [UnannClassType](#)
- [UnannReferenceType](#)

UnannClassType:

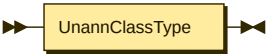


UnannClassType
 ::= ((PackageName | UnannClassOrInterfaceType) '.' Annotation*)? TypeIdentifier TypeArguments?

referenced by:

- [CatchType](#)
- [UnannClassOrInterfaceType](#)
- [UnannInterfaceType](#)

UnannInterfaceType:



UnannInterfaceType ::= UnannClassType

referenced by:

- [UnannClassOrInterfaceType](#)

UnannTypeVariable:

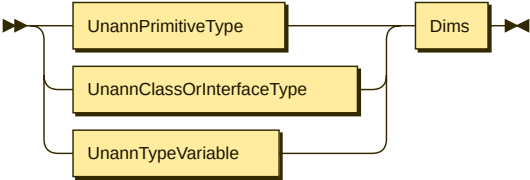


UnannTypeVariable ::= TypeIdentifier

referenced by:

- [UnannArrayType](#)
- [UnannReferenceType](#)

UnannArrayType:

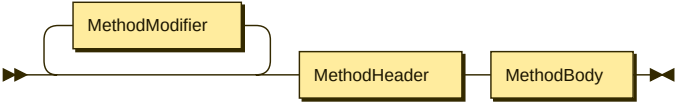


UnannArrayType ::= (UnannPrimitiveType | UnannClassOrInterfaceType | UnannTypeVariable) Dims

referenced by:

- [UnannReferenceType](#)

MethodDeclaration:

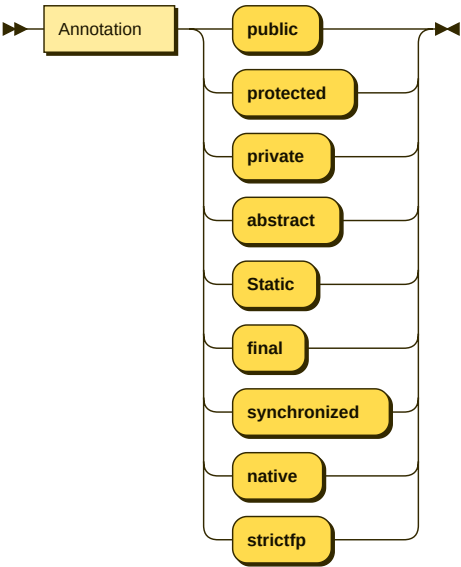


MethodDeclaration ::= MethodModifier* MethodHeader MethodBody

referenced by:

- [ClassMemberDeclaration](#)

MethodModifier:

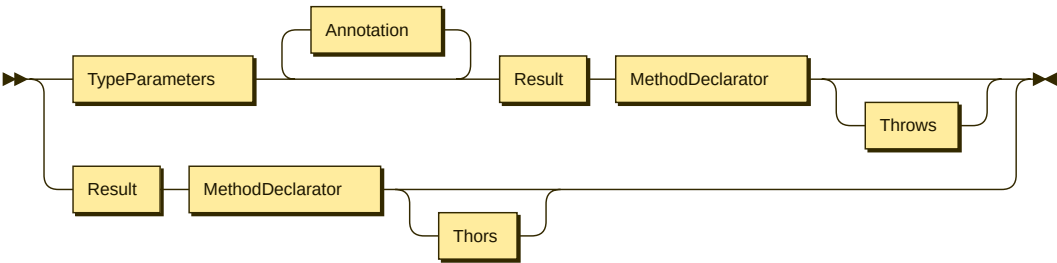


```
MethodModifier ::= Annotation ( 'public' | 'protected' | 'private' | 'abstract' | 'Static' | 'final' | 'synchronized' | 'native' | 'strictfp' )
```

referenced by:

- [MethodDeclaration](#)

MethodHeader:

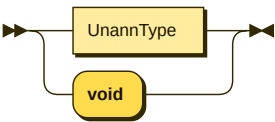


```
MethodHeader ::= Result MethodDeclarator Thors? | TypeParameters Annotation* Result MethodDeclarator Throws?
```

referenced by:

- [InterfaceMethodDeclaration](#)
- [MethodDeclaration](#)

Result:

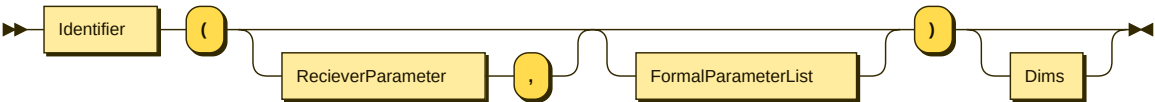


```
Result ::= UnannType | 'void'
```

referenced by:

- [MethodHeader](#)

MethodDeclarator:



```
MethodDeclarator
```

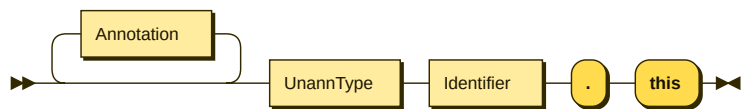


```
 ::= Identifier '(' ( RecieverParameter ',' )? FormalParameterList? ')' Dims?
```

referenced by:

- [MethodHeader](#)

RecieverParameter:

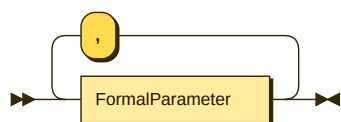


```
RecieverParameter
 ::= Annotation* UnannType Identifier '.' 'this'
```

referenced by:

- [ConstructorDeclarator](#)
- [MethodDeclarator](#)

FormalParameterList:

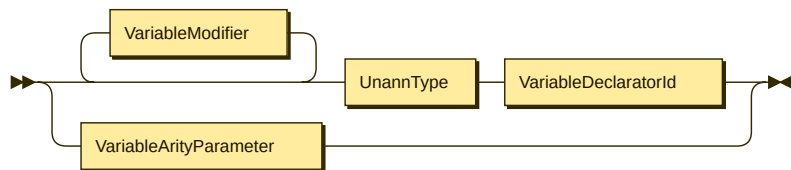


```
FormalParameterList
 ::= FormalParameter ( ',' FormalParameter )*
```

referenced by:

- [ConstructorDeclarator](#)
- [MethodDeclarator](#)

FormalParameter:

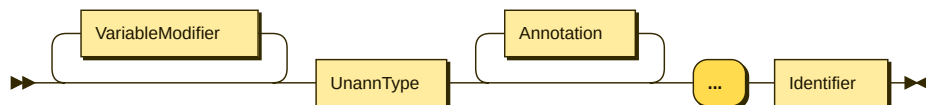


```
FormalParameter
 ::= VariableModifier* UnannType VariableDeclaratorId
 | VariableArityParameter
```

referenced by:

- [FormalParameterList](#)

VariableArityParameter:

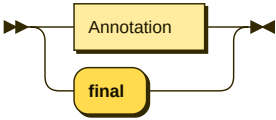


```
VariableArityParameter
 ::= VariableModifier* UnannType Annotation* '...' Identifier
```

referenced by:

- [FormalParameter](#)
- [LambdaParameter](#)

VariableModifier:

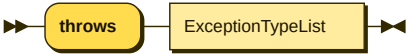


VariableModifier ::= Annotation | 'final'

referenced by:

- [CatchFormalParameter](#)
- [EnhancedForStatement](#)
- [EnhancedForStatementNoShortIf](#)
- [FormalParameter](#)
- [LambdaParameter](#)
- [LocalVariableDeclaration](#)
- [Resource](#)
- [VariableArityParameter](#)

Throws:

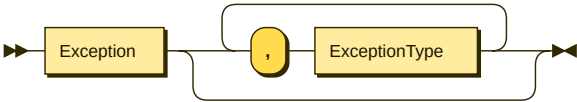


Throws ::= 'throws' ExceptionTypeList

referenced by:

- [ConstructorDeclaration](#)
- [MethodHeader](#)

ExceptionTypeList:

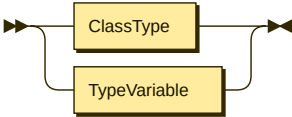


ExceptionTypeList ::= Exception (',' ExceptionType)*

referenced by:

- [Throws](#)

ExceptionType:

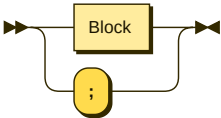


ExceptionType ::= ClassType | TypeVariable

referenced by:

- [ExceptionTypeList](#)

MethodBody:



MethodBody ::= Block | ';'

referenced by:

- [InterfaceMethodDeclaration](#)

- [MethodDeclaration](#)

InstanceInitializer:



```
InstanceInitializer ::= 'static' Block
```

referenced by:

- [ClassBodyDeclaration](#)

StaticInitializer:

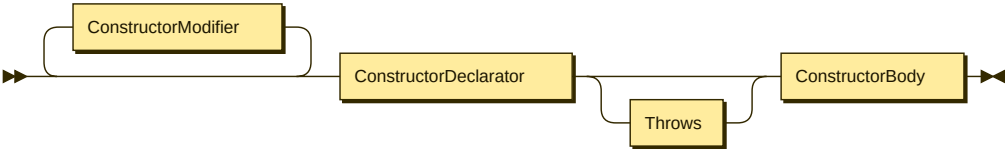


```
StaticInitializer ::= 'static' Block
```

referenced by:

- [ClassBodyDeclaration](#)

ConstructorDeclaration:

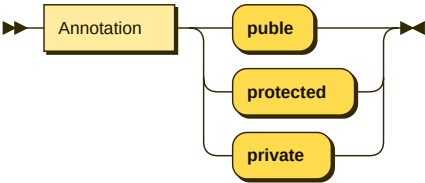


```
ConstructorDeclaration ::= ConstructorModifier* ConstructorDeclarator Throws? ConstructorBody
```

referenced by:

