

Nomes e Valores

<nome> := <id> { .<id> }

<valor> := <valor_inteiro> | <valor_real> | <caracter> | <string> | null | <booleano>

Pacotes

<InicoCompilacao> := [<DeclaracaoPacote>] { <ListaImport> } { <ListaDeclaracao> }

<declaracaoPacote> := package <nome> ;

<ListaImport> := import <nome> [. *] ;

<ListaDeclaracao> := ; | <DeclaracaoClass> | <DeclaracaoInterface>

Tipos

<Tipo> := <TipoPrimitivo> | <RefTipo>

<TipoPrimitivo> := byte | short | int | long | char | float | double | boolean

<RefTipo> := <ClassInterfaceTipo> | <VectorTipo>

<ClassInterfaceTipo> := <nome>

<VectorTipo> := <nome> [] { [] }

<TipoInterface> := <ClassInterfaceTipo>

<TipoClass> := <ClassInterfaceTipo>

Modificadores

<Modificador> := { Public | private | protected | static | abstract | final | native | synchronized | transient | volatile }

Declaração de Interface

<DeclaracaoInterface> := [<Modificador>] interface <id> [<Extends>] <CorpoInterface>

<Extends> := extends <ClassInterfaceTipo> { , <ClassInterfaceTipo> }

<CorpoInterface> := { [<ListaDeclaracaoMembrosInterface>] }

<ListaDeclaracaoMembrosInterface> := <DeclaracaoMembrosInterface> {
<DeclaracaoMembrosInterface> }

<DeclaracaoMembrosInterface> := <DeclaracaoClass> | <DeclaracaoInterface> |
<DeclaracaoMetodoAbstrato> | <DeclaracaoConstante>

<DeclaracaoConstante> := <DeclaracaoCampo>

<DeclaracaoMetodoAbstrato> := <HdrMetodo> ;

Declaração de Class

<DeclaracaoClass> := [<modificador>] class <Id> [extends <ClassInterfaceTipo>] [<Interfaces>] <CorpoClass>

<Interfaces> := implements <ListaTipoInterface>

<ListaTipoInterface> := <TipoInteface> { , <TipoInterfcae> }

<CorpoClass> := { [<ListaDeclaracaoCorpoClass>] }

<ListaDeclaracaoCorpoClass> := <DeclaracaoCorpoClass> { <DeclaracaoCorpoClass> }

<DeclaracaoCorpoClass> := [<modificador>] <DeclaracaoCorpoClass1>

<DeclaracaoCorpoClass1> := class <DeclaracaoClass> | interface <DeclaracaoInteface> | <InicicalizacaoStatic> | (<tipo> | void) <DeclaracaoMembrosClass> | <id> <DeclaracaoConstrutor>

<DeclaracaoMembrosClass> := <DeclaracaoCampo> | <DeclaracaoMetodo>

Declaração de Metodo

<DeclaracaoMetodo> := <HdrMetodo> <CorpoMetodo>

<HdrMetodo> := [<Modificador>] (<Tipo> | void) <DefinicaoMetodo> [<Excepcao>]

<DefinicaoMetodo> := <id> ([<ListaPArumentrosFormais>]) { [] }

<ListaParametrosFormais> := <ParametrosFormais> { , <ParametrosFormais> }

<ParametrosFormais> := <Tipo> <IdDeclaracaoVariavel>

<Excepcao> := throws <ListaTipoClass>

<ListaTipoClass> := <TipoClass> { , <TipoClass> }

<CorpoMetodo> := ; | <Bloco>

Declaração de variaveis e Campos

<DeclaracaoCampo> := <Modificador> <Tipo> <DeclaracaoCampo1>;

<DeclaracaoCampo1> := <DeclaracaoVariavel> { , <DeclaracaoVariavel> }

<ListaDeclaracaoVariavel> := <DeclaracaoVariavel> { , <DeclaracaoVariavel> }

<DeclaracaoVariavel> := <IdDeclaracaoVariavel> [= <InicicalizacaoVariavel>]

<IdDeclaracaoVariavel> := <Id> [[]]

Inicializadores

<InicicalizacaoStatic> := <Bloco>

<InicializadoresVariaveis> := <InicializacaoVariavel> [, <InicializacaoVariavel>]

<inicializacaoArray> := { [<InicializadoresVariaveis>] [,] }

<InicializacaoVariavel> := <expressao> | <InicializacaoArray>

Declaracao de Construtor

<DeclaracaoConstrutor> := <DefinicaoConstrutor> [<excepcao>] <CorpoConstrutor>

<DefinicaoConstrutor> := <Id> ([<ListaParametrosFormias>])

<corpoConstrutor> := { [**super** ([<listaArgumentos>]);] { <InvExpressaoConstrutor> } [<listaDeclaracoesBloco>] }

<InvExpressaoConstrutor> := **this**.<nome>= <InicicalizacaoVariavel>;

Blocos e Declarações

<bloco> := { [<listaDeclaracoesBloco>] }

<listaDeclaracoesBloco> := <declaracoesBloco> { < declaracoesBloco > }

<declaracoesBloco> := <declaracaoVariavelLocal0> | <declaracao> | <declaracaoClass>

<declaracaoVariavelLocal0> := <declaracaoVariavelLocal>;

<declaracaoVariavelLocal> := <tipo> <listaDeclaracaoVariavel>

<declaracao> := <declaracoes> | <declaracaoRotulo> | <declaracaoIF> | <declaracaoWhile> | <declaracaoFor>

<declaracoes> := <bloco> | <declaracaoVazia> | <declaracaoExpressao> | <declaracaoSwitch> | <declaracaoDo> | <declaracaoBreak> | <declaracaoContinue> | <DeclaracaoReturn> | <declaracaoSynchronized> | <declaracaoThrow> | <declaracaoTry>

