

José Carlos Herrera 1007217
Renato Cabrera 1010617
TABLA DE ESTADOS

```
Program' ::= Program
1 Program ::= Decl +
2 Decl ::= VariableDecl
3 Decl ::= FunctionDecl
4 Decl ::= ConstDecl
5 Decl ::= ClassDecl
6 Decl ::= InterfaceDecl
7 VariableDecl ::= Variable ;
8 Variable ::= Type ident
9 ConstDecl ::= static ConstType ident ;
10 ConstType ::= int
11 ConstType ::= double
12 ConstType ::= boolean
13 ConstType ::= string
14 Type ::= int
15 Type ::= double
16 Type ::= boolean
17 Type ::= string
18 Type ::= ident
19 Type ::= Type[ ]
20 FunctionDecl ::= Type ident ( Formals ) StmtBlock
21 FunctionDecl ::= void ident ( Formals ) StmtBlock
22 Formals ::= Variable, Formals
23 Formals ::= Variable
24 ClassDecl ::= class ident (extends ident)? (implements (ident)+ (, ident)* )? { Field* }
25 Field ::= VariableDecl
26 Field ::= FunctionDecl
27 Field ::= ConstDecl
28 InterfaceDecl ::= interface ident { Prototype* }
29 Prototype ::= Type ident (Formals) ;
```

```
30 Prototype ::= void ident (Formals);
31 StmtBlock ::= { VariableDecl* ConstDecl* Stmt* }
32 Stmt ::= Expr ? ;
33 Stmt ::= IfStmt
34 Stmt ::= WhileStmt
35 Stmt ::= ForStmt
36 Stmt ::= BreakStmt
37 Stmt ::= ReturnStmt
38 Stmt ::= PrintStmt
39 Stmt ::= StmtBlock
40 IfStmt ::= if (Expr) Stmt (else Stmt)?
41 WhileStmt ::= while (Expr) Stmt
42 ForStmt ::= for (Expr; Expr; Expr) Stmt
43 ReturnStmt ::= return Expr;
44 BreakStmt ::= break ;
45 PrintStmt ::= System.out.println(Expr+ ,) ;
46 Expr ::= LValue = Expr
47 Expr ::= Constant
48 Expr ::= Lvalue
49 Expr ::= this
50 Expr ::= (Expr)
51 Expr ::= Expr - Expr
52 Expr ::= Expr | Expr
53 Expr ::= Expr % Expr
54 Expr ::= - Expr
55 Expr ::= Expr > Expr
56 Expr ::= Expr >= Expr
57 Expr ::= Expr != Expr
58 Expr ::= Expr || Expr
59 Expr ::= ! Expr
60 Expr ::= New (ident)
61 LValue ::= ident
62 LValue ::= Expr .ident
63 Constant ::= intConstant
```

```
64 Constant ::= doubleConstant
65 Constant ::= booleanConstant
66 Constant ::= stringConstant
67 Constant ::= null
```

A => α.Bβ

No.	ESTADOS	LOOKAHEAD	IR_A
I0	Program' ::= .Program	\$	IrA(I0, Program) = I1
	Program ::= .Decl +	\$	IrA(I0, Decl) = I2
	Decl ::= .VariableDecl	\$	IrA(I0, VariableDecl) = I3
	Decl ::= .FunctionDecl	\$	IrA(I0, FunctionDecl) = I4
	Decl ::= .ConstDecl	\$	IrA(I0, ConstDecl) = I5
	Decl ::= .ClassDecl	\$	IrA(I0, ClassDecl) = I6
	Decl ::= .InterfaceDecl	\$	IrA(I0, InterfaceDecl) = I7
	VariableDecl ::= .Variable ;	\$	IrA(I0, Variable) = I8
	FunctionDecl ::= .Type ident (Formals) StmtBlock	\$	IrA(I0, Type) = I9
	FunctionDecl ::= . void ident (Formals) StmtBlock	\$	IrA(I0, void) = I10
	ConstDecl ::= . static ConstType ident ;	\$	IrA(I0, static) = I11
	ClassDecl ::= . class ident (extends ident)? (implements (ident)+ (, ident)*)? { Field* }	\$	IrA(I0, class) = I12
	InterfaceDecl ::= . interface ident { Prototype* }	\$	IrA(I0, interface) = I13
	Variable ::= .Type ident	;	IrA(I0, Type) = I9
	Type ::= . int	ident, []	IrA(I0, int) = I14
	Type ::= . double	ident, []	IrA(I0, double) = I15
	Type ::= . boolean	ident, []	IrA(I0, boolean) = I16
	Type ::= . string	ident, []	IrA(I0, string) = I17
	Type ::= . ident	ident, []	IrA(I0, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I0, Type) = I9
I1	Program' ::= Program.	\$	ACEPTAR
I2	Program ::= Decl. +	\$	r1

I3	Decl ::= VariableDecl.	\$	r2
I4	Decl ::= FunctionDecl.	\$	r3
I5	Decl ::= ConstDecl.	\$	r4
I6	Decl ::= ClassDecl.	\$	r5
I7	Decl ::= .InterfaceDecl	\$	r6
I8	VariableDecl ::= Variable . ;	\$	IrA(I8, ;) = I19
I9	FunctionDecl ::= Type .ident (Formals) StmtBlock	\$	IrA(I9, ident) = I20
	Variable ::= Type .ident	;	IrA(I9, ident) = I20
	Type ::= Type.[]	ident, []	IrA(I9, []) = I21
I10	FunctionDecl ::= void .ident (Formals) StmtBlock	\$	IrA(I10, ident) = I22
I11	ConstDecl ::= static .ConstType ident ;	\$	IrA(I11, ConstType) = I23
	ConstType ::= .int	ident	IrA(I11, int) = I24
	ConstType ::= .double	ident	IrA(I11, double) = I25
	ConstType ::= .boolean	ident	IrA(I11, boolean) = I26
	ConstType ::= .string	ident	IrA(I11, string) = I27
I12	ClassDecl ::= class .ident (extends ident)? (implements (ident)+ (, ident)*)? { Field* }	\$	IrA(I12, ident) = I28
I13	InterfaceDecl ::= interface .ident { Prototype* }	\$	IrA(I13, ident) = I29
I14	Type ::= int.	ident, []	
I15	Type ::= double.	ident, []	
I16	Type ::= boolean.	ident, []	

I17	Type ::= string .	ident, []	
I18	Type ::= ident .	ident, []	
I19	VariableDecl ::= Variable ;.	\$	
I20	FunctionDecl ::= Type ident .(Formals) StmtBlock	\$	IrA(I20, () = I30
	Variable ::= Type ident .	;	
I21	Type ::= Type[].	ident, []	
I22	FunctionDecl ::= void ident .(Formals) StmtBlock	\$	IrA(I22, () = I31
I23	ConstDecl ::= static ConstType . ident ;	\$	IrA(I23, ident) = I32
I24	ConstType ::= int .	ident	
I25	ConstType ::= double .	ident	
I26	ConstType ::= boolean .	ident	
I27	ConstType ::= string .	ident	
I28	ClassDecl ::= class ident .(extends ident)? (implements (ident)+ (, ident)*)? { Field* }	\$	IrA(I28, () = I33
I29	InterfaceDecl ::= interface ident .{ Prototype* }	\$	IrA(I29, { } = I34
I30	FunctionDecl ::= Type ident (.Formals) StmtBlock	\$	IrA(I30, Formals) = I35
	Formals ::= .Variable, Formals)	IrA(I30, Variable) = I36
	Formals ::= .Variable)	IrA(I30, Variable) = I36
	Variable ::= .Type ident	, ,)	IrA(I30, Type) = I37
	Type ::= . int	ident, []	IrA(I30, int) = I14
	Type ::= . double	ident, []	IrA(30, double) = I15

	Type ::= .boolean	ident, []	IrA(I30, boolean) = I16
	Type ::= .string	ident, []	IrA(I30, string) = I17
	Type ::= .ident	ident, []	IrA(I30, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I30, Type) = I9
I31	FunctionDecl ::= void ident (.Formals) StmtBlock	\$	IrA(I31, Formals) = I38
	Formals ::= .Variable , Formals)	IrA(I30, Variable) = I36
	Formals ::= .Variable)	IrA(I30, Variable) = I36
	Variable ::= .Type ident	, ,)	IrA(I30, Type) = I37
	Type ::= .int	ident, []	IrA(I30, int) = I14
	Type ::= .double	ident, []	IrA(30, double) = I15
	Type ::= .boolean	ident, []	IrA(I30, boolean) = I16
	Type ::= .string	ident, []	IrA(I30, string) = I17
	Type ::= .ident	ident, []	IrA(I30, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I30, Type) = I9
I32	ConstDecl ::= static ConstType ident . ;	\$	IrA(I32, ;) = I39
I33	ClassDecl ::= class ident (.extends ident)? (implements (ident)+ (, ident)*)? { Field* }	\$	IrA(I33, extends) = I40
I34	InterfaceDecl ::= interface ident { .Prototype* }	\$	IrA(I34, Prototype) = I41
	Prototype ::= .Type ident (Formals) ;	}	IrA(I34, Type) = I42
	Prototype ::= .void ident (Formals);	}	IrA(I34, void) = I43
	Type ::= int	ident, []	IrA(I34, int) = I14
	Type ::= double	ident, []	IrA(34, double) = I15
	Type ::= boolean	ident, []	IrA(I34, boolean) = I16
	Type ::= string	ident, []	IrA(I34, string) = I17
	Type ::= ident	ident, []	IrA(I34, ident) = I18
	Type ::= Type[]	ident, []	IrA(I34, Type) = I9
I35	FunctionDecl ::= Type ident (Formals .) StmtBlock	\$	IrA(I35,)) = I44
I36	Formals ::= Variable., Formals)	IrA(I36, ,) = I45
	Formals ::= Variable.)	

