CS 365 — Programming Language Concepts

Classes and objects

Mar 24, 2008

OJay Concrete Syntax

Program → *Classes* void main () '{' *Declarations Statements* '}'

Classes → { *Class* }*

Class → class *Identifier* '{' { *Field* }* { *Method* }* '}'

Field → private *Type Identifiers* ;

Method → public *Type Identifier* ([*Parameters*])
'{' *Declarations Statements* '}'

Parameters → *Parameter* { , *Parameter* }

Parameter → *Type Identifier*

Declarations → { *Declaration* }*

Declaration → *Type Identifiers;*

Type → int | boolean | *Identifier*

Identifiers → *Identifer* { , *Identifier* }*

Statements → { *Statement* }*

Statement → ; | *Block* | *Assignment* | *IfStatement* | *WhileStatement* |
PrintStatement |*InvocationStatement* | *ReturnStatement*

Block → '{' *Declarations Statements* '}'

Assignment → *Identifier* = *Expression* ;

IfStatement → if (*Expression*) *Statement* { else *Statement* }$_{opt}$

OJay Concrete Syntax, continued

WhileStatement	$\longrightarrow$	while (*Experssion*) *Statement*
PrintStatement	$\longrightarrow$	System.out.println (*Expression*) ;
InvocationStatement	$\longrightarrow$	*Invocation* ;
Arguments	$\longrightarrow$	*Expression* { , *Expression* }*
ReturnStatement	$\longrightarrow$	return *Expression* ;
Expression	$\longrightarrow$	*Conjunction* { || *Conjunction* }*
Conjunction	$\longrightarrow$	*Relation* { && *Relation* }*
Relation	$\longrightarrow$	*Addition* { [| < | <= | > | >= | == | ! =] *Addition* }$_{opt}$
Addition	$\longrightarrow$	*Term* { [+ | -] *Term* }*
Term	$\longrightarrow$	*Negation* { ['*' | /] *Negation* }*
Negation	$\longrightarrow$	{ ! }$_{opt}$ *Factor*
Factor	$\longrightarrow$	*Identifier* | *Literal* | (*Expression*) | *Invocation* | *Instantiation*
Invocation	$\longrightarrow$	*Identifier* . *Identifier* (*Arguments*)
Instantiation	$\longrightarrow$	new *Identifier* ()

OJay Abstract Syntax

Program	$\longrightarrow$	*Class*$*$ *Block*
Class	$\longrightarrow$	*Identifier Declaration*$*$ *Method*$*$
Method	$\longrightarrow$	**Type Identifier** *Parameter*$*$ *Block*
Parameter	$\longrightarrow$	**Type Identifier**
Declaration	$\longrightarrow$	**Type Identifier**$*$
Statement	$\longrightarrow$	*Skip* | *Block* | *Assignment* | *Conditional* | *Loop* | *Print* | *InvkStmt* | *Return*
Skip	$\longrightarrow$	
Block	$\longrightarrow$	*Declaration*$*$ *Statement*$*$
Assignment	$\longrightarrow$	**Identifier** *Expression*
Conditional	$\longrightarrow$	*Expression Statement Statement*
Loop	$\longrightarrow$	*Expression Statement*
Print	$\longrightarrow$	*Expression*
InvkStmt	$\longrightarrow$	*Invk*
Return	$\longrightarrow$	*Expression*

OJay Abstract Syntax, continued

Expression	$\longrightarrow$	*Variable* | *IntLitExpr* | *BoolLitExpr* | *BinaryExpr* | *UnaryExpr* | *Invk* | *Instantiation*
Variable	$\longrightarrow$	**Identifier**
IntLitExpr	$\longrightarrow$	**IntLiteral**
BoolLitExpr	$\longrightarrow$	**BoolLiteral**
BinaryExpr	$\longrightarrow$	*Expression* **Operator** *Expression*
UnaryExpr	$\longrightarrow$	**Operator** *Expression*
Invk	$\longrightarrow$	**Identifier Identifier** *Expression*$*$
Instantiation	$\longrightarrow$	**Identifier**