<declaracaoVazia> := ;

<declaracaoRotulo> := <id> : <declaracao>

<declaracaoExpressao> := <Atribuicao> | <PreIncremento> | <PreDecremento> | <PosEncremento> | <PosDecremento> | <InvocacaoMetodo> | <InstanciarClass>

<declaracaoIF> := if (<expressao>) <declaracao> [else <declaracao>]
<declaracaoSwitch> := switch (<expressao>) <BlocoSwitch>
<BlocoSwitch> := { {<rotuloSwitch> } }
<rotuloSwitch> := case <Expressao> : [<declaracao>] | default : [<declaracao>]
<declaracaoWhile> := while (<expressao>) <declaracao>
<declaracaoDo> := do <declaracao> while (<expressao>)
<declaracaoFor> := for((<inicializacaoFor>; <expressao>; <actualizacaoFor> | <tipo> <nome> : <nome>)) <declaracao>
<inicializacaoFor> := <ListaDeclaracaoExpressao> | <declaracaoVariavelLocal>
<actualizacaoFor> := <ListaDeclaracaoExpressao>
<ListaDeclaracaoExpressao> := <declaracaoExpressao> {<declaracaoExpressao> }
<declaracaoBreak> := break [<id>];
<declaracaoContinue> := continue [<id>];
<DeclaracaoReturn> := return [<expressao>];
<declaracaoThrow> := throw <expressao>;
<declaracaoSynchronized> := synchronized (<expressao>) <bloco>
<declaracaoTry> := try <bloco> <declaracaoTry1>
<declaracaoTry1> := <catches> [<declaracaoTry2>] | <declaracaoTry2>
<declaracaoTry2> := <declaracaoFinally>
<catches> := <catchclause> { <catchclause> }
<catchClause> := catch (<ParametrosFormais>) <bloco>
<declaracaoFinally> := finally <bloco>

Expressões

<Expressao> := <AtribuicaoExpressao>
<AtribuicaoExpressao> := <Atribuicao> | <OperacaoTernaria>
<Atribuicao> := <LHS> <AtribuicaoOperador> <AtribuicaoExpressao>
<LHS> := <nome> | <FieldAccess - Duvida> | <AcederVector>
<AtribuicaoOperador> := *= | = | /= | %= | += | -= | <= | >= | >>= | &= | ^= | |=

<OperacaoTernaria> := <ExpressaoOU> [? <Expressao> : <OperacaoTernaria>]
<ExpressaoOU> := <ExpressaoE> { || <ExpressaoE> }
<ExpressaoE>:= <ExpressaoOR>{ && <ExpressaoOR> }
<ExpressaoOR>:= <ExpressaoXOR> { | <ExpressaoXOR> }
<ExpressaoXOR>:= <ExpressaoAND>{ ^ <ExpressaoAND> }
<ExpressaoAND>:= <ExpressaoIgual>{ & <ExpressaoIgual> }
<ExpressaoIgual>:= <ExpressaoComparacao> { == <ExpressaoComparacao> }
<ExpressaoComparacao>:= <ExpressaoDeslocamento> { (< > | <= | >=)
<ExpressaoDeslocamento> | instanceof <ClassInterfaceTipo> }
<ExpressaoDeslocamento>:= <ExpressaoAdicaoSubtracao> { (<< | >> | >>>)
<ExpressaoAdicaoSubtracao> }
<ExpressaoAdicaoSubtracao>:= <MultiplicacaoDivisaoMod> { (+ | -)
<MultiplicacaoDivisaoMod> }
<MultiplicacaoDivisaoMod>:= <ExpressaoUnaria> { (* | / | %) <ExpressaoUnaria> }
<ExpressaoUnariaSemMaisMenos> ::= <Casting> | <ExpressaoPosterior> |
~<ExpressaoUnaria>| !<ExpressaoUnaria>
<Casting> := ((<Tipo> [<ParentesesRectos>]) <ExpressaoUnaria> | <Expressao>)
<ExpressaoUnariaSemMaisMenos>)
<ExpressaoPosterior>:= <Principal> | <nome> | <PosIncremento> | <PosDecremento>
<PosIncremento> := <ExpressaoPosterior> ++
<PosDecremento>:= <ExpressaoPosterior> --
<ExpressaoUnaria>:= (<PreIncremento> | <PreDecremento> |
<ExpressaoUnariaSemMaisMenos>) { (+ | -) (<PreIncremento>) | (<PreDecremento>) |
<ExpressaoUnariaSemMaisMenos>) }
<PreIncremento> := ++ <ExpressaoUnaria>
<PreDecremento> := -- <ExpressaoUnaria>
<Principal> := <PrincipalSemNovoVector> | <CriacaoVector>
<PrincipalSemNovoVector>:= <valor> | this | (<expressao>) | <InstanciarClass> |
<CampoAcesso> | <InvocacaoMetodo> | <AcederVector>
<CriacaoVector>:= new (<Tipo>) <ListaExpressaoParentesesRectos> [<ParentesesRectos>]
<InstanciarClass> := new <TipoClass> ([<listaArgumentos>]) [<CorpoClass>]
<CampoAcesso> := <Principal> . <Id> | super.<Id>
<InvocacaoMetodo> := (<nome> | <Principal>.<Id> | super.<id>) ([<listaArgumentos>])
<AcederVector> := (<nome> | <PrincipalSemNovoVector>) [<Expressao>]

<listaArgumentos>:= <Expressao> { ,<Expressao> }

**<ListaExpressaoParentesesRectos>:= <ExpressaoParentesesRectos>
{<ExpressaoParentesesRectos> }**

<ExpressaoParentesesRectos>:= [<Expressao>]

<ParentesesRectos>:= [] {[]}