I37	Variable ::= Type .ident	, ,)	IrA(I37, ident) = I46
	Type ::= Type.[]	ident, []	IrA(I37, []) = I318
I38	FunctionDecl ::= void ident (Formals .) StmtBlock	\$	IrA(I38,)) = I47
I39	ConstDecl ::= static ConstType ident ;.	\$	
I40	ClassDecl ::= class ident (extends .ident)? (implements (ident)+ (, ident)*)? { Field* }	\$	IrA(I40, ident) = I48
I41	InterfaceDecl ::= interface ident { Prototype* .}	\$	IrA(I41, }) = I49
I42	Prototype ::= Type .ident (Formals) ;	}	IrA(I42, ident) = I50
I43	Prototype ::= void .ident (Formals);	}	IrA(I43, ident) = I51
I44	FunctionDecl ::= Type ident (Formals). StmtBlock	\$	IrA(I44, StmtBlock) = I52
	StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* }	\$	IrA(I44, {) = I53
I45	Formals ::= Variable, .Formals)	IrA(I45, Formals) = I54
	Formals ::= .Variable, Formals)	IrA(I45, Variable) = I36
	Formals ::= .Variable)	IrA(I45, Variable) = I36
	Variable ::= .Type ident	, ,)	IrA(I45, Type) = I37
	Type ::= .int	ident, []	IrA(I45, int) = I14
	Type ::= .double	ident, []	IrA(45, double) = I15
	Type ::= .boolean	ident, []	IrA(I45, boolean) = I16
	Type ::= .string	ident, []	IrA(I45, string) = I17
	Type ::= .ident	ident, []	IrA(I45, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I45, Type) = I9
I46	Variable ::= Type ident.	, ,)	
I47	FunctionDecl ::= void ident (Formals) .StmtBlock	\$	IrA(I47, StmtBlock) = I55
	StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* }	\$	IrA(I47, {) = I53

I48	ClassDecl ::= class ident (extends ident)? (.implements (ident)+ (, ident)*)? { Field* }	\$	IrA(I48, implements) = I56
I49	InterfaceDecl ::= interface ident { Prototype* }.	\$	
I50	Prototype ::= Type ident . (Formals) ;	}	IrA(I50, () = I57
I51	Prototype ::= void ident . (Formals);	}	IrA(I51, () = I58
I52	FunctionDecl ::= Type ident (Formals) StmtBlock.	\$	
I53	StmtBlock ::= { . VariableDecl* ConstDecl* Stmt* }	\$	IrA(I53, VariableDecl) = I59
	VariableDecl ::= . Variable ;	static	IrA(I53, Variable) = I60
	Variable ::= . Type ident	;	IrA(I53, Type) = I9
	Type ::= .int	ident, []	IrA(I53, int) = I14
	Type ::= .double	ident, []	IrA(I53, double) = I15
	Type ::= .boolean	ident, []	IrA(I53, boolean) = I16
	Type ::= .string	ident, []	IrA(I53, string) = I17
	Type ::= .ident	ident, []	IrA(I53, ident) = I18
	Type ::= . Type[]	ident, []	IrA(I53, Type) = I9
I54	Formals ::= Variable, Formals.)	
I55	FunctionDecl ::= void ident (Formals) StmtBlock.	\$	
I56	ClassDecl ::= class ident (extends ident)? (implements .ident + ,)? { Field* }	\$	IrA(I56, ident) = I61
I57	Prototype ::= Type ident . (Formals) ;	}	IrA(I57, Formals) = I62
	Formals ::= . Variable, Formals)	IrA(I57, Variable) = I36
	Formals ::= . Variable)	IrA(I57, Variable) = I36
	Variable ::= Type ident	, ,)	IrA(I57, Type) = I37
	Type ::= int	ident, []	IrA(I57, int) = I14
	Type ::= double	ident, []	IrA(I57, double) = I15
	Type ::= boolean	ident, []	IrA(I57, boolean) = I16

	Type ::= string	ident, []	IrA(I57, string) = I17
	Type ::= ident	ident, []	IrA(I57, ident) = I18
	Type ::= Type[]	ident, []	IrA(I57, Type) = I9
I58	Prototype ::= void ident (.Formals);	}	IrA(I58, Formals) = I63
	Formals ::= .Variable, Formals)	IrA(I58, Variable) = I36
	Formals ::= .Variable)	IrA(I58, Variable) = I36
	Variable ::= Type ident	, ,)	IrA(I58, Type) = I37
	Type ::= int	ident, []	IrA(I58, int) = I14
	Type ::= double	ident, []	IrA(I58, double) = I15
	Type ::= boolean	ident, []	IrA(I58, boolean) = I16
	Type ::= string	ident, []	IrA(I58, string) = I17
	Type ::= ident	ident, []	IrA(I58, ident) = I18
	Type ::= Type[]	ident, []	IrA(I58, Type) = I9
I59	StmtBlock ::= { VariableDecl* .ConstDecl* Stmt* }	\$	IrA(I59, ConstDecl) = I64
	ConstDecl ::= . static ConstType ident ;	if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	IrA(I59, static) = I65
I60	VariableDecl ::= Variable . ;	static	IrA(I60, ;) = I66
I61	ClassDecl ::= class ident (extends ident)? (implements ident+ . ,)? { Field* }	\$	IrA(I61, ,) = I67
I62	Prototype ::= Type ident (Formals .) ;	}	IrA(I62,)) = I68
I63	Prototype ::= void ident (Formals.);	}	IrA(I63,)) = I69
I64	StmtBlock ::= { VariableDecl* ConstDecl* .Stmt* }	\$	IrA(I64, Stmt) = I70
	Stmt ::= .Expr ? ;	}	IrA(I64, Expr) = I71
	Stmt ::= .IfStmt	}	IrA(I64, IfStmt) = I72
	Stmt ::= .WhileStmt	}	IrA(I64, WhileStmt) = I73

	Stmt ::= .ForStmt	}	IrA(I64, ForStmt) = 174
	Stmt ::= .BreakStmt	}	IrA(I64, BreakStmt) = 175
	Stmt ::= .ReturnStmt	}	IrA(I64, ReturnStmt) = 176
	Stmt ::= .PrintStmt	}	IrA(I64, PrintStmt) = 177
	Stmt ::= .StmtBlock	}	IrA(I64, StmtBlock) = 178
	StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* }	}	IrA(I64, { }) = 179
	IfStmt ::= . if (Expr) Stmt (else Stmt)?	}	IrA(I64, if) = 180
	WhileStmt ::= . while (Expr) Stmt	}	IrA(I64, while) = 181
	ForStmt ::= .for (Expr; Expr; Expr) Stmt	}	IrA(I64, for) = 182
	ReturnStmt ::= . return Expr;	}	IrA(I64, return) = 183
	BreakStmt ::= . break ;	}	IrA(I64, break) = 184
	PrintStmt ::= .System.out.println(Expr+ ,) ;	}	IrA(I64, System.out.println) = 185
	Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = 186
	Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = 187
	Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = 186
	Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = 188
	Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, ()) = 189
	Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = 190
	Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I64, !) = 191
	Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I64, New) = 192
	LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I64, ident) = 193
	LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171
	Constant ::= .intConstant	;; -, , % , > , >= , != , , .	IrA(I64, intConstant) = 194
	Constant ::= .doubleConstant	;; -, , % , > , >= , != , , .	IrA(I64, doubleConstant) = 195
	Constant ::= . booleanConstant	;; -, , % , > , >= , != , , .	IrA(I64, booleanConstant) = 196
	Constant ::= .stringConstant	;; -, , % , > , >= , != , , .	IrA(I64, stringConstant) = 197
	Constant ::= .null	;; -, , % , > , >= , != , , .	IrA(I64, null}) = 198

I65	ConstDecl ::= static .ConstType ident ;	if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	IrA(I65, ConstType) = I99
	ConstType ::= .int	ident	IrA(I65, int) = I100
	ConstType ::= .double	ident	IrA(I65, double) = I101
	ConstType ::= .boolean	ident	IrA(I65, boolean) = I102
	ConstType ::= .string	ident	IrA(I65, string) = I103
I66	VariableDecl ::= Variable ;.	static	IrA(I67, { }) = I104 IrA(I68, ;) = I105 IrA(I69, ;) = I106 IrA(I70, { }) = I107
I67	ClassDecl ::= class ident (extends ident)? (implements ident+ ,)? .{ Field* }	\$	
I68	Prototype ::= Type ident (Formals) . ;	}	
I69	Prototype ::= void ident (Formals) . ;	}	
I70	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* . }	\$	
I71	Stmt ::= Expr ? . ;	}	IrA(I71, ;) = I108
	Expr ::= Expr .- Expr	;, -, , %, >, >=, !=, , .	IrA(I71, -) = I109
	Expr ::= Expr . Expr	;, -, , %, >, >=, !=, , .	IrA(I71,) = I110
	Expr ::= Expr . % Expr	;, -, , %, >, >=, !=, , .	IrA(I71, %) = I111
	Expr ::= Expr . > Expr	;, -, , %, >, >=, !=, , .	IrA(I71, >) = I112
	Expr ::= Expr . >= Expr	;, -, , %, >, >=, !=, , .	IrA(I71, >=) = I113
	Expr ::= Expr . != Expr	;, -, , %, >, >=, !=, , .	IrA(I71, !=) = I114
	Expr ::= Expr . Expr	;, -, , %, >, >=, !=, , .	IrA(I71,) = I115
	LValue ::= Expr . ident	;, -, , %, >, >=, !=, , .	IrA(I71, .) = I116
I72	Stmt ::= IfStmt.	}	
I73	Stmt ::= WhileStmt.	}	

I74	Stmt ::= ForStmt.	}	
I75	Stmt ::= BreakStmt.	}	
I76	Stmt ::= ReturnStmt.	}	
I77	Stmt ::= PrintStmt.	}	
I78	Stmt ::= StmtBlock.	}	
I79	StmtBlock ::= { .VariableDecl* ConstDecl* Stmt* }	}	IrA(I79, VariableDecl) = 117
	VariableDecl ::= .Variable ;	static	IrA(I79, Variable) = 160
	Variable ::= .Type ident	;	IrA(I79, Type) = 19
	Type ::= . int	ident, []	IrA(I79, int) = 114
	Type ::= . double	ident, []	IrA(I79, double) = 115
	Type ::= . boolean	ident, []	IrA(I79, boolean) = 116
	Type ::= . string	ident, []	IrA(I79, string) = 117
	Type ::= . ident	ident, []	IrA(I79, ident) = 118
	Type ::= .Type[]	ident, []	IrA(I79, Type) = 19
I80	IfStmt ::= if .(Expr) Stmt (else Stmt)?	}	IrA(I80, () = 118
I81	WhileStmt ::= while .(Expr) Stmt	}	IrA(I81, () = 119
I82	ForStmt ::= for .(Expr; Expr; Expr) Stmt	}	IrA(I82, () = 120
I83	ReturnStmt ::= return .Expr;	}	IrA(I80, Expr) = 121
	Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = 186
	Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = 187
	Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = 186
	Expr ::= . this	;; -, , % , > , >= , != , , .	IrA(I64, this) = 188
	Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, () = 189
	Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = 171

I84	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71	
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71	
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90	
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71	
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71	
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71	
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71	
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91	
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92	
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93	
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71	
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94	
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95	
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96	
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97	
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98	
		BreakStmt ::= break . ;	}	IrA(I80, ;) = I22
	I85			
		PrintStmt ::= System.out.println . (Expr+ ,) ;	}	IrA(I80, () = I23
I86	Expr ::= LValue . = Expr	; , - , , % , > , >= , != , , .	IrA(I86, =) = I24	
	Expr ::= Lvalue.	; , - , , % , > , >= , != , , .		
I87	Expr ::= Constant.	; , - , , % , > , >= , != , , .		
I88	Expr ::= this.	; , - , , % , > , >= , != , , .		
I89	Expr ::= (.Expr)	; , - , , % , > , >= , != , , .	IrA(I89, Expr) = I125	
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126	
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127	
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126	
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128	
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129	

Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139

I90

Expr ::= - .Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I140
Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I90, LValue) = I86
Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I90, Constant) = I87
Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I90, LValue) = I86
Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I90, this) = I88
Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I90, () = I89
Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I90, -) = I90
Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I90, !) = I91
Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I90, New) = I92

LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I90, ident) = I93
LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I90, Expr) = I71
Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I90, intConstant) = I94
Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I90, doubleConstant) = I95
Constant ::= . booleanConstant	; , - , , % , > , >= , != , , .	IrA(I89, booleanConstant) = I96
Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I90, stringConstant) = I97
Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I90, null) = I98

I91

Expr ::= ! .Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I141
Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I91, this) = I88
Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I91, () = I89
Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I91, -) = I90
Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91
Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92
LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93
LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
Constant ::= . booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98

I92

Expr ::= New .(ident)	; , - , , % , > , >= , != , , .	IrA(I92, () = I142
------------------------------	--------------------------------------	---------------------

I93	LValue ::= ident .	; , - , , % , > , >= , != , , .	
I94	Constant ::= intConstant .	; , - , , % , > , >= , != , , .	
I95	Constant ::= doubleConstant .	; , - , , % , > , >= , != , , .	
I96	Constant ::= booleanConstant .	; , - , , % , > , >= , != , , .	
I97	Constant ::= stringConstant .	; , - , , % , > , >= , != , , .	
I98	Constant ::= null .	; , - , , % , > , >= , != , , .	
I99	ConstDecl ::= static ConstType . ident ;	if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	IrA(I99, ident) = I143
I100	ConstType ::= int .	ident	
I101	ConstType ::= double .	ident	
I102	ConstType ::= boolean .	ident	
I103	ConstType ::= string .	ident	
I104	ClassDecl ::= class ident (extends ident)? (implements ident+ ,)? { .Field* }	\$	IrA(I104, Field) = I144
	Field ::= .VariableDecl	}	IrA(I104, VariableDecl) = I145
	Field ::= .FunctionDecl	}	IrA(I104, FunctionDecl) = I146
	Field ::= .ConstDecl	}	IrA(I104, ConstDecl) = I147
	VariableDecl ::= .Variable ;	}	IrA(I104, Variable) = I148
	Variable ::= .Type ident	;	IrA(I104, Type) = I9
	Type ::= .int	ident, []	IrA(I104, int) = I14
	Type ::= .double	ident, []	IrA(I104, double) = I15

	Type ::= .boolean	ident, []	IrA(I104, boolean) = I16
	Type ::= .string	ident, []	IrA(I104, string) = I17
	Type ::= .ident	ident, []	IrA(I104, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I104, Type) = I9
	FunctionDecl ::= .Type ident (Formals) StmtBlock	}	IrA(I104, Type) = I149
	FunctionDecl ::= .void ident (Formals) StmtBlock	}	IrA(I104, void) = I150
	ConstDecl ::= .static ConstType ident ;	}	IrA(I104, static) = I151
I105	Prototype ::= Type ident (Formals) ; .	}	
I106	Prototype ::= void ident (Formals) ; .	}	
I107	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* }.	\$	
I108	Stmt ::= Expr ? ; .	}	
I109	Expr ::= Expr - .Expr	;; -, , % , > , >= , != , , .	IrA(I109, Expr) = I152
	Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I91, Constant) = I87
	Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I91, this) = I88
	Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I91, () = I89
	Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I91, -) = I90
	Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I91, !) = I91
	Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I91, New) = I92
	LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I91, ident) = I93
	LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I91, Expr) = I71

	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98
I110	Expr ::= Expr .Expr	; , - , , % , > , >= , != , , .	IrA(I110, Expr) = I153
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I91, this) = I88
	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I91, () = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I91, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98
I111	Expr ::= Expr % .Expr	; , - , , % , > , >= , != , , .	IrA(I111, Expr) = I154
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86

Expr ::= .this	;;-, ,%,>,>=,!=, ,.	IrA(I91, this) = I88
Expr ::= .(Expr)	;;-, ,%,>,>=,!=, ,.	IrA(I91, () = I89
Expr ::= .Expr - Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr % Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .- Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, -) = I90
Expr ::= .Expr > Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr >= Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr != Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .! Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, !) = I91
Expr ::= .New (ident)	;;-, ,%,>,>=,!=, ,.	IrA(I91, New) = I92
LValue ::= .ident	;;-, ,%,>,>=,!=, ,.	IrA(I91, ident) = I93
LValue ::= .Expr .ident	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Constant ::= .intConstant	;;-, ,%,>,>=,!=, ,.	IrA(I91, intConstant) = I94
Constant ::= .doubleConstant	;;-, ,%,>,>=,!=, ,.	IrA(I91, doubleConstant) = I95
Constant ::= .booleanConstant	;;-, ,%,>,>=,!=, ,.	IrA(I91, booleanConstant) = I96
Constant ::= .stringConstant	;;-, ,%,>,>=,!=, ,.	IrA(I91, stringConstant) = I97
Constant ::= .null	;;-, ,%,>,>=,!=, ,.	IrA(I91, null) = I98

I112

Expr ::= Expr > .Expr	;;-, ,%,>,>=,!=, ,.	IrA(I112, Expr) = I155
Expr ::= .LValue = Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, LValue) = I86
Expr ::= .Constant	;;-, ,%,>,>=,!=, ,.	IrA(I91, Constant) = I87
Expr ::= .Lvalue	;;-, ,%,>,>=,!=, ,.	IrA(I91, LValue) = I86
Expr ::= .this	;;-, ,%,>,>=,!=, ,.	IrA(I91, this) = I88
Expr ::= .(Expr)	;;-, ,%,>,>=,!=, ,.	IrA(I91, () = I89
Expr ::= .Expr - Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr % Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .- Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, -) = I90
Expr ::= .Expr > Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr >= Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr != Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	;;-, ,%,>,>=,!=, ,.	IrA(I91, Expr) = I71

Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91
Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92
LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93
LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98

I113

Expr ::= Expr >= .Expr	; , - , , % , > , >= , != , , .	IrA(I113, Expr) = I156
Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I91, this) = I88
Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I91, () = I89
Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I91, -) = I90
Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91
Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92
LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93
LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98

I114	Expr ::= Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I114, Expr) = I157
	Expr ::= Expr.LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= Expr.Constant	; , - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
	Expr ::= Expr.Lvalue	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= Expr.this	; , - , , % , > , >= , != , , .	IrA(I91, this) = I88
	Expr ::= Expr.(Expr)	; , - , , % , > , >= , != , , .	IrA(I91, () = I89
	Expr ::= Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, -) = I90
	Expr ::= Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr ! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91
	Expr ::= Expr.New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92
	LValue ::= Expr.ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93
	LValue ::= Expr.Expr.ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Constant ::= Expr.intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
	Constant ::= Expr.doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
	Constant ::= Expr.booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
	Constant ::= Expr.stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
	Constant ::= Expr.null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98
I115	Expr ::= Expr Expr	; , - , , % , > , >= , != , , .	IrA(I115, Expr) = I158
	Expr ::= Expr.LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= Expr.Constant	; , - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
	Expr ::= Expr.Lvalue	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= Expr.this	; , - , , % , > , >= , != , , .	IrA(I91, this) = I88
	Expr ::= Expr.(Expr)	; , - , , % , > , >= , != , , .	IrA(I91, () = I89
	Expr ::= Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, -) = I90

	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98
I116	LValue ::= Expr . .ident	; , - , , % , > , >= , != , , .	IrA(I116, ident) = I159
I117	StmtBlock ::= { VariableDecl* .ConstDecl* Stmt* }	}	IrA(I117, ConstDecl) = I160
	ConstDecl ::= .static ConstType ident ;	if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	IrA(I117, static) = I65
I118	IfStmt ::= if (.Expr) Stmt (else Stmt)?	}	IrA(I118, Expr) = I161
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130

	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I119	WhileStmt ::= while (.Expr) Stmt	}	IrA(I119, Expr) = I162
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137

	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I120	ForStmt ::= for (.Expr; Expr; Expr) Stmt	}	IrA(I120, Expr) = I163
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
	Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I91, this) = I88
	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I91, () = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I91, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98
I121	ReturnStmt ::= return Expr ;	}	IrA(I121, ;) = I164
I122	BreakStmt ::= break ;.	}	
I123	PrintStmt ::= System.out.println (.Expr+ ,) ;	}	IrA(I123, Expr) = I165
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167

Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
Constant ::= . booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179

I124

Expr ::= LValue = .Expr	;; - , , % , > , >= , != , , .	IrA(I124, Expr) = I180
Expr ::= .LValue = Expr	;; - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
Expr ::= .Constant	;; - , , % , > , >= , != , , .	IrA(I91, Constant) = I87
Expr ::= .Lvalue	;; - , , % , > , >= , != , , .	IrA(I91, LValue) = I86
Expr ::= .this	;; - , , % , > , >= , != , , .	IrA(I91, this) = I88
Expr ::= .(Expr)	;; - , , % , > , >= , != , , .	IrA(I91, ()) = I89
Expr ::= .Expr - Expr	;; - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr Expr	;; - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr % Expr	;; - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .- Expr	;; - , , % , > , >= , != , , .	IrA(I91, -) = I90
Expr ::= .Expr > Expr	;; - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr >= Expr	;; - , , % , > , >= , != , , .	IrA(I91, Expr) = I71
Expr ::= .Expr != Expr	;; - , , % , > , >= , != , , .	IrA(I91, Expr) = I71

	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71	
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I91, !) = I91	
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I91, New) = I92	
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I91, ident) = I93	
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I91, Expr) = I71	
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I94	
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I95	
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I96	
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I97	
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I91, null) = I98	
I125	Expr ::= (Expr.)	; , - , , % , > , >= , != , , .	IrA(I125,)) = I181	
I126	Expr ::= LValue .= Expr	- , , % , > , >= , != , , . ,)	IrA(I126, =) = I182	
I127	Expr ::= Constant.	- , , % , > , >= , != , , . ,)		
I128	Expr ::= this.	- , , % , > , >= , != , , . ,)		
I129	Expr ::= (.Expr)	- , , % , > , >= , != , , . ,)	IrA(I129, Expr) = I183	
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126	
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127	
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126	
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128	
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129	
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130	
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130	
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130	
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131	
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130	
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130	
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130	
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130	
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132	

	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I130	Expr ::= Expr .- Expr	- , , % , > , >= , != , , . ,)	IrA(I130, -) = I184
	Expr ::= Expr . Expr	- , , % , > , >= , != , , . ,)	IrA(I130,) = I185
	Expr ::= Expr .% Expr	- , , % , > , >= , != , , . ,)	IrA(I130, %) = I186
	Expr ::= Expr .> Expr	- , , % , > , >= , != , , . ,)	IrA(I130, >) = I187
	Expr ::= Expr .>= Expr	- , , % , > , >= , != , , . ,)	IrA(I130, >=) = I188
	Expr ::= Expr .!= Expr	- , , % , > , >= , != , , . ,)	IrA(I130, !=) = I189
	Expr ::= Expr . Expr	- , , % , > , >= , != , , . ,)	IrA(I130,) = I190
	LValue ::= Expr . .ident	- , , % , > , >= , != , , . ,)	IrA(I130, .) = I191
I131	Expr ::= - .Expr	- , , % , > , >= , != , , . ,)	IrA(I131, Expr) = I192
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133

	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I132	Expr ::= ! .Expr	- , , % , > , >= , != , , . ,)	IrA(I132, Expr) = I193
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I133	Expr ::= New .(ident)	- , , % , > , >= , != , , . ,)	IrA(I133, () = I194