- [ClassBodyDeclaration](#)

ConstructorModifier:

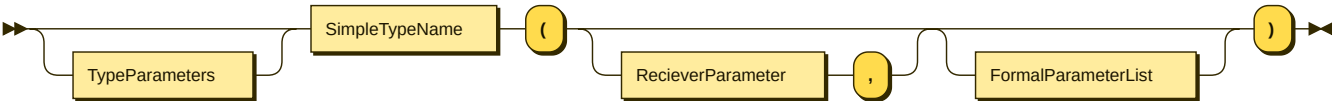


```
ConstructorModifier ::= Annotation ( 'public' | 'protected' | 'private' )
```

referenced by:

- [ConstructorDeclaration](#)

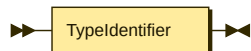
ConstructorDeclarator:



```
ConstructorDeclarator ::= TypeParameters? SimpleTypeName '(' ( ReceiverParameter ',' )? FormalParameterList? ')'
```

referenced by:

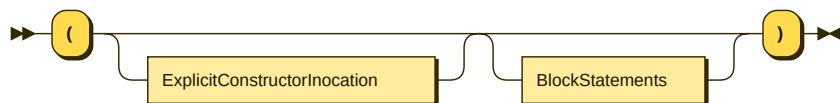
- [ConstructorDeclaration](#)

SimpleTypeName:

SimpleTypeName
::= TypeIdentifier

referenced by:

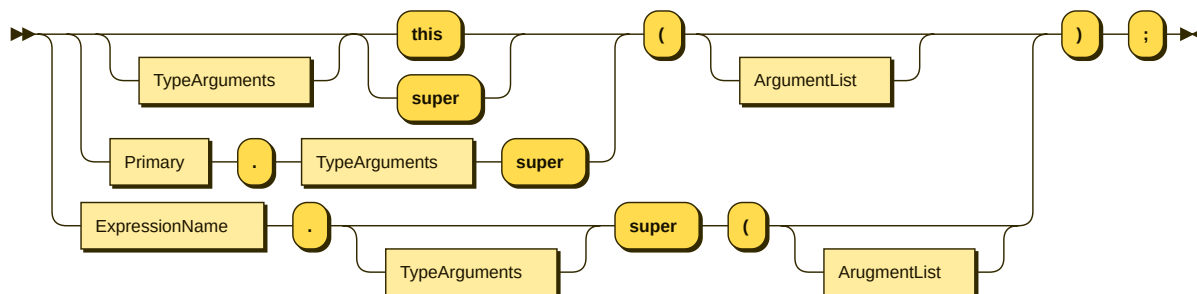
- [ConstructorDeclarator](#)

ConstructorBody:

ConstructorBody
::= '(' ExplicitConstructorInocation? BlockStatements? ')'

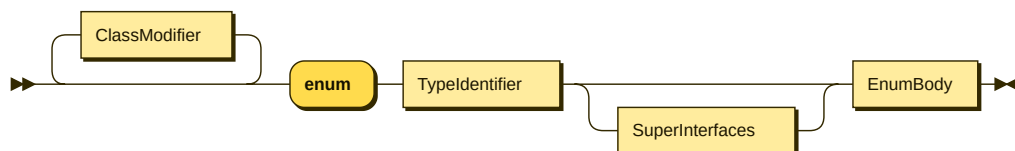
referenced by:

- [ConstructorDeclaration](#)

ExplicitConstructorInvocation:

ExplicitConstructorInvocation
::= ((TypeArguments? ('this' | 'super') | Primary '.' TypeArguments 'super') '(' ArgumentList? | ExpressionName '.' TypeArguments? 'super' '(' ArugmentList?)) ';' ;

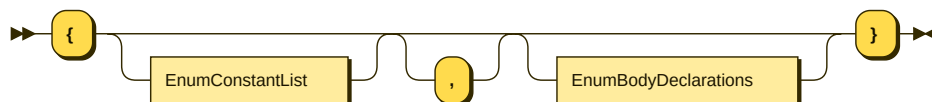
no references

EnumDeclaration:

EnumDeclaration
::= ClassModifier* 'enum' TypeIdentifier SuperInterfaces? EnumBody

referenced by:

- [ClassDeclaration](#)

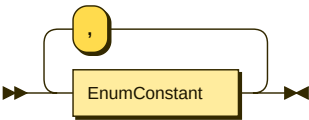
EnumBody:

EnumBody ::= '{' EnumConstantList? ','? EnumBodyDeclarations? '}'

referenced by:

- [EnumDeclaration](#)

EnumConstantList:

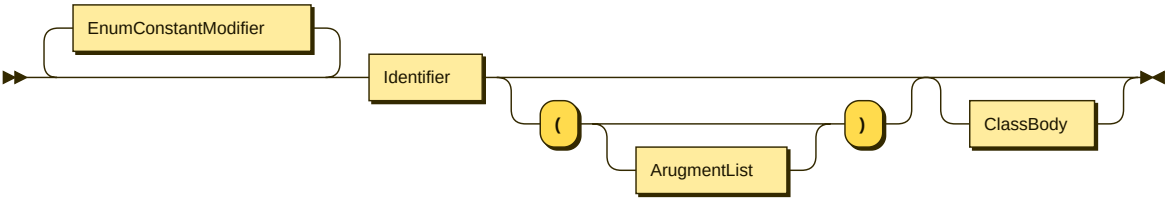


EnumConstantList ::= EnumConstant (',' EnumConstant) *

referenced by:

- EnumBody.

EnumConstant:

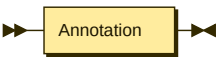


EnumConstant ::= EnumConstantModifier* Identifier ('(' ArugmentList? ')')? ClassBody?

referenced by:

- EnumConstantList

EnumConstantModifier:

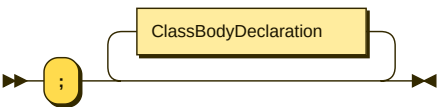


EnumConstantModifier ::= Annotation

referenced by:

- EnumConstant

EnumBodyDeclarations:

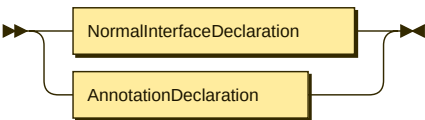


EnumBodyDeclarations ::= ';' ClassBodyDeclaration *

referenced by:

- EnumBody.

InterfaceDeclaration:



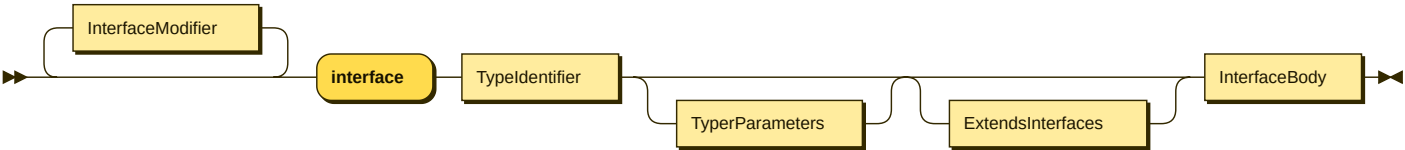
InterfaceDeclaration ::= NormalInterfaceDeclaration | AnnotationDeclaration

referenced by:

- ClassMemberDeclaration
- InterfaceMemberDeclaration

- [TypeDeclaration](#)

NormalInterfaceDeclaration:

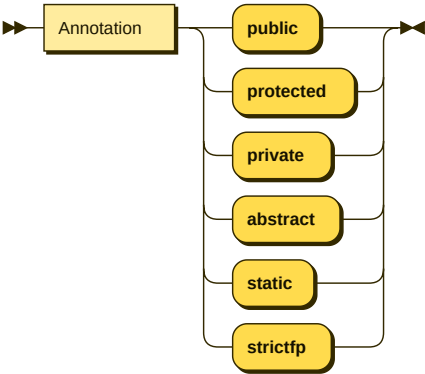


```
NormalInterfaceDeclaration ::= InterfaceModifier* 'interface' TypeIdentifier TyperParameters? ExtendsInterfaces? InterfaceBody
```

referenced by:

- [InterfaceDeclaration](#)

InterfaceModifier:



```
InterfaceModifier ::= Annotation ( 'public' | 'protected' | 'private' | 'abstract' | 'static' | 'strictfp' )
```

referenced by:

- [AnnotationTypeDeclaration](#)
- [NormalInterfaceDeclaration](#)

ExtendsInterfaces:

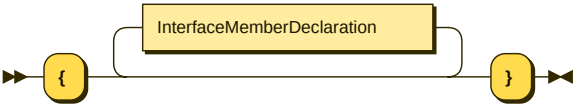


```
ExtendsInterfaces ::= 'extends' InterfaceTypeList
```

referenced by:

- [NormalInterfaceDeclaration](#)

InterfaceBody:

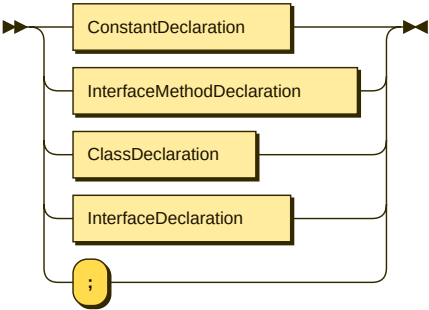


```
InterfaceBody ::= '{' InterfaceMemberDeclaration* '}'
```

referenced by:

- [NormalInterfaceDeclaration](#)

InterfaceMemberDeclaration:

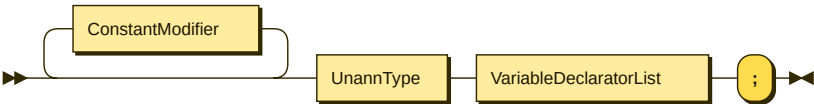


```
InterfaceMemberDeclaration ::= ConstantDeclaration | InterfaceMethodDeclaration | ClassDeclaration | InterfaceDeclaration | ';' ;
```

referenced by:

- [InterfaceBody](#).