I134	LValue ::= ident .	- , , % , > , >= , != , , . ,)	
I135	Constant ::= intConstant .	- , , % , > , >= , != , , . ,)	
I136	Constant ::= doubleConstant .	- , , % , > , >= , != , , . ,)	
I137	Constant ::= booleanConstant .	- , , % , > , >= , != , , . ,)	
I138	Constant ::= stringConstant .	- , , % , > , >= , != , , . ,)	
I139	Constant ::= null .	- , , % , > , >= , != , , . ,)	
I140	Expr ::= - Expr.	; , - , , % , > , >= , != , , .	
I141	Expr ::= ! Expr.	; , - , , % , > , >= , != , , .	
I142	Expr ::= New (.ident)	; , - , , % , > , >= , != , , .	IrA(I142, ident) = I195
I143	ConstDecl ::= static ConstType ident . ;	if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	IrA(I143, ;) = I196
I144	ClassDecl ::= class ident (extends ident)? (implements ident+ ,)? { Field* . }	\$	IrA(I144, }) = I197
I145	Field ::= VariableDecl.	}	
I146	Field ::= FunctionDecl.	}	
I147	Field ::= ConstDecl.	}	
I148	VariableDecl ::= Variable . ;	}	IrA(I148, ;) = I198

I149	FunctionDecl ::= Type . ident (Formals) StmtBlock	}	IrA(I149, ident) = I199
I150	FunctionDecl ::= void . ident (Formals) StmtBlock	}	IrA(I150, ident) = I200
I151	ConstDecl ::= static .ConstType ident ;	}	IrA(I151, ConstType) = I201
	ConstType ::= . int	ident	IrA(I151, int) = I100
	ConstType ::= . double	ident	IrA(I151, double) = I101
	ConstType ::= . boolean	ident	IrA(I151, boolean) = I102
	ConstType ::= . string	ident	IrA(I151, string) = I103
I152	Expr ::= Expr - Expr.	;; -, , % , > , >= , != , , .	
I153	Expr ::= Expr Expr.	;; -, , % , > , >= , != , , .	
I154	Expr ::= Expr % Expr.	;; -, , % , > , >= , != , , .	
I155	Expr ::= Expr > Expr.	;; -, , % , > , >= , != , , .	
I156	Expr ::= Expr >= Expr.	;; -, , % , > , >= , != , , .	
I157	Expr ::= Expr != Expr.	;; -, , % , > , >= , != , , .	
I158	Expr ::= Expr Expr.	;; -, , % , > , >= , != , , .	
I159	LValue ::= Expr . ident	;; -, , % , > , >= , != , , .	IrA(I159, ident) = I202
I160	StmtBlock ::= { VariableDecl* ConstDecl* .Stmt* }	}	IrA(I160, Stmt) = I203
	Stmt ::= .Expr ? ;	}	IrA(I64, Expr) = I71
	Stmt ::= .IfStmt	}	IrA(I64, IfStmt) = I72
	Stmt ::= .WhileStmt	}	IrA(I64, WhileStmt) = I73
	Stmt ::= .ForStmt	}	IrA(I64, ForStmt) = I74
	Stmt ::= .BreakStmt	}	IrA(I64, BreakStmt) = I75
	Stmt ::= .ReturnStmt	}	IrA(I64, ReturnStmt) = I76
	Stmt ::= .PrintStmt	}	IrA(I64, PrintStmt) = I77

	Stmt ::= .StmtBlock	}	IrA(I64, StmtBlock) = I78
	StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* }	}	IrA(I64, { }) = I79
	IfStmt ::= . if (Expr) Stmt (else Stmt)?	}	IrA(I64, if) = I80
	WhileStmt ::= . while (Expr) Stmt	}	IrA(I64, while) = I81
	ForStmt ::= .for (Expr; Expr; Expr) Stmt	}	IrA(I64, for) = I82
	ReturnStmt ::= . return Expr;	}	IrA(I64, return) = I83
	BreakStmt ::= . break ;	}	IrA(I64, break) = I84
	PrintStmt ::= .System.out.println(Expr+ ,) ;	}	IrA(I64, System.out.println) = I85
	Expr ::= .LValue = Expr	;; -, , % , >, >=, !=, , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	;; -, , % , >, >=, !=, , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	;; -, , % , >, >=, !=, , .	IrA(I64, LValue) = I86
	Expr ::= .this	;; -, , % , >, >=, !=, , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	;; -, , % , >, >=, !=, , .	IrA(I64, ()) = I89
	Expr ::= .Expr - Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	;; -, , % , >, >=, !=, , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	;; -, , % , >, >=, !=, , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	;; -, , % , >, >=, !=, , .	IrA(I64, New) = I92
	LValue ::= .ident	;; -, , % , >, >=, !=, , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	;; -, , % , >, >=, !=, , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	;; -, , % , >, >=, !=, , .	IrA(I64, doubleConstant) = I95
	Constant ::= . booleanConstant	;; -, , % , >, >=, !=, , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	;; -, , % , >, >=, !=, , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	;; -, , % , >, >=, !=, , .	IrA(I64, null}) = I98
I161			
	IfStmt ::= if (Expr.) Stmt (else Stmt)?	}	IrA(I161,)) = I204
I162			
	WhileStmt ::= while (Expr.) Stmt	}	IrA(I162,)) = I205

I163	ForStmt ::= for (Expr .; Expr; Expr) Stmt	}	IrA(I163, ;) = I206
I164	ReturnStmt ::= return Expr ; .	}	
I165	PrintStmt ::= System.out.println (Expr+ .,) ;	}	IrA(I165, ,) = I207
I166	Expr ::= LValue . = Expr	(,) , - , , % , > , >= , != , , .	IrA(I166, =) = I208
	Expr ::= Lvalue.	(,) , - , , % , > , >= , != , , .	
I167	Expr ::= Constant.	(,) , - , , % , > , >= , != , , .	
I168	Expr ::= this .	(,) , - , , % , > , >= , != , , .	
I169	Expr ::= (.Expr)	(,) , - , , % , > , >= , != , , .	IrA(I169, Expr) = I209
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136

	Constant ::= booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I170	Expr ::= Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I170, -) = I210
	Expr ::= Expr . Expr	(,) , - , , % , > , >= , != , , .	IrA(I170,) = I211
	Expr ::= Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I170, %) = I212
	Expr ::= Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I170, >) = I213
	Expr ::= Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I170, >=) = I214
	Expr ::= Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I170, !=) = I215
	Expr ::= Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I170,) = I216
	LValue ::= Expr . ident	(,) , - , , % , > , >= , != , , .	IrA(I170, .) = I217
I171	Expr ::= - Expr	(,) , - , , % , > , >= , != , , .	IrA(I171, Expr) = I218
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
	Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
	Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= ! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
	LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
	LValue ::= .Expr . ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
	Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
	Constant ::= booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177

	Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
	Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179
I172	Expr ::= ! .Expr	(,) , - , , % , > , >= , != , , .	IrA(I172, Expr) = I219
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
	Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, () = I169
	Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
	LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
	LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
	Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
	Constant ::= .booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
	Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
	Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179
I173	Expr ::= New .(ident)	(,) , - , , % , > , >= , != , , .	IrA(I173, () = I220
I174	LValue ::= ident.	(,) , - , , % , > , >= , != , , .	
I175	Constant ::= intConstant.	(,) , - , , % , > , >= , != , , .	
I176	Constant ::= doubleConstant.	(,) , - , , % , > , >= , != , , .	

I177	Constant ::= booleanConstant .	(,) , - , , % , > , >= , != , , .	
I178	Constant ::= stringConstant .	(,) , - , , % , > , >= , != , , .	
I179	Constant ::= null .	(,) , - , , % , > , >= , != , , .	
I180	Expr ::= LValue = Expr.	; , - , , % , > , >= , != , , .	
I181	Expr ::= (Expr).	; , - , , % , > , >= , != , , .	
I182	Expr ::= LValue = .Expr	- , , % , > , >= , != , , . ,)	IrA(I182, Expr) = I221
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139

I183	Expr ::= (Expr.)	- , , % , > , >= , != , , . ,)	IrA(I183,)) = I317
I184	Expr ::= Expr - .Expr	- , , % , > , >= , != , , . ,)	IrA(I184, Expr) = I222
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I185	Expr ::= Expr .Expr	- , , % , > , >= , != , , . ,)	IrA(I185, Expr) = I223
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130

	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I186	Expr ::= Expr % .Expr	- , , % , > , >= , != , , . ,)	IrA(I186, Expr) = I224
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134

I187	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
	Expr ::= Expr > .Expr	- , , % , > , >= , != , , . ,)	IrA(I187, Expr) = I225
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I188	Expr ::= Expr >= .Expr	- , , % , > , >= , != , , . ,)	IrA(I188, Expr) = I226
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127

Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139

I189

Expr ::= Expr != .Expr	- , , % , > , >= , != , , . ,)	IrA(I189, Expr) = I227
Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130

Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139

I190

Expr ::= Expr .Expr	- , , % , > , >= , != , , . ,)	IrA(I190, Expr) = I228
Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139

I191	LValue ::= Expr . ident	- , , % , > , >= , != , , . ,)	IrA(I191, ident) = I229
	Expr ::= - Expr.	- , , % , > , >= , != , , . ,)	
	Expr ::= ! Expr.	- , , % , > , >= , != , , . ,)	
I194	Expr ::= New (ident)	- , , % , > , >= , != , , . ,)	IrA(I194, ident) = I230
I195	Expr ::= New (ident.)	; , - , , % , > , >= , != , , .	IrA(I195,) = I231
	ConstDecl ::= static ConstType ident ; .	if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	
	ClassDecl ::= class ident (extends ident) ? (implements ident + ,) ? { Field* } .	\$	
I198	VariableDecl ::= Variable ; .	}	
I199	FunctionDecl ::= Type ident . (Formals) StmtBlock	}	IrA(I199, () = I232
I200	FunctionDecl ::= void ident . (Formals) StmtBlock	}	IrA(I200, () = I233
I201	ConstDecl ::= static ConstType ident ;	}	IrA(I201, ident) = I234
I202	LValue ::= Expr . ident.	; , - , , % , > , >= , != , , .	
I203	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* . }	}	IrA(I203, }) = I235
I204	IfStmt ::= if (Expr) . Stmt (else Stmt) ?	}	IrA(I203, Stmt) = I236
	Stmt ::= . Expr ? ;	(IrA(I64, Expr) = I321
	Stmt ::= . IfStmt	(IrA(I64, IfStmt) = I322

Stmt ::= .WhileStmt	(IrA(I64, WhileStmt) = I323
Stmt ::= .ForStmt	(IrA(I64, ForStmt) = I324
Stmt ::= .BreakStmt	(IrA(I64, BreakStmt) = I325
Stmt ::= .ReturnStmt	(IrA(I64, ReturnStmt) = I326
Stmt ::= .PrintStmt	(IrA(I64, PrintStmt) = I327
Stmt ::= .StmtBlock	(IrA(I64, StmtBlock) = I328
StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* }	(IrA(I64, {) = I329
IfStmt ::= .if (Expr) Stmt (else Stmt)?	(IrA(I64, if) = I330
WhileStmt ::= .while (Expr) Stmt	(IrA(I64, while) = I331
ForStmt ::= .for (Expr; Expr; Expr) Stmt	(IrA(I64, for) = I332
ReturnStmt ::= .return Expr;	(IrA(I64, return) = I333
BreakStmt ::= .break ;	(IrA(I64, break) = I334
PrintStmt ::= .System.out.println(Expr+ ,) ;	(IrA(I64, System.out.println) = I335
Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, () = I89
Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Constant ::= .intConstant	;; -, , % , > , >= , != , , .	IrA(I64, intConstant) = I94
Constant ::= .doubleConstant	;; -, , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
Constant ::= .booleanConstant	;; -, , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
Constant ::= .stringConstant	;; -, , % , > , >= , != , , .	IrA(I64, stringConstant) = I97

I205

Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
WhileStmt ::= while (Expr) .Stmt	}	IrA(I205, Stmt) = I237
Stmt ::= .Expr ? ;	}	IrA(I64, Expr) = I71
Stmt ::= .IfStmt	}	IrA(I64, IfStmt) = I72
Stmt ::= .WhileStmt	}	IrA(I64, WhileStmt) = I73
Stmt ::= .ForStmt	}	IrA(I64, ForStmt) = I74
Stmt ::= .BreakStmt	}	IrA(I64, BreakStmt) = I75
Stmt ::= .ReturnStmt	}	IrA(I64, ReturnStmt) = I76
Stmt ::= .PrintStmt	}	IrA(I64, PrintStmt) = I77
Stmt ::= .StmtBlock	}	IrA(I64, StmtBlock) = I78
StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* }	}	IrA(I64, { }) = I79
IfStmt ::= . if (Expr) Stmt (else Stmt)?	}	IrA(I64, if) = I80
WhileStmt ::= . while (Expr) Stmt	}	IrA(I64, while) = I81
ForStmt ::= . for (Expr; Expr; Expr) Stmt	}	IrA(I64, for) = I82
ReturnStmt ::= . return Expr;	}	IrA(I64, return) = I83
BreakStmt ::= . break ;	}	IrA(I64, break) = I84
PrintStmt ::= . System.out.println (Expr+ ,) ;	}	IrA(I64, System.out.println) = I85
Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= . this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, ()) = I89
Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= . New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= . ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93

I206	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
	ForStmt ::= for (Expr; .Expr; Expr) Stmt	}	IrA(I206, Expr) = I238
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, () = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I207	PrintStmt ::= System.out.println (Expr+ , .) ;	}	IrA(I207,)) = I239
I208	Expr ::= LValue = .Expr	(,) , - , , % , > , >= , != , , .	IrA(I208, Expr) = I240

Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
Constant ::= .booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179

I209

Expr ::= (Expr.)	(,) , - , , % , > , >= , != , , .	IrA(I209,)) = I241
------------------	--	----------------------

I210

Expr ::= Expr - .Expr	(,) , - , , % , > , >= , != , , .	IrA(I210, Expr) = I242
Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170

	Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
	LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
	LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
	Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
	Constant ::= .booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
	Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
	Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179
I211	Expr ::= Expr .Expr	(,) , - , , % , > , >= , != , , .	IrA(I211, Expr) = I243
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
	Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, () = I169
	Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
	LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
	LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175

	Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
	Constant ::= . booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
	Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
	Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179
I212	Expr ::= Expr % .Expr	(,) , - , , % , > , >= , != , , .	IrA(I212, Expr) = I244
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
	Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
	Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
	LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
	LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
	Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
	Constant ::= . booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
	Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
	Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179
I213	Expr ::= Expr > .Expr	(,) , - , , % , > , >= , != , , .	IrA(I213, Expr) = I245
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168

Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
Constant ::= .booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179

I214

Expr ::= Expr >= .Expr	(,) , - , , % , > , >= , != , , .	IrA(I214, Expr) = I246
Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172

Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
Constant ::= .booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179

I215

Expr ::= Expr != .Expr	(,) , - , , % , > , >= , != , , .	IrA(I215, Expr) = I247
Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
Constant ::= .booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179

I216	Expr ::= Expr .Expr	(,) , - , , % , > , >= , != , , .	IrA(I216, Expr) = I248
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
	Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, ()) = I169
	Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
	LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
	LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
	Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
	Constant ::= . booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
	Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
	Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179
I217	LValue ::= Expr . ident	(,) , - , , % , > , >= , != , , .	IrA(I217, ident) = I249
I218	Expr ::= - Expr.	(,) , - , , % , > , >= , != , , .	
I219	Expr ::= ! Expr.	(,) , - , , % , > , >= , != , , .	
I220	Expr ::= New (.ident)	(,) , - , , % , > , >= , != , , .	IrA(I220, ident) = I250

I221	Expr ::= LValue = Expr.	- , , % , > , >= , != , , . ,)	
I222	Expr ::= Expr - Expr.	- , , % , > , >= , != , , . ,)	
I223	Expr ::= Expr Expr.	- , , % , > , >= , != , , . ,)	
I224	Expr ::= Expr % Expr.	- , , % , > , >= , != , , . ,)	
I225	Expr ::= Expr > Expr.	- , , % , > , >= , != , , . ,)	
I226	Expr ::= Expr >= Expr.	- , , % , > , >= , != , , . ,)	
I227	Expr ::= Expr != Expr.	- , , % , > , >= , != , , . ,)	
I228	Expr ::= Expr Expr.	- , , % , > , >= , != , , . ,)	
I229	LValue ::= Expr .ident.	- , , % , > , >= , != , , . ,)	
I230	Expr ::= New (ident.)	- , , % , > , >= , != , , . ,)	
I231	Expr ::= New (ident).	; , - , , % , > , >= , != , , .	
I232	FunctionDecl ::= Type ident (.Formals) StmtBlock	}	IrA(I232, Formals) = I252
	Formals ::= .Variable, Formals)	IrA(I30, Variable) = I36
	Formals ::= .Variable)	IrA(I30, Variable) = I36
	Variable ::= .Type ident	, ,)	IrA(I30, Type) = I37
	Type ::= . int	ident, []	IrA(I30, int) = I14
	Type ::= . double	ident, []	IrA(30, double) = I15
	Type ::= . boolean	ident, []	IrA(I30, boolean) = I16
	Type ::= . string	ident, []	IrA(I30, string) = I17
	Type ::= . ident	ident, []	IrA(I30, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I30, Type) = I9
I233	FunctionDecl ::= void ident (.Formals) StmtBlock	}	IrA(I233, Formals) = I253

	Formals ::= .Variable, Formals)	IrA(I30, Variable) = I36
	Formals ::= .Variable)	IrA(I30, Variable) = I36
	Variable ::= .Type ident	, ,)	IrA(I30, Type) = I37
	Type ::= . int	ident, []	IrA(I30, int) = I14
	Type ::= . double	ident, []	IrA(30, double) = I15
	Type ::= . boolean	ident, []	IrA(I30, boolean) = I16
	Type ::= . string	ident, []	IrA(I30, string) = I17
	Type ::= . ident	ident, []	IrA(I30, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I30, Type) = I9
I234	ConstDecl ::= static ConstType ident ;	}	IrA(I234, ;) = I254
I235	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* }.	}	
I236	IfStmt ::= if (Expr) Stmt .(else Stmt)?	}	IrA(I236, () = I255
I237	WhileStmt ::= while (Expr) Stmt-	}	IrA(I237, ;) = I319
	Stmt ::= Expr ? .;	(
I238	ForStmt ::= for (Expr; Expr.; Expr) Stmt	}	IrA(I238, ;) = I256
	Stmt ::= IfStmt.	(
I239	PrintStmt ::= System.out.println (Expr+ ,) . ;	}	IrA(I239, ;) = I257
I240	Expr ::= LValue = Expr.	(,) , - , , % , > , >= , != , , .	
I241	Expr ::= (Expr).	(,) , - , , % , > , >= , != , , .	
I242	Expr ::= Expr - Expr.	(,) , - , , % , > , >= , != , , .	
I243	Expr ::= Expr Expr.	(,) , - , , % , > , >= , != , , .	
I244	Expr ::= Expr % Expr.	(,) , - , , % , > , >= , != , , .	

I245	Expr ::= Expr > Expr.	(,) , - , , % , > , >= , != , , .	
	Expr ::= Expr >= Expr.	(,) , - , , % , > , >= , != , , .	
	Expr ::= Expr != Expr.	(,) , - , , % , > , >= , != , , .	
I248	Expr ::= Expr Expr.	(,) , - , , % , > , >= , != , , .	
I249	LValue ::= Expr .ident.	(,) , - , , % , > , >= , != , , .	
I250	Expr ::= New (ident.)	(,) , - , , % , > , >= , != , , .	IrA(I250,)) = I258
I251	Expr ::= New (ident).	- , , % , > , >= , != , , . ,)	
I252	FunctionDecl ::= Type ident (Formals .) StmtBlock	}	IrA(I252,)) = I259
I253	FunctionDecl ::= void ident (Formals .) StmtBlock	}	IrA(I253,)) = I260
I254	ConstDecl ::= static ConstType ident ;.	}	
I255	IfStmt ::= if (Expr) Stmt (.else Stmt)?	}	IrA(I255, else) = I261
I256	ForStmt ::= for (Expr; Expr; .Expr) Stmt	}	IrA(I256, Expr) = I262
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130

	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I257	PrintStmt ::= System.out.println (Expr+ ,) ; .	}	
I258	Expr ::= New (ident).	(,) , - , , % , > , >= , != , , .	
I259	FunctionDecl ::= Type ident (Formals) .StmtBlock	}	IrA(I259, StmtBlock) = I263
	StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* }	}	IrA(I259, { }) = I79
I260	FunctionDecl ::= void ident (Formals) .StmtBlock	}	IrA(I260, StmtBlock) = I264
	StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* }	}	IrA(I260, { }) = I79
I261	IfStmt ::= if (Expr) Stmt (else .Stmt)?	}	IrA(I261, Stmt) = I265
	Stmt ::= .Expr ? ;)	IrA(I64, Expr) = I266
	Stmt ::= .IfStmt)	IrA(I64, IfStmt) = I267
	Stmt ::= .WhileStmt)	IrA(I64, WhileStmt) = I268
	Stmt ::= .ForStmt)	IrA(I64, ForStmt) = I269
	Stmt ::= .BreakStmt)	IrA(I64, BreakStmt) = I270
	Stmt ::= .ReturnStmt)	IrA(I64, ReturnStmt) = I271
	Stmt ::= .PrintStmt)	IrA(I64, PrintStmt) = I272
	Stmt ::= .StmtBlock)	IrA(I64, StmtBlock) = I273
	StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* })	IrA(I64, { }) = I274
	IfStmt ::= . if (Expr) Stmt (else Stmt)?)	IrA(I64, if) = I275