ConstantDeclaration:

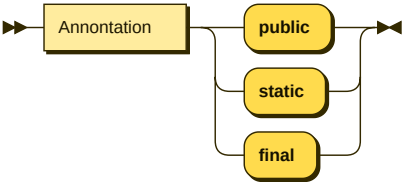


```
ConstantDeclaration ::= ConstantModifier* UnannType VariableDeclaratorList ';' ;
```

referenced by:

- [AnnotationTypeBody](#).
- [InterfaceMemberDeclaration](#)

ConstantModifier:

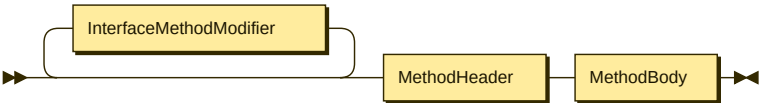


```
ConstantModifier ::= Annotation ( 'public' | 'static' | 'final' )
```

referenced by:

- [ConstantDeclaration](#)

InterfaceMethodDeclaration:

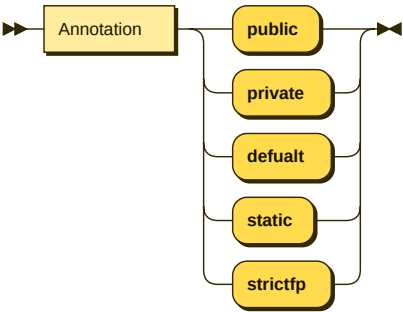


```
InterfaceMethodDeclaration ::= InterfaceMethodModifier* MethodHeader MethodBody
```

referenced by:

- [InterfaceMemberDeclaration](#)

InterfaceMethodModifier:

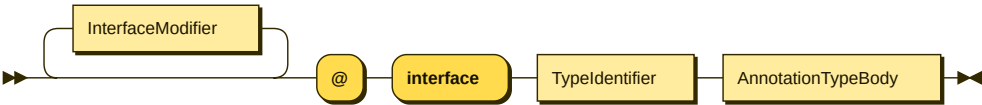


```
InterfaceMethodModifier ::= Annotation ( 'public' | 'private' | 'default' | 'static' | 'strictfp' )
```

referenced by:

- [InterfaceMethodDeclaration](#)

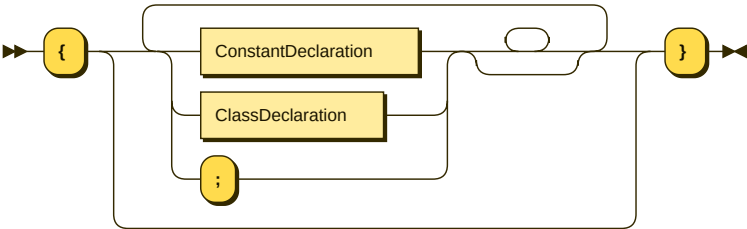
AnnotationTypeDeclaration:



```
AnnotationTypeDeclaration ::= InterfaceModifier* '@' 'interface' TypeIdentifier AnnotationTypeBody
```

no references

AnnotationTypeBody:



```
AnnotationTypeBody ::= '{' ( ( ConstantDeclaration | ClassDeclaration | ';' ) ( )* )* '}'
```

referenced by:

- [AnnotationTypeDeclaration](#)

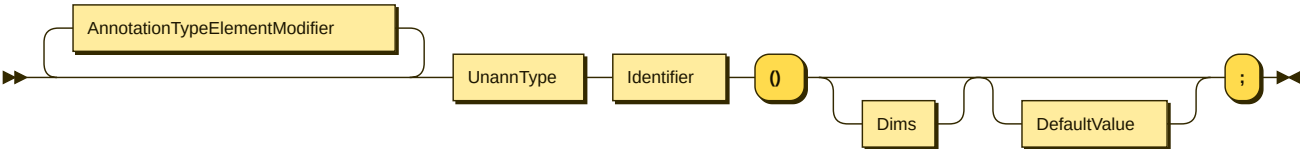
AnnotantionTypeElementDeclaration:



```
AnnotantionTypeElementDeclaration ::=
```

no references

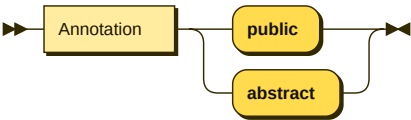
AnnotationTypeElementDeclaration:



```
AnnotationTypeElementDeclaration ::= AnnotationTypeElementModifier* UnannType Identifier '()' Dims? DefaultValue? ';' ;'
```

no references

AnnotationTypeElementModifier:

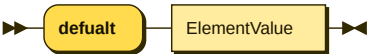


```
AnnotationTypeElementModifier
    ::= Annotation ( 'public' | 'abstract' )
```

referenced by:

- [AnnotationTypeElementDeclaration](#)

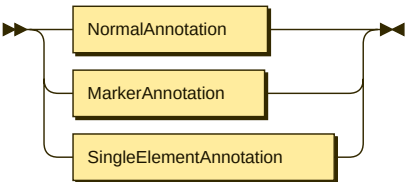
DefaultValue:



```
DefaultValue
    ::= 'default' ElementValue
```

no references

Annotation:

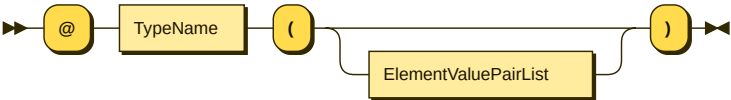


```
Annotation
    ::= NormalAnnotation
       | MarkerAnnotation
       | SingleElementAnnotation
```

referenced by:

- [AnnotationTypeElementModifier](#)
- [ClassModifier](#)
- [ClassOrInterfaceTypeToInstantiate](#)
- [ClassType](#)
- [ConstructorModifier](#)
- [DimExpr](#)
- [Dims](#)
- [ElementValue](#)
- [EnumConstantModifier](#)
- [FieldModifier](#)
- [InterfaceMethodModifier](#)
- [InterfaceModifier](#)
- [MethodHeader](#)
- [MethodModifier](#)
- [ModuleDeclaration](#)
- [PackageModifier](#)
- [PrimitiveType](#)
- [ReceiverParameter](#)
- [TypeParameterModifier](#)
- [TypeVariable](#)
- [UnannClassType](#)
- [VariableArityParameter](#)
- [VariableModifier](#)
- [Wildcard](#)

NormalAnnotation:

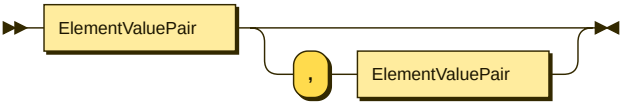


```
NormalAnnotation
    ::= '@' TypeName '(' ElementValuePairList? ')'
```

referenced by:

- [Annotation](#)

ElementValuePairList:

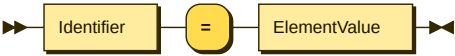


```
ElementValuePairList ::= ElementValuePair ( ',' ElementValuePair )?
```

referenced by:

- [NormalAnnotation](#)

ElementValuePair:

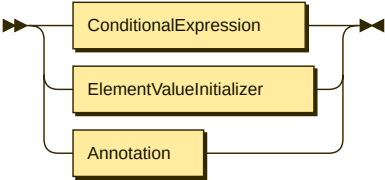


```
ElementValuePair ::= Identifier '=' ElementValue
```

referenced by:

- [ElementValuePairList](#)

ElementValue:

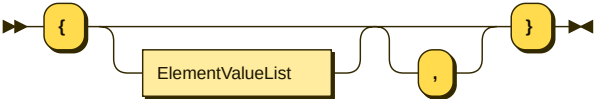


```
ElementValue ::= ConditionalExpression  
              | ElementValueInitializer  
              | Annotation
```

referenced by:

- [DefaultValue](#)
- [ElementValuePair](#)
- [SingleElementAnnotation](#)

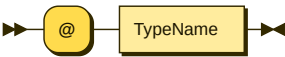
ElementValueArrayInitializer:



```
ElementValueArrayInitializer ::= '{' ElementValueList? ','? '}'
```

no references

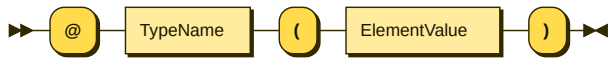
MarkerAnnotation:



```
MarkerAnnotation ::= '@' TypeName
```

referenced by:

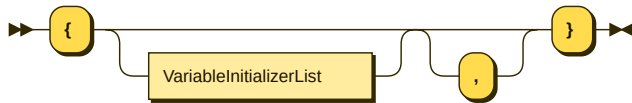
- [Annotation](#)

SingleElementAnnotation:

SingleElementAnnotation
 ::= '@' TypeName '(' ElementValue ')'

referenced by:

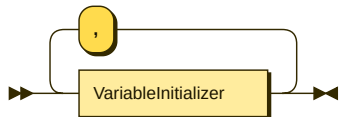
- [Annotation](#)

ArrayInitializer:

ArrayInitializer
 ::= '{' VariableInitializerList? ','? '}'

referenced by:

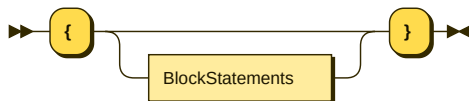
- [ArrayCreationExpression](#)
- [VariableInitializer](#)

VariableInitializerList:

VariableInitializerList
 ::= VariableInitializer (',' VariableInitializer)*

referenced by:

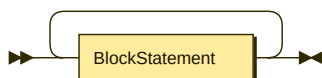
- [ArrayInitializer](#)

Block:

Block ::= '{' BlockStatements? '}'

referenced by:

- [CatchClause](#)
- [Finally](#)
- [InstanceInitializer](#)
- [LambdaBody](#)
- [MethodBody](#)
- [StatementWithoutTrailingSubstatement](#)
- [StaticInitializer](#)
- [SynchronizedStatement](#)
- [TryStatement](#)
- [TryWithResourcesStatement](#)

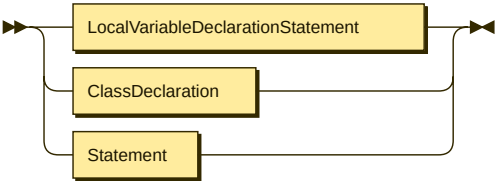
BlockStatements:

BlockStatements
 ::= BlockStatement+

referenced by:

- [Block](#)
- [ConstructorBody](#)
- [SwitchBlockStatementGroup](#)

BlockStatement:

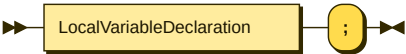


```
BlockStatement
  ::= LocalVariableDeclarationStatement
     | ClassDeclaration
     | Statement
```

referenced by:

- [BlockStatements](#)

LocalVariableDeclarationStatement:

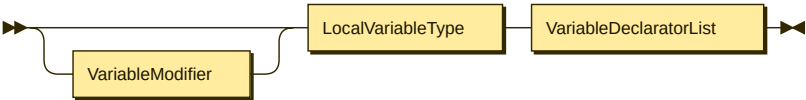


```
LocalVariableDeclarationStatement
  ::= LocalVariableDeclaration ';' ;
```

referenced by:

- [BlockStatement](#)

LocalVariableDeclaration:

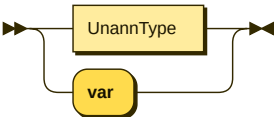


```
LocalVariableDeclaration
  ::= VariableModifier? LocalVariableType VariableDeclaratorList
```

referenced by:

- [ForInit](#)
- [LocalVariableDeclarationStatement](#)

LocalVariableType:

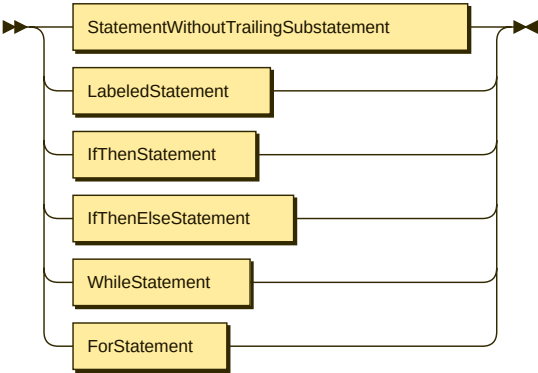


```
LocalVariableType
  ::= UnannType
     | 'var'
```

referenced by:

- [EnhancedForStatement](#)
- [EnhancedForStatementNoShortIf](#)
- [LocalVariableDeclaration](#)
- [Resource](#)

Statement:

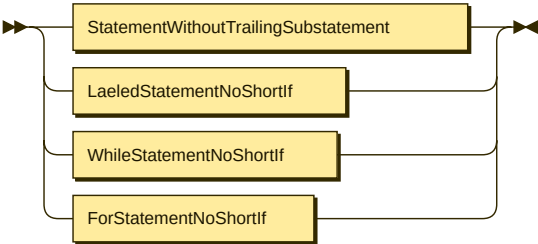


```
Statement
  ::= StatementWithoutTrailingSubstatement
     | LabeledStatement
     | IfThenStatement
     | IfThenElseStatement
     | WhileStatement
     | ForStatement
```

referenced by:

- [BasicForStatement](#)
- [BlockStatement](#)
- [DoStatement](#)
- [EnhancedForStatement](#)
- [IfThenElseStatement](#)
- [IfThenStatement](#)
- [LabeledStatement](#)
- [WhileStatement](#)

StatementNoShortIf:

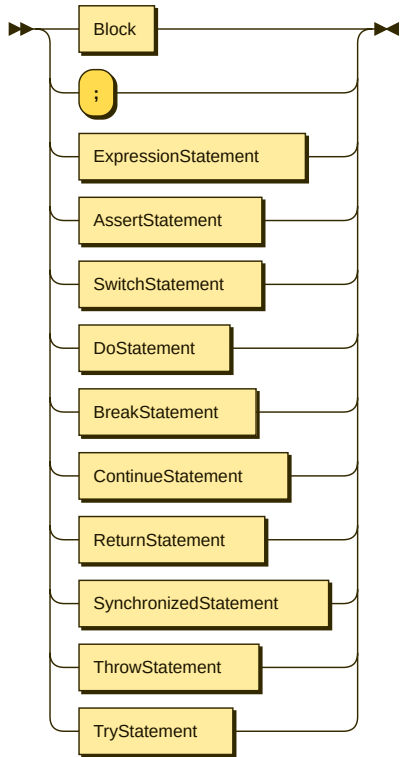


```
StatementNoShortIf
  ::= StatementWithoutTrailingSubstatement
     | LaeledStatementNoShortIf
     | WhileStatementNoShortIf
     | ForStatementNoShortIf
```

referenced by:

- [BasicForStatementNoShortIf](#)
- [IfThenElseStatement](#)
- [IfThenElseStatementNoShortIf](#)
- [WhileStatementNoShortIf](#)

StatementWithoutTrailingSubstatement:

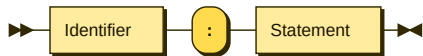


```
StatementWithoutTrailingSubstatement
::= Block
    | ';'
    | ExpressionStatement
    | AssertStatement
    | SwitchStatement
    | DoStatement
    | BreakStatement
    | ContinueStatement
    | ReturnStatement
    | SynchronizedStatement
    | ThrowStatement
    | TryStatement
```

referenced by:

- [Statement](#)
- [StatementNoShortif](#)

LabeledStatement:

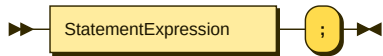


```
LabeledStatement
::= Identifier ':' Statement
```

referenced by:

- [Statement](#)

ExpressionStatement:

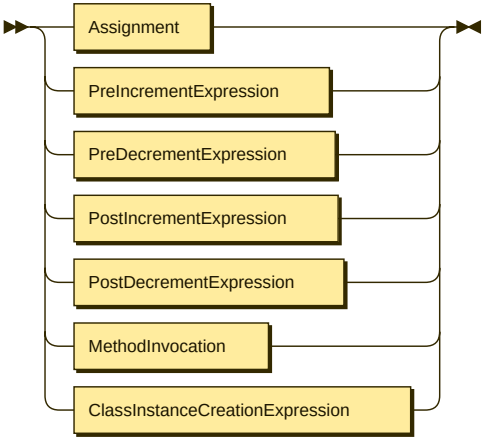


```
ExpressionStatement
::= StatementExpression ';' ;
```

referenced by:

- [StatementWithoutTrailingSubstatement](#)

StatementExpression:



```
StatementExpression ::= Assignment
                    | PreIncrementExpression
                    | PreDecrementExpression
                    | PostIncrementExpression
                    | PostDecrementExpression
                    | MethodInvocation
                    | ClassInstanceCreationExpression
```

referenced by:

- [ExpressionStatement](#)
- [StatementExpressionList](#)

IfThenStatement:



```
IfThenStatement ::= 'if' '(' Expression ')' Statement
```

referenced by:

- [Statement](#)

IfThenElseStatement:



```
IfThenElseStatement ::= 'if' '(' Expression ')' StatementNoShortIf 'else' Statement
```

referenced by:

- [Statement](#)

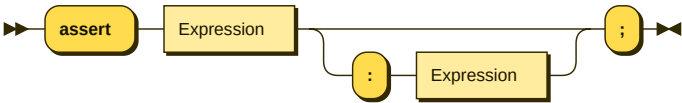
IfThenElseStatementNoShortIf:



```
IfThenElseStatementNoShortIf ::= 'if' '(' Expression ')' StatementNoShortIf 'else' StatementNoShortIf
```

no references

AssertStatement:

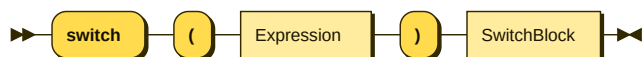


```
AssertStatement ::= 'assert' Expression ( ':' Expression )? ';'
```

referenced by:

- [StatementWithoutTrailingSubstatement](#)

SwitchStatement:

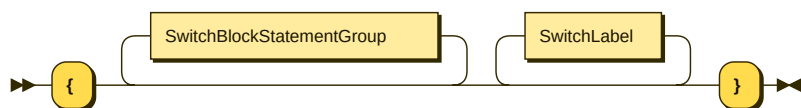


SwitchStatement
 ::= 'switch' '(' Expression ')' SwitchBlock

referenced by:

- [StatementWithoutTrailingSubstatement](#)

SwitchBlock:

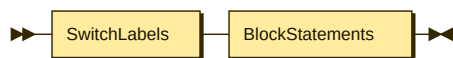


SwitchBlock
 ::= '{' SwitchBlockStatementGroup* SwitchLabel* '}'

referenced by:

- [SwitchStatement](#)

SwitchBlockStatementGroup:

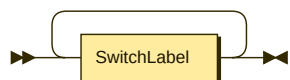


SwitchBlockStatementGroup
 ::= SwitchLabels BlockStatements

referenced by:

- [SwitchBlock](#)

SwitchLabels:

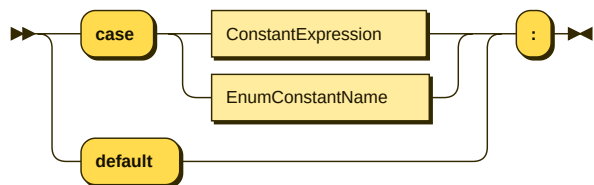


SwitchLabels
 ::= SwitchLabel+

referenced by:

- [SwitchBlockStatementGroup](#)

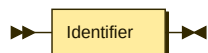
SwitchLabel:



SwitchLabel
 ::= ('case' (ConstantExpression | EnumConstantName) | 'default') ':'

referenced by:

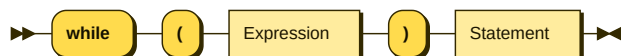
- [SwitchBlock](#)
- [SwitchLabels](#)

EnumConstantName:

```
EnumConstantName
    ::= Identifier
```

referenced by:

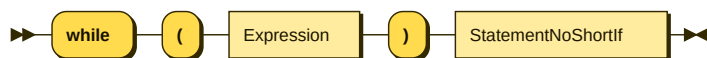
- [SwitchLabel](#)

WhileStatement:

```
WhileStatement
    ::= 'while' '(' Expression ')' Statement
```

referenced by:

- [Statement](#)

WhileStatementNoShortIf:

```
WhileStatementNoShortIf
    ::= 'while' '(' Expression ')' StatementNoShortIf
```

referenced by:

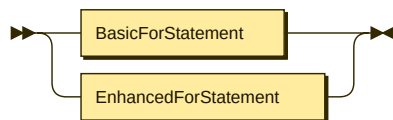
- [StatementNoShortIf](#)

DoStatement:

```
DoStatement
    ::= 'do' Statement 'while' '(' Expression ')' ';' ;
```

referenced by:

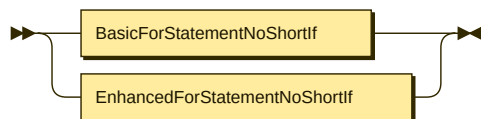
- [StatementWithoutTrailingSubstatement](#)

ForStatement:

```
ForStatement
    ::= BasicForStatement
    | EnhancedForStatement
```

referenced by:

- [Statement](#)

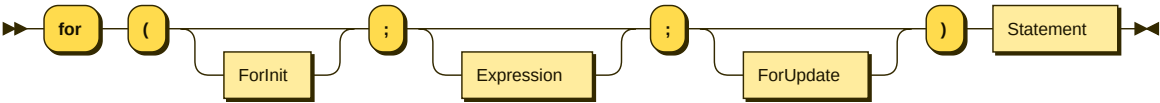
ForStatementNoShortIf:

```
ForStatementNoShortIf
    ::= BasicForStatementNoShortIf
    | EnhancedForStatementNoShortIf
```

referenced by:

- [StatementNoShortIf](#)

BasicForStatement:

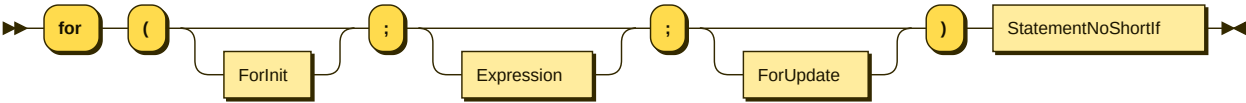


BasicForStatement ::= 'for' '(' ForInit? ';' Expression? ';' ForUpdate? ')' Statement

referenced by:

- [ForStatement](#)

BasicForStatementNoShortIf:



BasicForStatementNoShortIf ::= 'for' '(' ForInit? ';' Expression? ';' ForUpdate? ')' StatementNoShortIf

referenced by:

- [ForStatementNoShortIf](#)

ForInit:

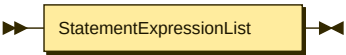


ForInit ::= StatementExpressionList
 | LocalVariableDeclaration

referenced by:

- [BasicForStatement](#)
- [BasicForStatementNoShortIf](#)

ForUpdate:



ForUpdate ::= StatementExpressionList

referenced by:

- [BasicForStatement](#)
- [BasicForStatementNoShortIf](#)

StatementExpressionList:



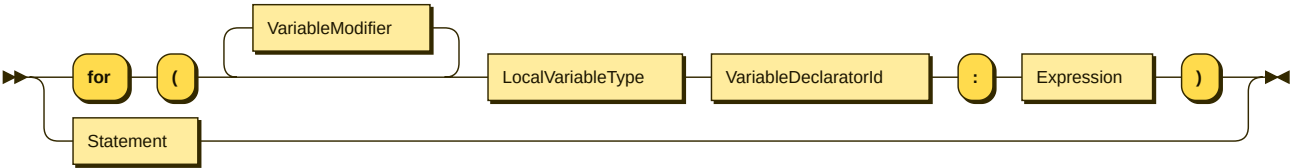
StatementExpressionList ::= StatementExpression (',' StatementExpression)*

referenced by:

- [ForInit](#)

- [ForUpdate](#)

EnhancedForStatement:

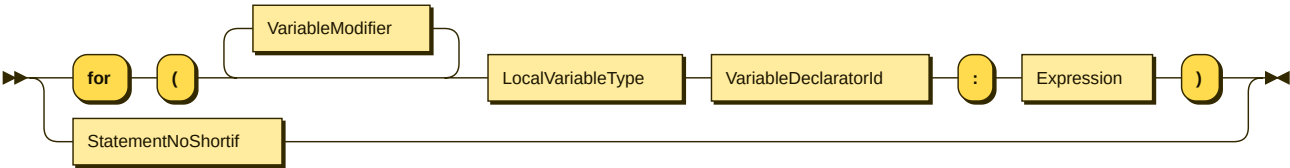


```
EnhancedForStatement ::= 'for' '(' VariableModifier* LocalVariableType VariableDeclaratorId ':' Expression ')' Statement
```

referenced by:

- [ForStatement](#)

EnhancedForStatementNoShortif:

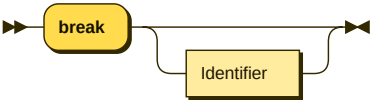


```
EnhancedForStatementNoShortIf ::= 'for' '(' VariableModifier* LocalVariableType VariableDeclaratorId ':' Expression ')' StatementNoShortif
```

referenced by:

- [ForStatementNoShortif](#)

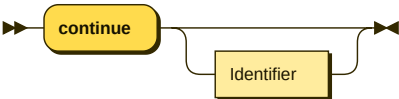
Breakstatement:



```
Breakstatement ::= 'break' Identifier?
```

no references

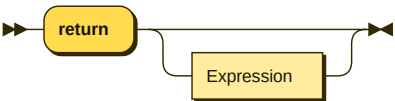
Continue:



```
Continue ::= 'continue' Identifier?
```

no references

ReturnStatement:



```
ReturnStatement ::= 'return' Expression?
```

referenced by:

- [StatementWithoutTrailingSubstatement](#)

ThrowStatement:



ThrowStatement ::= 'throw' Expression ';'

referenced by:

- [StatementWithoutTrailingSubstatement](#)

SynchronizedStatement:

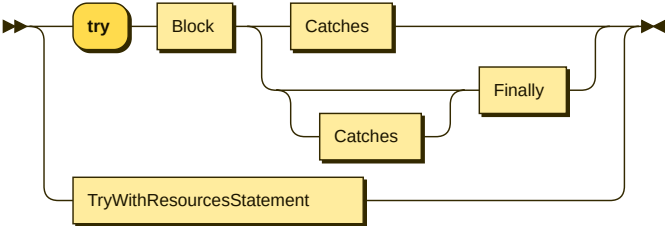


SynchronizedStatement ::= 'synchronized' '(' Expression ')' Block

referenced by:

- [StatementWithoutTrailingSubstatement](#)

TryStatement:

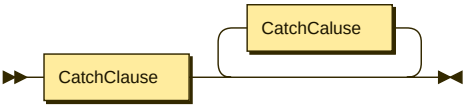


TryStatement ::= 'try' Block (Catches | Catches? Finally) | TryWithResourcesStatement

referenced by:

- [StatementWithoutTrailingSubstatement](#)

Catches:



Catches ::= CatchClause CatchCaluse*

referenced by:

- [TryStatement](#)
- [TryWithResourcesStatement](#)

CatchCaluse:

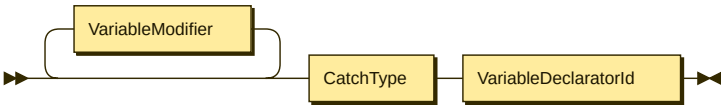


CatchCaluse ::= 'catch' '(' CatchFormalParameter ')' Block

referenced by:

- [Catches](#)

CatchFormalParameter:

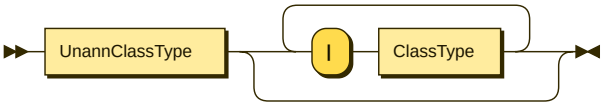


CatchFormalParameter ::= VariableModifier* CatchType VariableDeclaratorId

referenced by:

- [CatchCaluse](#)

CatchType:



CatchType ::= UnannClassType ('|' ClassType)*

referenced by:

- [CatchFormalParameter](#)

Finally:

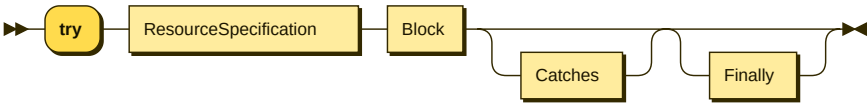


Finally ::= 'finally' Block

referenced by:

- [TryStatement](#)
- [TryWithResourcesStatement](#)

TryWithResourcesStatement:

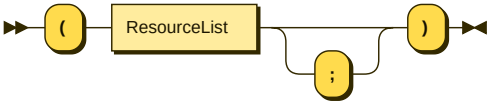


TryWithResourcesStatement ::= 'try' ResourceSpecification Block Catches? Finally?

referenced by:

- [TryStatement](#)

ResourceSpecification:

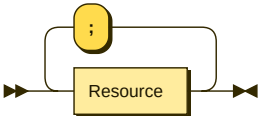


ResourceSpecification ::= '(' ResourceList ';' '?' ')'

referenced by:

- [TryWithResourcesStatement](#)

ResourceList:

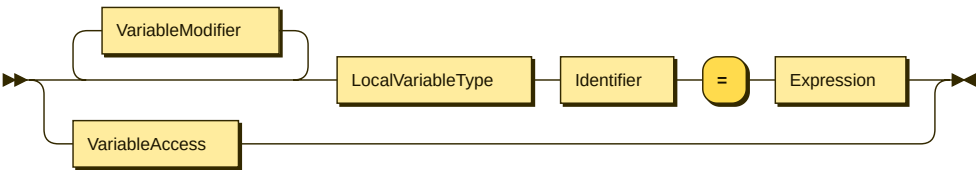


```
ResourceList ::= Resource ( ';' Resource )*
```

referenced by:

- [ResourceSpecification](#)

Resource:

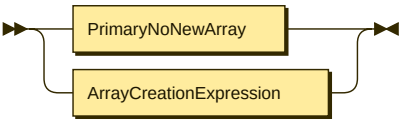


```
Resource ::= VariableModifier* LocalVariableType Identifier '=' Expression  
          | VariableAccess
```

referenced by:

- [ResourceList](#)

Primary:

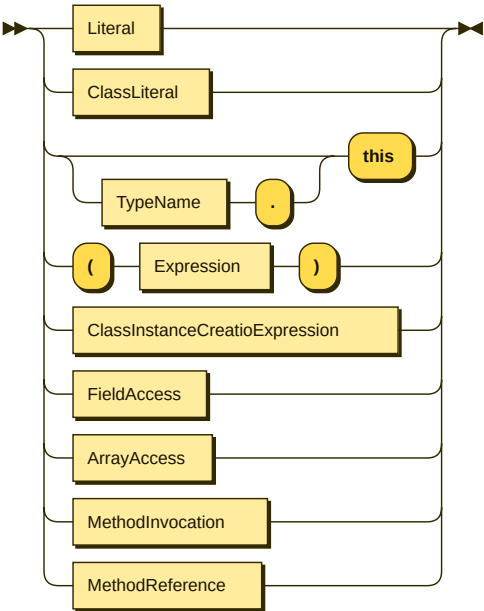


```
Primary ::= PrimaryNoNewArray  
          | ArrayCreationExpression
```

referenced by:

- [ClassInstanceCreationExpression](#)
- [ExplicitConstructorInvocation](#)
- [FieldAccess](#)
- [MethodInvocation](#)
- [MethodReference](#)
- [PostfixExpression](#)

PrimaryNoNewArray:



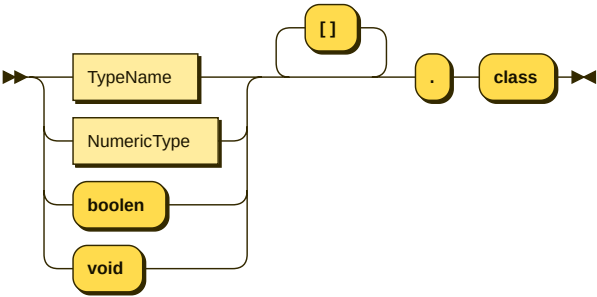
```
PrimaryNoNewArray ::= Literal  
                  | ClassLiteral  
                  | ( TypeName '.' )? 'this'  
                  | '(' Expression ')'  
                  | ClassInstanceCreatioExpression  
                  | FieldAccess
```

- | [ArrayAccess](#)
- | [MethodInvocation](#)
- | [MethodReference](#)

referenced by:

- [ArrayAccess](#)
- [Primary](#)

ClassLiteral:

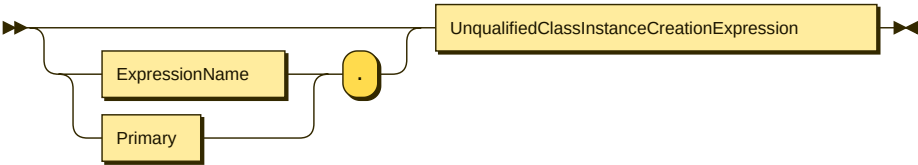


```
ClassLiteral ::= ( TypeName | NumericType | 'boolean' | 'void' ) '[' ']' '*' '.' 'class'
```

referenced by:

- [PrimaryNoNewArray](#)

ClassInstanceCreationExpression:

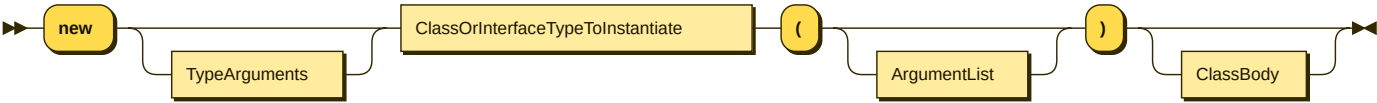


```
ClassInstanceCreationExpression ::= ( ( ExpressionName | Primary ) '.' )? UnqualifiedClassInstanceCreationExpression
```

referenced by:

- [StatementExpression](#)

UnqualifiedClassInstanceCreationExpression:

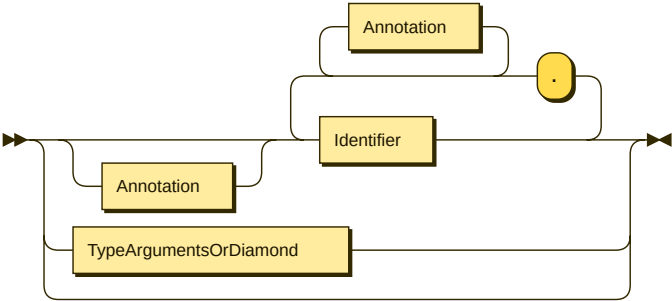


```
UnqualifiedClassInstanceCreationExpression ::= 'new' TypeArguments? ClassOrInterfaceTypeToInstantiate '(' ArgumentList? ')' ClassBody?
```

referenced by:

- [ClassInstanceCreationExpression](#)

ClassOrInterfaceTypeToInstantiate:

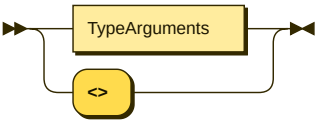


```
ClassOrInterfaceTypeToInstantiate ::= ( Annotation? Identifier ( '.' Annotation* Identifier )* | TypeArgumentsOrDiamond )?
```

referenced by:

- [UnqualifiedClassInstanceCreationExpression](#)

TypeArgumentsOrDiamond:

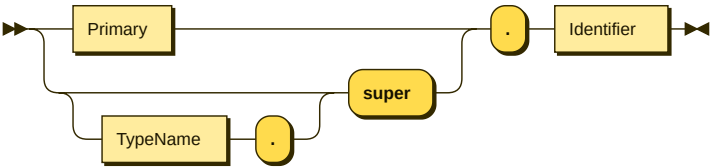


```
TypeArgumentsOrDiamond ::= TypeArguments | '<>'
```

referenced by:

- [ClassOrInterfaceTypeToInstantiate](#)

FieldAccess:

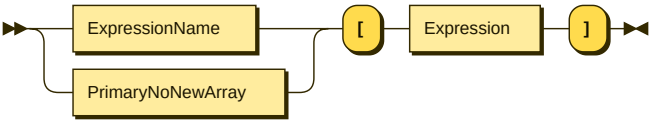


```
FieldAccess ::= ( Primary | ( TypeName '.' )? 'super' ) '.' Identifier
```

referenced by:

- [LeftHandSide](#)
- [PrimaryNoNewArray](#)

ArrayAccess:

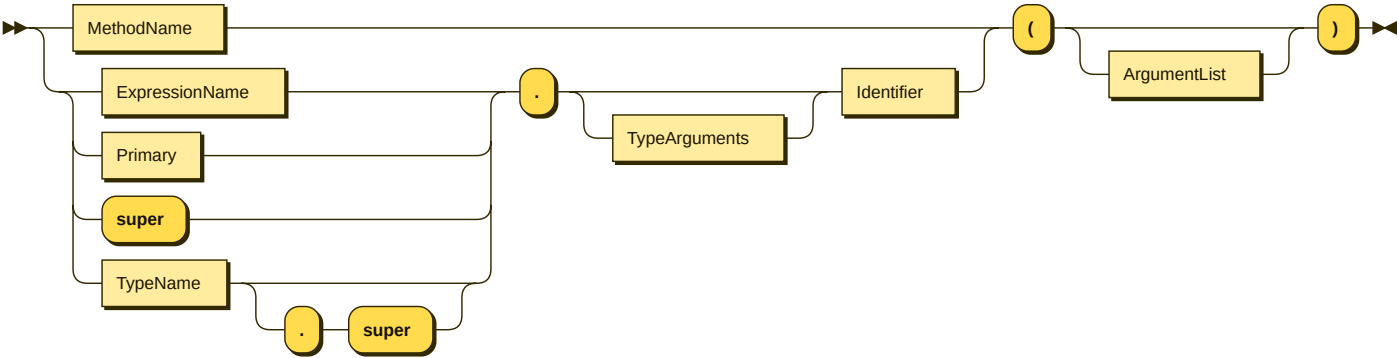


```
ArrayAccess ::= ( ExpressionName | PrimaryNoNewArray ) '[' Expression ']'
```

referenced by:

- [LeftHandSide](#)
- [PrimaryNoNewArray](#)

MethodInvocation:



```
MethodInvocation
```

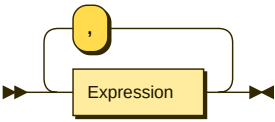


```
 ::= ( MethodName | ( ExpressionName | Primary | 'super' | TypeName ( '.' 'super' )? ) '.' TypeArguments? Identifier ) '('  
   ArgumentList? ')'
```

referenced by:

- [PrimaryNoNewArray](#)
- [StatementExpression](#)

ArgumentList:

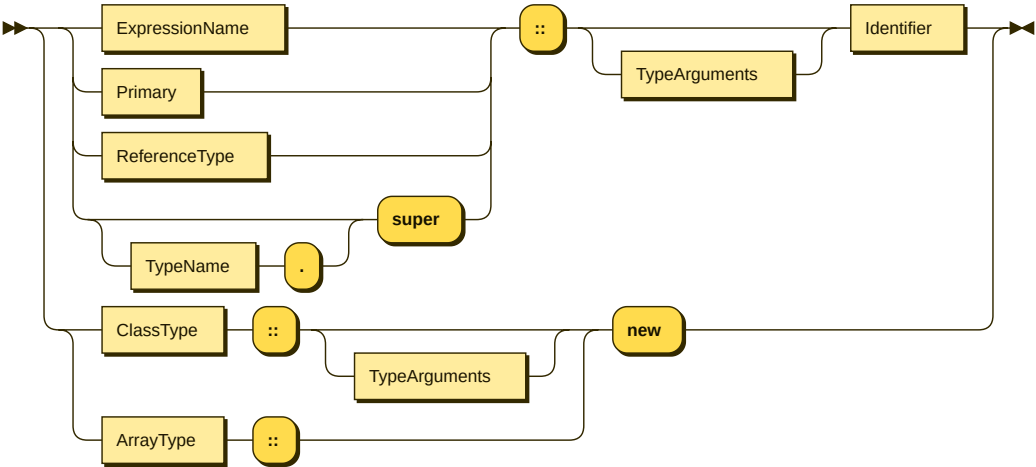


ArgumentList
 ::= Expression (',' Expression)*

referenced by:

- [ExplicitConstructorInvocation](#)
- [MethodInvocation](#)
- [UnqualifiedClassInstanceCreationExpression](#)

MethodReference:

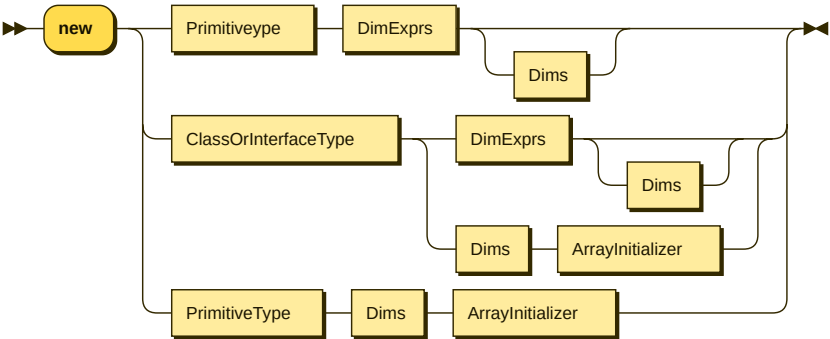


MethodReference
 ::= (ExpressionName | Primary | ReferenceType | (TypeName '.')? 'super') '::' TypeArguments? Identifier
 | (ClassType '::' TypeArguments? | ArrayType '::') 'new'

referenced by:

- [PrimaryNoNewArray](#)

ArrayCreationExpression:



ArrayCreationExpression
 ::= 'new' (PrimitiveType DimExprs Dims? | ClassOrInterfaceType (DimExprs Dims? | Dims ArrayInitializer) | PrimitiveType Dims
 ArrayInitializer)

referenced by:

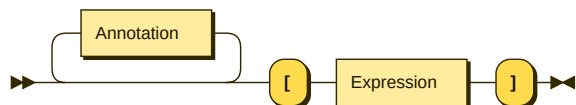
- [Primary](#)

DimExprs:

DimExprs ::= DimExpr+

referenced by:

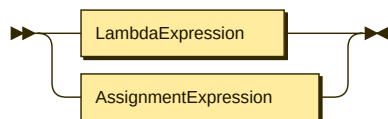
- [ArrayCreationExpression](#)

DimExpr:

DimExpr ::= Annotation* '[' Expression ']'

referenced by:

- [DimExprs](#)

Expression:

Expression
::= LambdaExpression
| AssignmentExpression

referenced by:

- [ArgumentList](#)
- [ArrayAccess](#)
- [AssertStatement](#)
- [Assignment](#)
- [BasicForStatement](#)
- [BasicForStatementNoShortif](#)
- [ConditionalExpression](#)
- [ConstantExpression](#)
- [DimExpr](#)
- [DoStatement](#)
- [EnhancedForStatement](#)
- [EnhancedForStatementNoShortif](#)
- [IfThenElseStatement](#)
- [IfThenElseStatementNoShortif](#)
- [IfThenStatement](#)
- [LambdaBody](#)
- [PrimaryNoNewArray](#)
- [Resource](#)
- [ReturnStatement](#)
- [SwitchStatement](#)
- [SynchronizedStatement](#)
- [ThrowStatement](#)
- [VariableInitializer](#)
- [WhileStatement](#)
- [WhileStatementNoShortif](#)

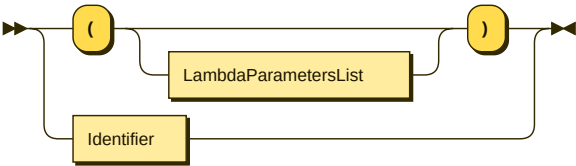
LambdaExpression:

LambdaExpression
::= LambdaParameters '->' LambdaBody

referenced by:

- [CastExpression](#)
- [ConditionalExpression](#)
- [Expression](#)

LambdaParameters:

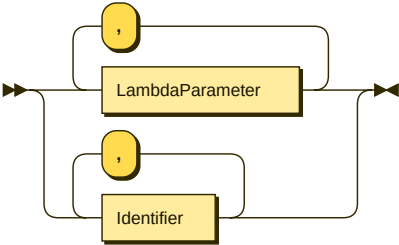


```
LambdaParameters
  ::= '(' LambdaParametersList? ')'
  | Identifier
```

referenced by:

- [LambdaExpression](#)

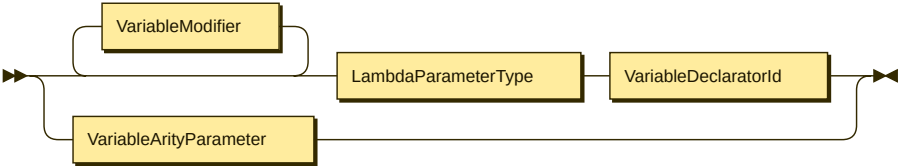
LambdaParameterList:



```
LambdaParameterList
  ::= LambdaParameter ( ',' LambdaParameter )*
  | Identifier ( ',' Identifier )*
```

no references

LambdaParameter:

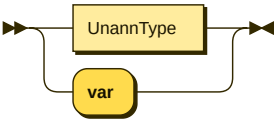


```
LambdaParameter
  ::= VariableModifier* LambdaParameterType VariableDeclaratorId
  | VariableArityParameter
```

referenced by:

- [LambdaParameterList](#)

LambdaParameterType:

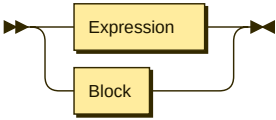


```
LambdaParameterType
  ::= UnannType
  | 'var'
```

referenced by:

- [LambdaParameter](#)

LambdaBody:

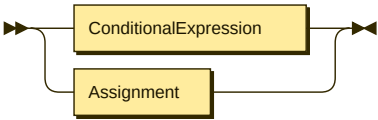


LambdaBody ::= Expression | Block

referenced by:

- [LambdaExpression](#)

AssignmentExpression:

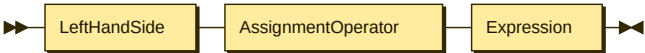


AssignmentExpression ::= ConditionalExpression | Assignment

referenced by:

- [Expression](#)

Assignment:

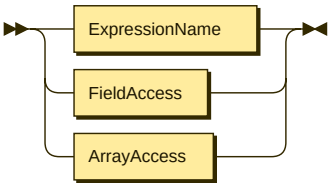


Assignment ::= LeftHandSide AssignmentOperator Expression

referenced by:

- [AssignmentExpression](#)
- [StatementExpression](#)

LeftHandSide:

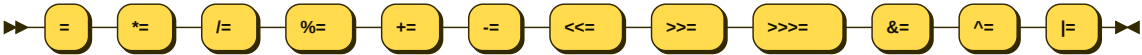


LeftHandSide ::= ExpressionName | FieldAccess | ArrayAccess

referenced by:

- [Assignment](#)

AssignmentOperator:

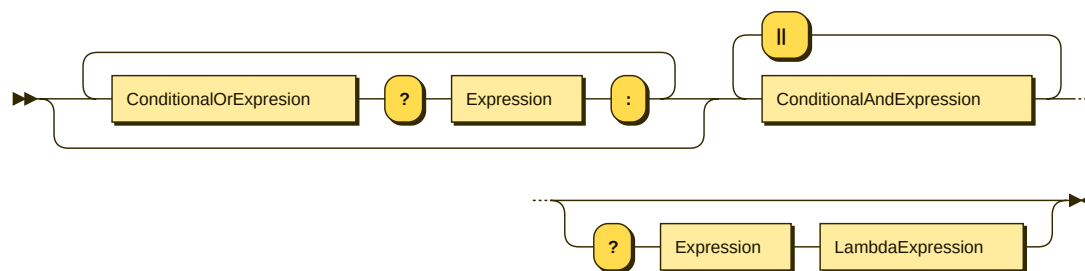


AssignmentOperator ::= '=' '*' '/' '%' '+' '-' '<<' '>>' '>>>' '&' '^' '|'

referenced by:

- [Assignment](#)

ConditionalExpression:

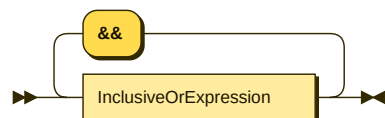


ConditionalExpression
 ::= (ConditionalOrExpression '?' Expression ':') * ConditionalAndExpression ('||' ConditionalAndExpression) * ('?' Expression LambdaExpression) ?

referenced by:

- [AssignmentExpression](#)
- [ElementValue](#)

ConditionalAndExpression:

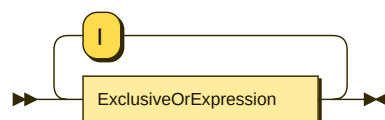


ConditionalAndExpression
 ::= InclusiveOrExpression ('&&' InclusiveOrExpression) *

referenced by:

- [ConditionalExpression](#)

InclusiveOrExpression:

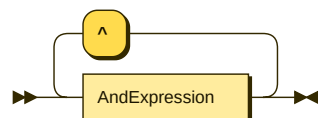


InclusiveOrExpression
 ::= ExclusiveOrExpression ('|' ExclusiveOrExpression) *

referenced by:

- [ConditionalAndExpression](#)

ExclusiveOrExpression:

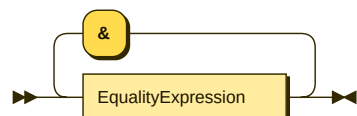


ExclusiveOrExpression
 ::= AndExpression ('^' AndExpression) *

referenced by:

- [InclusiveOrExpression](#)

AndExpression:

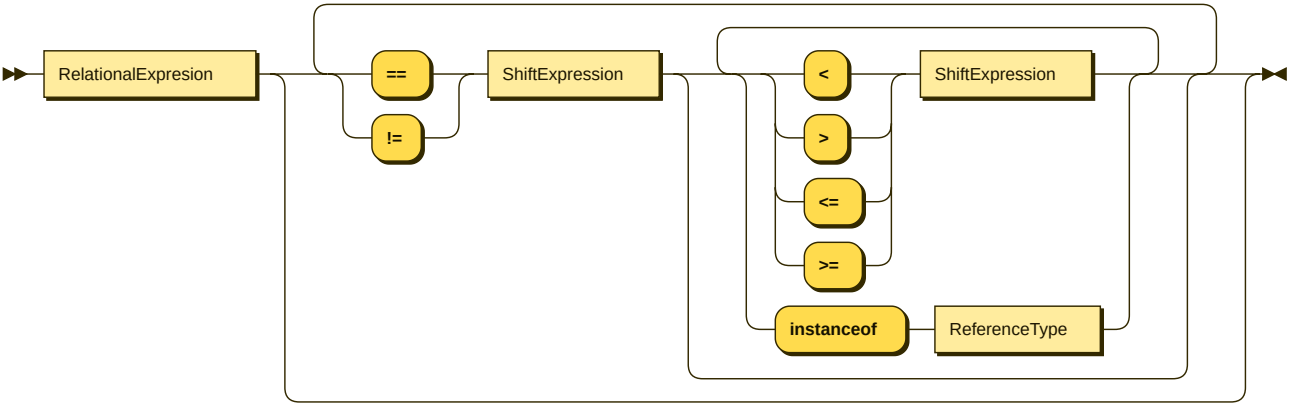


AndExpression
 ::= EqualityExpression ('&' EqualityExpression) *

referenced by:

- [ExclusiveOrExpression](#)

EqualityExpression:

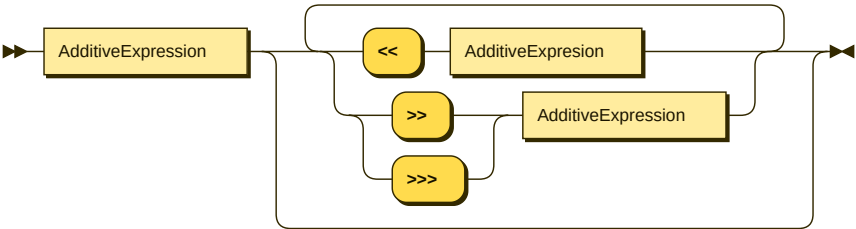


```
EqualityExpression ::= RelationalExpression ( ( '=' | '!=' ) ShiftExpression ( ( '<' | '>' | '<=' | '>=' ) ShiftExpression | 'instanceof' ReferenceType )* )*
```

referenced by:

- [AndExpression](#)

ShiftExpression:

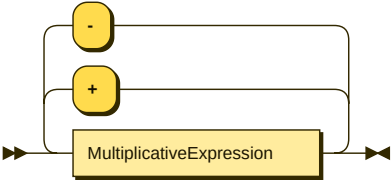


```
ShiftExpression ::= AdditiveExpression ( '<<' AdditiveExpression | ( '>>' | '>>>' ) AdditiveExpression )*
```

referenced by:

- [EqualityExpression](#)

AdditiveExpression:

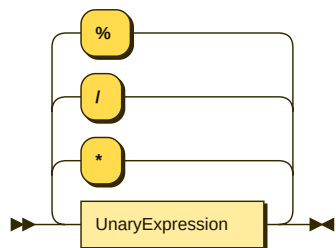


```
AdditiveExpression ::= MultiplicativeExpression ( ( '+' | '-' ) MultiplicativeExpression )*
```

referenced by:

- [ShiftExpression](#)

MultiplicativeExpression:

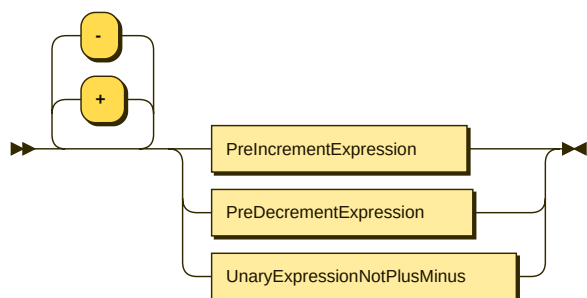


MultiplicativeExpression
`::= UnaryExpression (('*' | '/' | '%') UnaryExpression)*`

referenced by:

- [AdditiveExpression](#)

UnaryExpression:

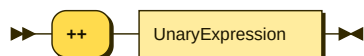


UnaryExpression
`::= ('+' | '-')* (PreIncrementExpression | PreDecrementExpression | UnaryExpressionNotPlusMinus)`

referenced by:

- [CastExpression](#)
- [MultiplicativeExpression](#)
- [PreDecrementExpression](#)
- [PreIncrementExpression](#)
- [UnaryExpressionNotPlusMinus](#)

PreIncrementExpression:

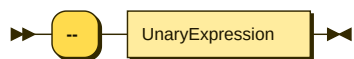


PreIncrementExpression
`::= '++' UnaryExpression`

referenced by:

- [StatementExpression](#)
- [UnaryExpression](#)

PreDecrementExpression:

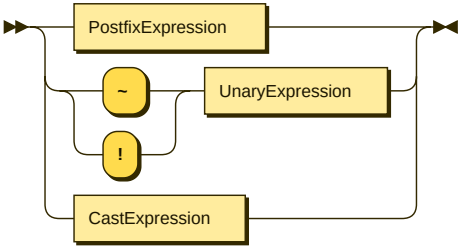


PreDecrementExpression
`::= '--' UnaryExpression`

referenced by:

- [StatementExpression](#)
- [UnaryExpression](#)

UnaryExpressionNotPlusMinus:

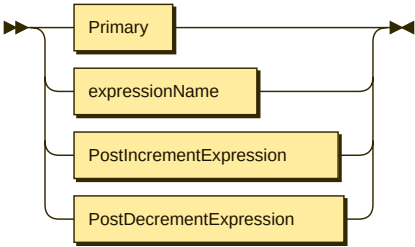


```
UnaryExpressionNotPlusMinus ::= PostfixExpression
                             | ( '~' | '!' ) UnaryExpression
                             | CastExpression
```

referenced by:

- [CastExpression](#)
- [UnaryExpression](#)

PostfixExpression:

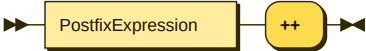


```
PostfixExpression ::= Primary
                  | expressionName
                  | PostIncrementExpression
                  | PostDecrementExpression
```

referenced by:

- [PostDecrementExpression](#)
- [PostIncrementExpression](#)
- [UnaryExpressionNotPlusMinus](#)

PostIncrementExpression:

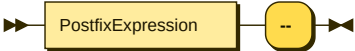


```
PostIncrementExpression ::= PostfixExpression '++'
```

referenced by:

- [PostfixExpression](#)
- [StatementExpression](#)

PostDecrementExpression:

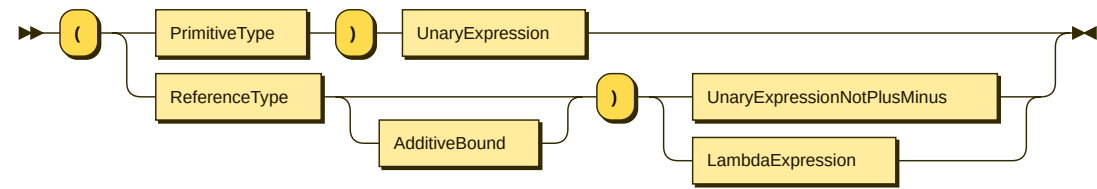


```
PostDecrementExpression ::= PostfixExpression '--'
```

referenced by:

- [PostfixExpression](#)
- [StatementExpression](#)

CastExpression:



CastExpression ::= '(' (PrimitiveType ')' UnaryExpression | ReferenceType AdditiveBound? ')' (UnaryExpressionNotPlusMinus | LambdaExpression))

referenced by:

- [UnaryExpressionNotPlusMinus](#)

ConstantExpression:



ConstantExpression ::= Expression

referenced by:

- [SwitchLabel](#)