WhileStmt ::= while (Expr) Stmt)	IrA(I64, while) = I276
ForStmt ::= for (Expr; Expr; Expr) Stmt)	IrA(I64, for) = I277
ReturnStmt ::= return Expr;)	IrA(I64, return) = I278
BreakStmt ::= break ;)	IrA(I64, break) = I279
PrintStmt ::= .System.out.println (Expr+ ,) ;)	IrA(I64, System.out.println) = I280
Expr ::= .LValue = Expr	;; -, , % , >, >=, !=, , .	IrA(I64, LValue) = I86
Expr ::= .Constant	;; -, , % , >, >=, !=, , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	;; -, , % , >, >=, !=, , .	IrA(I64, LValue) = I86
Expr ::= .this	;; -, , % , >, >=, !=, , .	IrA(I64, this) = I88
Expr ::= .(Expr)	;; -, , % , >, >=, !=, , .	IrA(I64, () = I89
Expr ::= .Expr - Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	;; -, , % , >, >=, !=, , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	;; -, , % , >, >=, !=, , .	IrA(I64, !) = I91
Expr ::= .New (ident)	;; -, , % , >, >=, !=, , .	IrA(I64, New) = I92
LValue ::= .ident	;; -, , % , >, >=, !=, , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	;; -, , % , >, >=, !=, , .	IrA(I64, Expr) = I71
Constant ::= .intConstant	;; -, , % , >, >=, !=, , .	IrA(I64, intConstant) = I94
Constant ::= .doubleConstant	;; -, , % , >, >=, !=, , .	IrA(I64, doubleConstant) = I95
Constant ::= .booleanConstant	;; -, , % , >, >=, !=, , .	IrA(I64, booleanConstant) = I96
Constant ::= .stringConstant	;; -, , % , >, >=, !=, , .	IrA(I64, stringConstant) = I97
Constant ::= .null	;; -, , % , >, >=, !=, , .	IrA(I64, null}) = I98

I262

ForStmt ::= **for** (Expr; Expr; Expr.) Stmt

}

IrA(I262,)) = I281

I263

FunctionDecl ::= Type **ident** (Formals) StmtBlock.

}

I264

FunctionDecl ::= **void ident** (Formals) StmtBlock.

}

I265	IfStmt ::= if (Expr) Stmt (else Stmt.)?	}	IrA(I265,) = I282
I266	Stmt ::= Expr ? .;)	IrA(I266, ;) = I283
I267	Stmt ::= IfStmt.)	
I268	Stmt ::= WhileStmt.)	
I269	Stmt ::= ForStmt.)	
I270	Stmt ::= BreakStmt.)	
I271	Stmt ::= ReturnStmt.)	
I272	Stmt ::= PrintStmt.)	
I273	Stmt ::= StmtBlock.)	
I274	StmtBlock ::= { .VariableDecl* ConstDecl* Stmt* })	IrA(I274, VariableDecl) = I284
	VariableDecl ::= .Variable ;	static	IrA(I53, Variable) = I60
	Variable ::= .Type ident	;	IrA(I53, Type) = I9
	Type ::= .int	ident, []	IrA(I53, int) = I14
	Type ::= .double	ident, []	IrA(I53, double) = I15
	Type ::= .boolean	ident, []	IrA(I53, boolean) = I16
	Type ::= .string	ident, []	IrA(I53, string) = I17
	Type ::= .ident	ident, []	IrA(I53, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I53, Type) = I9
I275	IfStmt ::= if .(Expr) Stmt (else Stmt)?)	IrA(I275, () = I285
I276	WhileStmt ::= while .(Expr) Stmt)	IrA(I276, () = I286
I277	ForStmt ::= for .(Expr; Expr; Expr) Stmt)	IrA(I277, () = I287

1278	ReturnStmt ::= return .Expr;)	IrA(1278, Expr) = 1288
	Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(164, LValue) = 186
	Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(164, Constant) = 187
	Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(164, LValue) = 186
	Expr ::= . this	;; -, , % , > , >= , != , , .	IrA(164, this) = 188
	Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(164, () = 189
	Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(164, -) = 190
	Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(164, !) = 191
	Expr ::= . New (ident)	;; -, , % , > , >= , != , , .	IrA(164, New) = 192
	LValue ::= . ident	;; -, , % , > , >= , != , , .	IrA(164, ident) = 193
	LValue ::= .Expr . ident	;; -, , % , > , >= , != , , .	IrA(164, Expr) = 171
	Constant ::= . intConstant	;; -, , % , > , >= , != , , .	IrA(164, intConstant) = 194
	Constant ::= . doubleConstant	;; -, , % , > , >= , != , , .	IrA(164, doubleConstant) = 195
	Constant ::= . booleanConstant	;; -, , % , > , >= , != , , .	IrA(164, booleanConstant) = 196
	Constant ::= . stringConstant	;; -, , % , > , >= , != , , .	IrA(164, stringConstant) = 197
	Constant ::= . null	;; -, , % , > , >= , != , , .	IrA(164, null}) = 198
1279	BreakStmt ::= break . ;)	IrA(1279, ;) = 1289
1280	PrintStmt ::= System.out.println .(Expr+ ,) ;)	IrA(1280, () = 1290
1281	ForStmt ::= for (Expr; Expr; Expr) .Stmt	}	IrA(1281, Stmt) = 1291
	Stmt ::= .Expr ? ;	}	IrA(164, Expr) = 171
	Stmt ::= .IfStmt	}	IrA(164, IfStmt) = 172
	Stmt ::= .WhileStmt	}	IrA(164, WhileStmt) = 173
	Stmt ::= .ForStmt	}	IrA(164, ForStmt) = 174
	Stmt ::= .BreakStmt	}	IrA(164, BreakStmt) = 175

Stmt ::= .ReturnStmt	}	IrA(I64, ReturnStmt) = I76
Stmt ::= .PrintStmt	}	IrA(I64, PrintStmt) = I77
Stmt ::= .StmtBlock	}	IrA(I64, StmtBlock) = I78
StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* }	}	IrA(I64, {) = I79
IfStmt ::= .if (Expr) Stmt (else Stmt)?	}	IrA(I64, if) = I80
WhileStmt ::= .while (Expr) Stmt	}	IrA(I64, while) = I81
ForStmt ::= .for (Expr; Expr; Expr) Stmt	}	IrA(I64, for) = I82
ReturnStmt ::= .return Expr;	}	IrA(I64, return) = I83
BreakStmt ::= .break ;	}	IrA(I64, break) = I84
PrintStmt ::= .System.out.println(Expr+ ,) ;	}	IrA(I64, System.out.println) = I85
Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, () = I89
Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Constant ::= .intConstant	;; -, , % , > , >= , != , , .	IrA(I64, intConstant) = I94
Constant ::= .doubleConstant	;; -, , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
Constant ::= .booleanConstant	;; -, , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
Constant ::= .stringConstant	;; -, , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
Constant ::= .null	;; -, , % , > , >= , != , , .	IrA(I64, null}) = I98
IfStmt ::= if (Expr) Stmt (else Stmt)?.	}	

I283	Stmt ::= Expr ? ;.)	
I284	StmtBlock ::= { VariableDecl* .ConstDecl* Stmt* })	IrA(I284, ConstDecl) = I292
	ConstDecl ::= .static ConstType ident ;	if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	IrA(I59, static) = I65
I285	IfStmt ::= if (.Expr) Stmt (else Stmt)?)	IrA(I285, Expr) = I293
	Expr ::= .LValue = Expr	-, , %, >, >=, !=, , .,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	-, , %, >, >=, !=, , .,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	-, , %, >, >=, !=, , .,)	IrA(I89, LValue) = I126
	Expr ::= .this	-, , %, >, >=, !=, , .,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	-, , %, >, >=, !=, , .,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	-, , %, >, >=, !=, , .,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	-, , %, >, >=, !=, , .,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	-, , %, >, >=, !=, , .,)	IrA(I89, New) = I133
	LValue ::= .ident	-, , %, >, >=, !=, , .,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	-, , %, >, >=, !=, , .,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	-, , %, >, >=, !=, , .,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	-, , %, >, >=, !=, , .,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	-, , %, >, >=, !=, , .,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	-, , %, >, >=, !=, , .,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	-, , %, >, >=, !=, , .,)	IrA(I89, null) = I139

1286	WhileStmt ::= while (.Expr) Stmt)	IrA(I286, Expr) = I294
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= . this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= . New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= . ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr . ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= . intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= . doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= . stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= . null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
1287	ForStmt ::= for (.Expr; Expr; Expr) Stmt)	IrA(I287, Expr) = I295
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= . this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, () = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90

	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I288	ReturnStmt ::= return Expr .;)	IrA(I288, ;) = I296
I289	BreakStmt ::= break .;)	
I290	PrintStmt ::= System.out.println (.Expr+ ,) ;)	IrA(I290, ,) = I297
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
	Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, () = I169
	Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173

LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
Constant ::= . booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179

I291

ForStmt ::= for (Expr; Expr; Expr) Stmt.	}
---	---

I292

StmtBlock ::= { VariableDecl* ConstDecl* .Stmt* })	IrA(I292, Stmt) = I298
Stmt ::= .Expr ? ;	}	IrA(I64, Expr) = I71
Stmt ::= .IfStmt	}	IrA(I64, IfStmt) = I72
Stmt ::= .WhileStmt	}	IrA(I64, WhileStmt) = I73
Stmt ::= .ForStmt	}	IrA(I64, ForStmt) = I74
Stmt ::= .BreakStmt	}	IrA(I64, BreakStmt) = I75
Stmt ::= .ReturnStmt	}	IrA(I64, ReturnStmt) = I76
Stmt ::= .PrintStmt	}	IrA(I64, PrintStmt) = I77
Stmt ::= .StmtBlock	}	IrA(I64, StmtBlock) = I78
StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* }	}	IrA(I64, {) = I79
IfStmt ::= . if (Expr) Stmt (else Stmt)?	}	IrA(I64, if) = I80
WhileStmt ::= . while (Expr) Stmt	}	IrA(I64, while) = I81
ForStmt ::= .for (Expr; Expr; Expr) Stmt	}	IrA(I64, for) = I82
ReturnStmt ::= . return Expr;	}	IrA(I64, return) = I83
BreakStmt ::= . break ;	}	IrA(I64, break) = I84
PrintStmt ::= .System.out.println (Expr+ ,) ;	}	IrA(I64, System.out.println) = I85
Expr ::= .LValue = Expr	;; - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	;; - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	;; - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .this	;; - , , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	;; - , , % , > , >= , != , , .	IrA(I64, () = I89
Expr ::= .Expr - Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71

	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I293	IfStmt ::= if (Expr.) Stmt (else Stmt)?)	IrA(I293,)) = I299
I294	WhileStmt ::= while (Expr.) Stmt)	IrA(I294,)) = I300
I295	ForStmt ::= for (Expr .; Expr; Expr) Stmt)	IrA(I295, ;) = I301
I296	ReturnStmt ::= return Expr ; .)	
I297	PrintStmt ::= System.out.println (Expr+ .,) ;)	IrA(I297, ,) = I302
I298	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* .})	IrA(I298, }) = I303
I299	IfStmt ::= if (Expr) .Stmt (else Stmt)?)	IrA(I299, Stmt) = I304
	Stmt ::= .Expr ? ;	(IrA(I64, Expr) = I321
	Stmt ::= .IfStmt	(IrA(I64, IfStmt) = I322
	Stmt ::= .WhileStmt	(IrA(I64, WhileStmt) = I323
	Stmt ::= .ForStmt	(IrA(I64, ForStmt) = I324
	Stmt ::= .BreakStmt	(IrA(I64, BreakStmt) = I325
	Stmt ::= .ReturnStmt	(IrA(I64, ReturnStmt) = I326

	Stmt ::= .PrintStmt	(IrA(I64, PrintStmt) = I327
	Stmt ::= .StmtBlock	(IrA(I64, StmtBlock) = I328
	StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* }	(IrA(I64, { }) = I329
	IfStmt ::= . if (Expr) Stmt (else Stmt)?	(IrA(I64, if) = I330
	WhileStmt ::= . while (Expr) Stmt	(IrA(I64, while) = I331
	ForStmt ::= .for (Expr; Expr; Expr) Stmt	(IrA(I64, for) = I332
	ReturnStmt ::= . return Expr;	(IrA(I64, return) = I333
	BreakStmt ::= . break ;	(IrA(I64, break) = I334
	PrintStmt ::= .System.out.println(Expr+ ,) ;	(IrA(I64, System.out.println) = I335
	Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, () = I89
	Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	;; -, , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	;; -, , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= . booleanConstant	;; -, , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	;; -, , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	;; -, , % , > , >= , != , , .	IrA(I64, null}) = I98
I300			
	WhileStmt ::= while (Expr) .Stmt)	IrA(I300, Stmt) = I305
	Stmt ::= .Expr ? ;)	IrA(I64, Expr) = I266

Stmt ::= .IfStmt)	IrA(I64, IfStmt) = I267
Stmt ::= .WhileStmt)	IrA(I64, WhileStmt) = I268
Stmt ::= .ForStmt)	IrA(I64, ForStmt) = I269
Stmt ::= .BreakStmt)	IrA(I64, BreakStmt) = I270
Stmt ::= .ReturnStmt)	IrA(I64, ReturnStmt) = I271
Stmt ::= .PrintStmt)	IrA(I64, PrintStmt) = I272
Stmt ::= .StmtBlock)	IrA(I64, StmtBlock) = I273
StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* })	IrA(I64, { }) = I274
IfStmt ::= . if (Expr) Stmt (else Stmt)?)	IrA(I64, if) = I275
WhileStmt ::= . while (Expr) Stmt)	IrA(I64, while) = I276
ForStmt ::= .for (Expr; Expr; Expr) Stmt)	IrA(I64, for) = I277
ReturnStmt ::= . return Expr;)	IrA(I64, return) = I278
BreakStmt ::= . break ;)	IrA(I64, break) = I279
PrintStmt ::= .System.out.println(Expr+ ,) ;)	IrA(I64, System.out.println) = I280
Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, ()) = I89
Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Constant ::= .intConstant	;; -, , % , > , >= , != , , .	IrA(I64, intConstant) = I94
Constant ::= .doubleConstant	;; -, , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
Constant ::= . booleanConstant	;; -, , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96

	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I301	ForStmt ::= for (Expr; .Expr; Expr) Stmt)	IrA(I301, Expr) = I306
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, () = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I302	PrintStmt ::= System.out.println (Expr+ , .) ;)	IrA(I302,)) = I307
I303	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* }.)	
I304	IfStmt ::= if (Expr) Stmt .(else Stmt)?)	IrA(I304, () = I308
I305	WhileStmt ::= while (Expr) Stmt.)	

I306	ForStmt ::= for (Expr; Expr.; Expr) Stmt)	IrA(I306, ;) = I309
I307	PrintStmt ::= System.out.println (Expr+ ,) . ;)	IrA(I307, ;) = I310
I308	IfStmt ::= if (Expr) Stmt (.else Stmt)?)	IrA(I308, else) = I311
I309	ForStmt ::= for (Expr; Expr; .Expr) Stmt)	IrA(I309, Expr) = I312
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= .booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I310	PrintStmt ::= System.out.println (Expr+ ,) ; .)	
I311	IfStmt ::= if (Expr) Stmt (else .Stmt)?)	IrA(I311, Stmt) = I313

Stmt ::= .Expr ? ;)	IrA(I64, Expr) = I266
Stmt ::= .IfStmt)	IrA(I64, IfStmt) = I267
Stmt ::= .WhileStmt)	IrA(I64, WhileStmt) = I268
Stmt ::= .ForStmt)	IrA(I64, ForStmt) = I269
Stmt ::= .BreakStmt)	IrA(I64, BreakStmt) = I270
Stmt ::= .ReturnStmt)	IrA(I64, ReturnStmt) = I271
Stmt ::= .PrintStmt)	IrA(I64, PrintStmt) = I272
Stmt ::= .StmtBlock)	IrA(I64, StmtBlock) = I273
StmtBlock ::= .{ VariableDecl* ConstDecl* Stmt* })	IrA(I64, { }) = I274
IfStmt ::= .if (Expr) Stmt (else Stmt)?)	IrA(I64, if) = I275
WhileStmt ::= .while (Expr) Stmt)	IrA(I64, while) = I276
ForStmt ::= .for (Expr; Expr; Expr) Stmt)	IrA(I64, for) = I277
ReturnStmt ::= .return Expr;)	IrA(I64, return) = I278
BreakStmt ::= .break ;)	IrA(I64, break) = I279
PrintStmt ::= .System.out.println(Expr+ ,) ;)	IrA(I64, System.out.println) = I280
Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, ()) = I89
Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Constant ::= .intConstant	;; -, , % , > , >= , != , , .	IrA(I64, intConstant) = I94
Constant ::= .doubleConstant	;; -, , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95

	Constant ::= booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I312	ForStmt ::= for (Expr; Expr; Expr.) Stmt)	IrA(I312,)) = I314
I313	IfStmt ::= if (Expr) Stmt (else Stmt.)?)	IrA(I313,)) = I315
I314	ForStmt ::= for (Expr; Expr; Expr) .Stmt)	IrA(I314, Stmt) = I316
	Stmt ::= .Expr ? ;)	IrA(I64, Expr) = I266
	Stmt ::= .IfStmt)	IrA(I64, IfStmt) = I267
	Stmt ::= .WhileStmt)	IrA(I64, WhileStmt) = I268
	Stmt ::= .ForStmt)	IrA(I64, ForStmt) = I269
	Stmt ::= .BreakStmt)	IrA(I64, BreakStmt) = I270
	Stmt ::= .ReturnStmt)	IrA(I64, ReturnStmt) = I271
	Stmt ::= .PrintStmt)	IrA(I64, PrintStmt) = I272
	Stmt ::= .StmtBlock)	IrA(I64, StmtBlock) = I273
	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* })	IrA(I64, { }) = I274
	IfStmt ::= if (Expr) Stmt (else Stmt)?)	IrA(I64, if) = I275
	WhileStmt ::= while (Expr) Stmt)	IrA(I64, while) = I276
	ForStmt ::= .for (Expr; Expr; Expr) Stmt)	IrA(I64, for) = I277
	ReturnStmt ::= return Expr;)	IrA(I64, return) = I278
	BreakStmt ::= break ;)	IrA(I64, break) = I279
	PrintStmt ::= .System.out.println (Expr+ ,) ;)	IrA(I64, System.out.println) = I280
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, ()) = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71

	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= . booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I315	IfStmt ::= if (Expr) Stmt (else Stmt)?.)	
I316	ForStmt ::= for (Expr; Expr; Expr) Stmt.)	
I317	Expr ::= (Expr) .	- , , % , > , >= , != , , . ,)	
I318	Type ::= Type[].	ident, []	
I319	Stmt ::= Expr ? ; .	(
I320	Expr ::= New (ident).	- , , % , > , >= , != , , . ,)	
I321	Stmt ::= Expr ? .;	(IrA(I321, ;) = I336
I322	Stmt ::= IfStmt.	(
I323	Stmt ::= WhileStmt.	(
I324	Stmt ::= ForStmt.	(
I325	Stmt ::= BreakStmt.	(

I326	Stmt ::= ReturnStmt.	(
I327	Stmt ::= PrintStmt.	(
I328	Stmt ::= StmtBlock.	(
I329	StmtBlock ::= { .VariableDecl* ConstDecl* Stmt* }	(IrA(I321, VariableDecl) = I337
	VariableDecl ::= .Variable ;	static	IrA(I53, Variable) = I60
	Variable ::= .Type ident	;	IrA(I53, Type) = I9
	Type ::= .int	ident, []	IrA(I53, int) = I14
	Type ::= .double	ident, []	IrA(I53, double) = I15
	Type ::= .boolean	ident, []	IrA(I53, boolean) = I16
	Type ::= .string	ident, []	IrA(I53, string) = I17
	Type ::= .ident	ident, []	IrA(I53, ident) = I18
	Type ::= .Type[]	ident, []	IrA(I53, Type) = I9
I330	IfStmt ::= if .(Expr) Stmt (else Stmt)?	(
I331	WhileStmt ::= while .(Expr) Stmt	(IrA(I321, () = I338
I332	ForStmt ::= for .(Expr; Expr; Expr) Stmt	(IrA(I321, () = I339
I333	ReturnStmt ::= return .Expr;	(IrA(I321, Expr) = I340
	Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, () = I89
	Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71

	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= . booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I334	BreakStmt ::= break ;	(IrA(I64, ;) = I341
I335	PrintStmt ::= System.out.println .(Expr+ ,) ;	(IrA(I64, () = I342
I336	Stmt ::= Expr ? ;.	(
I337	StmtBlock ::= { VariableDecl* .ConstDecl* Stmt* }	(IrA(I64, constDecl) = I343
ConstDecl ::= .static ConstType ident ;		if, while, for, return, break, System.out.println, {, ident, intConstant, doubleConstant, booleanConstant, stringConstant, null, this, (, -, !, New,	IrA(I59, static) = I65
I338	WhileStmt ::= while (.Expr) Stmt	(IrA(I64, Expr) = I344
Expr ::= .LValue = Expr		- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
Expr ::= .Constant		- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
Expr ::= .Lvalue		- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
Expr ::= .this		- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
Expr ::= .(Expr)		- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
Expr ::= .Expr - Expr		- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr		- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130

Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137
Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139

I339

ForStmt ::= for (.Expr; Expr; Expr) Stmt	(IrA(I64, Expr) = I345
Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, () = I89
Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71

	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I340	ReturnStmt ::= return Expr.;	(IrA(I64, ;) = I346
I341	BreakStmt ::= break ;.	(
I342	PrintStmt ::= System.out.println (.Expr+ ,) ;	(IrA(I64, Expr) = I347
	Expr ::= .LValue = Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .Constant	(,) , - , , % , > , >= , != , , .	IrA(I91, Constant) = I167
	Expr ::= .Lvalue	(,) , - , , % , > , >= , != , , .	IrA(I91, LValue) = I166
	Expr ::= .this	(,) , - , , % , > , >= , != , , .	IrA(I91, this) = I168
	Expr ::= .(Expr)	(,) , - , , % , > , >= , != , , .	IrA(I91, () = I169
	Expr ::= .Expr - Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr % Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .- Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, -) = I171
	Expr ::= .Expr > Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr >= Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr != Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .Expr Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Expr ::= .! Expr	(,) , - , , % , > , >= , != , , .	IrA(I91, !) = I172
	Expr ::= .New (ident)	(,) , - , , % , > , >= , != , , .	IrA(I91, New) = I173
	LValue ::= .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, ident) = I174
	LValue ::= .Expr .ident	(,) , - , , % , > , >= , != , , .	IrA(I91, Expr) = I170
	Constant ::= .intConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, intConstant) = I175
	Constant ::= .doubleConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, doubleConstant) = I176
	Constant ::= .booleanConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, booleanConstant) = I177
	Constant ::= .stringConstant	(,) , - , , % , > , >= , != , , .	IrA(I91, stringConstant) = I178
	Constant ::= .null	(,) , - , , % , > , >= , != , , .	IrA(I91, null) = I179

StmtBlock ::= { VariableDecl* ConstDecl* .Stmt* }	(IrA(I64, Stmt) = I348
Stmt ::= .Expr ? ;	}	IrA(I64, Expr) = I71
Stmt ::= .IfStmt	}	IrA(I64, IfStmt) = I72
Stmt ::= .WhileStmt	}	IrA(I64, WhileStmt) = I73
Stmt ::= .ForStmt	}	IrA(I64, ForStmt) = I74
Stmt ::= .BreakStmt	}	IrA(I64, BreakStmt) = I75
Stmt ::= .ReturnStmt	}	IrA(I64, ReturnStmt) = I76
Stmt ::= .PrintStmt	}	IrA(I64, PrintStmt) = I77
Stmt ::= .StmtBlock	}	IrA(I64, StmtBlock) = I78
StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* }	}	IrA(I64, {) = I79
IfStmt ::= . if (Expr) Stmt (else Stmt)?	}	IrA(I64, if) = I80
WhileStmt ::= . while (Expr) Stmt	}	IrA(I64, while) = I81
ForStmt ::= . for (Expr; Expr; Expr) Stmt	}	IrA(I64, for) = I82
ReturnStmt ::= . return Expr;	}	IrA(I64, return) = I83
BreakStmt ::= . break ;	}	IrA(I64, break) = I84
PrintStmt ::= .System.out.println(Expr+ ,) ;	}	IrA(I64, System.out.println) = I85
Expr ::= .LValue = Expr	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	;; -, , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	;; -, , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .this	;; -, , % , > , >= , != , , .	IrA(I64, this) = I88
Expr ::= .(Expr)	;; -, , % , > , >= , != , , .	IrA(I64, () = I89
Expr ::= .Expr - Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr % Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .- Expr	;; -, , % , > , >= , != , , .	IrA(I64, -) = I90
Expr ::= .Expr > Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr >= Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr != Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	;; -, , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= .New (ident)	;; -, , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= .ident	;; -, , % , > , >= , != , , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	;; -, , % , > , >= , != , , .	IrA(I64, Expr) = I71
Constant ::= .intConstant	;; -, , % , > , >= , != , , .	IrA(I64, intConstant) = I94

	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= . booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I344	WhileStmt ::= while (Expr.) Stmt	(IrA(I64,)) = I359
I345	ForStmt ::= for (Expr.; Expr; Expr) Stmt	(IrA(I64, ;) = I349
I346	ReturnStmt ::= return Expr;.	(
I347	PrintStmt ::= System.out.println (Expr+ .,) ;	(IrA(I64, ,) = I350
I348	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* .}	(IrA(I64, }) = I351
I349	ForStmt ::= for (Expr; .Expr; Expr) Stmt	(IrA(I64, Expr) = I352
	Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, () = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94

	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= . booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I350	PrintStmt ::= System.out.println (Expr+ ,.) ;	(IrA(I64,)) = I353
I351	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* }.	(
I352	ForStmt ::= for (Expr; Expr.; Expr) Stmt	(IrA(I64, ;) = I354
I353	PrintStmt ::= System.out.println (Expr+ ,) .;	(IrA(I64, ;) = I355
I354	ForStmt ::= for (Expr; Expr; .Expr) Stmt	(IrA(I64, Expr) = I356
	Expr ::= .LValue = Expr	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .Constant	- , , % , > , >= , != , , . ,)	IrA(I89, Constant) = I127
	Expr ::= .Lvalue	- , , % , > , >= , != , , . ,)	IrA(I89, LValue) = I126
	Expr ::= .this	- , , % , > , >= , != , , . ,)	IrA(I89, this) = I128
	Expr ::= .(Expr)	- , , % , > , >= , != , , . ,)	IrA(I89, () = I129
	Expr ::= .Expr - Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr % Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .- Expr	- , , % , > , >= , != , , . ,)	IrA(I89, -) = I131
	Expr ::= .Expr > Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr >= Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr != Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .Expr Expr	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Expr ::= .! Expr	- , , % , > , >= , != , , . ,)	IrA(I89, !) = I132
	Expr ::= .New (ident)	- , , % , > , >= , != , , . ,)	IrA(I89, New) = I133
	LValue ::= .ident	- , , % , > , >= , != , , . ,)	IrA(I89, ident) = I134
	LValue ::= .Expr .ident	- , , % , > , >= , != , , . ,)	IrA(I89, Expr) = I130
	Constant ::= .intConstant	- , , % , > , >= , != , , . ,)	IrA(I89, intConstant) = I135
	Constant ::= .doubleConstant	- , , % , > , >= , != , , . ,)	IrA(I89, doubleConstant) = I136
	Constant ::= . booleanConstant	- , , % , > , >= , != , , . ,)	IrA(I89, booleanConstant) = I137

	Constant ::= .stringConstant	- , , % , > , >= , != , , . ,)	IrA(I89, stringConstant) = I138
	Constant ::= .null	- , , % , > , >= , != , , . ,)	IrA(I89, null) = I139
I355	PrintStmt ::= System.out.println (Expr+ ,) ;.	(
I356	ForStmt ::= for (Expr; Expr; Expr.) Stmt	(IrA(I64,)) = I357
I357	ForStmt ::= for (Expr; Expr; Expr) .Stmt	(IrA(I64, Stmt) = I358
	Stmt ::= .Expr ? ;	(IrA(I64, Expr) = I321
	Stmt ::= .IfStmt	(IrA(I64, IfStmt) = I322
	Stmt ::= .WhileStmt	(IrA(I64, WhileStmt) = I323
	Stmt ::= .ForStmt	(IrA(I64, ForStmt) = I324
	Stmt ::= .BreakStmt	(IrA(I64, BreakStmt) = I325
	Stmt ::= .ReturnStmt	(IrA(I64, ReturnStmt) = I326
	Stmt ::= .PrintStmt	(IrA(I64, PrintStmt) = I327
	Stmt ::= .StmtBlock	(IrA(I64, StmtBlock) = I328
	StmtBlock ::= { VariableDecl* ConstDecl* Stmt* }	(IrA(I64, { }) = I329
	IfStmt ::= if (Expr) Stmt (else Stmt)?	(IrA(I64, if) = I330
	WhileStmt ::= while (Expr) Stmt	(IrA(I64, while) = I331
	ForStmt ::= for (Expr; Expr; Expr) Stmt	(IrA(I64, for) = I332
	ReturnStmt ::= return Expr;	(IrA(I64, return) = I333
	BreakStmt ::= break ;	(IrA(I64, break) = I334
	PrintStmt ::= .System.out.println (Expr+ ,) ;	(IrA(I64, System.out.println) = I335
	Expr ::= .LValue = Expr	;; - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .Constant	;; - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
	Expr ::= .Lvalue	;; - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
	Expr ::= .this	;; - , , % , > , >= , != , , .	IrA(I64, this) = I88
	Expr ::= .(Expr)	;; - , , % , > , >= , != , , .	IrA(I64, ()) = I89
	Expr ::= .Expr - Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	;; - , , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	;; - , , % , > , >= , != , , .	IrA(I64, Expr) = I71

Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
Constant ::= . booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98

I358

ForStmt ::= for (Expr; Expr; Expr) Stmt.	(
---	---

I359

WhileStmt ::= while (Expr) .Stmt	(IrA(I64, Stmt) = I360
Stmt ::= .Expr ? ;	(IrA(I64, Expr) = I321
Stmt ::= .IfStmt	(IrA(I64, IfStmt) = I322
Stmt ::= .WhileStmt	(IrA(I64, WhileStmt) = I323
Stmt ::= .ForStmt	(IrA(I64, ForStmt) = I324
Stmt ::= .BreakStmt	(IrA(I64, BreakStmt) = I325
Stmt ::= .ReturnStmt	(IrA(I64, ReturnStmt) = I326
Stmt ::= .PrintStmt	(IrA(I64, PrintStmt) = I327
Stmt ::= .StmtBlock	(IrA(I64, StmtBlock) = I328
StmtBlock ::= . { VariableDecl* ConstDecl* Stmt* }	(IrA(I64, {) = I329
IfStmt ::= . if (Expr) Stmt (else Stmt)?	(IrA(I64, if) = I330
WhileStmt ::= . while (Expr) Stmt	(IrA(I64, while) = I331
ForStmt ::= . for (Expr; Expr; Expr) Stmt	(IrA(I64, for) = I332
ReturnStmt ::= . return Expr;	(IrA(I64, return) = I333
BreakStmt ::= . break ;	(IrA(I64, break) = I334
PrintStmt ::= . System.out.println (Expr+ ,) ;	(IrA(I64, System.out.println) = I335
Expr ::= .LValue = Expr	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= .Constant	; , - , , % , > , >= , != , , .	IrA(I64, Constant) = I87
Expr ::= .Lvalue	; , - , , % , > , >= , != , , .	IrA(I64, LValue) = I86
Expr ::= . this	; , - , , % , > , >= , != , , .	IrA(I64, this) = I88

	Expr ::= .(Expr)	; , - , , % , > , >= , != , , .	IrA(I64, () = I89
	Expr ::= .Expr - Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr % Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .- Expr	; , - , , % , > , >= , != , , .	IrA(I64, -) = I90
	Expr ::= .Expr > Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr >= Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr != Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .Expr Expr	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Expr ::= .! Expr	; , - , , % , > , >= , != , , .	IrA(I64, !) = I91
	Expr ::= .New (ident)	; , - , , % , > , >= , != , , .	IrA(I64, New) = I92
	LValue ::= .ident	; , - , , % , > , >= , != , , .	IrA(I64, ident) = I93
	LValue ::= .Expr .ident	; , - , , % , > , >= , != , , .	IrA(I64, Expr) = I71
	Constant ::= .intConstant	; , - , , % , > , >= , != , , .	IrA(I64, intConstant) = I94
	Constant ::= .doubleConstant	; , - , , % , > , >= , != , , .	IrA(I64, doubleConstant) = I95
	Constant ::= .booleanConstant	; , - , , % , > , >= , != , , .	IrA(I64, booleanConstant) = I96
	Constant ::= .stringConstant	; , - , , % , > , >= , != , , .	IrA(I64, stringConstant) = I97
	Constant ::= .null	; , - , , % , > , >= , != , , .	IrA(I64, null}) = I98
I360	WhileStmt ::= while (Expr) Stmt.	(