CS 365 — Programming Language Concepts

Enhanced control structures in imperative languages

Jan 30, 2008

JayJay Concrete Syntax

$\textit{Program}$	$\longrightarrow$	$\text{void main () '\{'}\ \textit{Declarations Statements}\ \text{'\}'}$
$\textit{Declarations}$	$\longrightarrow$	$\{\ \textit{Declaration}\ \}*$
$\textit{Declaration}$	$\longrightarrow$	$\textit{Type Identifiers;}$
$\textit{Type}$	$\longrightarrow$	$\text{int} \mid \text{boolean}$
$\textit{Identifiers}$	$\longrightarrow$	$\textit{Identifer}\ \{\ ,\ \textit{Identifier}\ \}*$
$\textit{Statements}$	$\longrightarrow$	$\{\ \textit{Statement}\ \}*$
$\textit{Statement}$	$\longrightarrow$	$;\mid \textit{Block} \mid \textit{Assignment} \mid \textit{IfStatement} \mid \textit{WhileStatement} \mid \textit{PrintStatement} \mid \underline{\textit{ForStatement}} \mid \underline{\textit{DoWhileStatement}} \mid \underline{\textit{SwitchStatement}} \mid \underline{\textit{BreakStatement}} \mid \underline{\textit{ContinueStatement}}$
$\textit{Block}$	$\longrightarrow$	$\text{'\{'}\ \underline{\textit{Declarations}}\ \textit{Statements}\ \text{'\}'}$
$\textit{Assignment}$	$\longrightarrow$	$\textit{Identifier} = \textit{Expression}\ ;$
$\textit{IfStatement}$	$\longrightarrow$	$\text{if}\ (\ \textit{Expression}\ )\ \textit{Statement}\ \{\ \text{else}\ \textit{Statement}\ \}_{opt}$
$\textit{WhileStatement}$	$\longrightarrow$	$\text{while}\ (\ \textit{Experssion}\ )\ \textit{Statement}$
$\textit{PrintStatement}$	$\longrightarrow$	$\text{System.out.println}\ (\ \textit{Expression}\ )\ ;$
$\underline{\textit{ForStatement}}$	$\longrightarrow$	$\underline{\text{for}\ (\ \{\ \textit{Initializer}\ \}_{opt}\ ;\ \{\ \textit{Expression}\ \}_{opt};\ \{\ \textit{Increment}\ \}_{opt}\ )\ \textit{Statement}}$
$\underline{\textit{Initializer}}$	$\longrightarrow$	$\underline{\{\ \textit{Type}\ \}_{opt}\ \textit{Identifier} = \textit{Expression}}$
$\underline{\textit{Increment}}$	$\longrightarrow$	$\underline{\textit{Identifier} = \textit{Expression}}$
$\underline{\textit{DoWhileStatement}}$	$\longrightarrow$	$\underline{\text{do}\ \textit{Statement}\ \text{while}\ (\ \textit{Expression}\ )\ ;}$

JayJay Concrete Syntax, continued

SwitchStatement	→	switch (*Addition*) '{' *Cases* { *DefaultCase* }$_{opt}$ '}'
Cases	→	{ *Case* } *
Case	→	case *Literal* : *Statements*
DefaultCase	→	default: *Statements*
BreakStatement	→	break ;
ContinueStatement	→	continue ;
Expression	→	*Conjunction* { || *Conjunction* }*
Conjunction	→	*Relation* { && *Relation* }*
Relation	→	*Addition* { [| < | <= | > | >= | == | ! =] *Addition* }$_{opt}$
Addition	→	*Term* { [+ | -] *Term* }*
Term	→	*Negation* { ['*' | /] *Negation* }*
Negation	→	{ ! }$_{opt}$ *Factor*
Factor	→	*Identifier* | *Literal* | (*Expression*)

JayJay Abstract syntax

$$
\begin{array}{rcl}
\mathit{Program} & \longrightarrow & \mathit{Declaration}{*}\ \mathit{Statement}{*} \\
\mathit{Declaration} & \longrightarrow & \textbf{Type Identifier}{*} \\
\mathit{Statement} & \longrightarrow & \mathit{Skip} \mid \mathit{Block} \mid \mathit{Assignment} \mid \mathit{Conditional} \mid \mathit{Loop} \mid \mathit{Print} \mid \\
 & & \underline{\mathit{CountLoop}} \mid \underline{\mathit{PTLoop}} \mid \underline{\mathit{Switch}} \mid \underline{\mathit{Break}} \mid \underline{\mathit{Continue}} \\
\mathit{Skip} & \longrightarrow & \\
\mathit{Block} & \longrightarrow & \underline{\mathit{Declaration}{*}}\ \mathit{Statement}{*} \\
\mathit{Assignment} & \longrightarrow & \mathit{Identifier}\ \mathit{Expression} \\
\mathit{Conditional} & \longrightarrow & \mathit{Expression}\ \mathit{Statement}\ \mathit{Statement} \\
\mathit{Loop} & \longrightarrow & \mathit{Expression}\ \mathit{Statement} \\
\mathit{Print} & \longrightarrow & \mathit{Expression} \\
\underline{\mathit{CountLoop}} & \longrightarrow & \underline{\mathit{Initializer}\ \mathit{Expression}\ \mathit{Statement}\ \mathit{Statement}} \\
\underline{\mathit{Initializer}} & \longrightarrow & \underline{\mathit{DeclInitializer} \mid \mathit{NonDeclInitializer}} \\
\underline{\mathit{DeclInitializer}} & \longrightarrow & \underline{\textbf{Type Identifier}\ \mathit{Expression}} \\
\underline{\mathit{NonDeclInitializer}} & \longrightarrow & \underline{\textbf{Identifier}\ \mathit{Expression}} \\
\underline{\mathit{PTLoop}} & \longrightarrow & \underline{\mathit{Statement}\ \mathit{Expression}}
\end{array}
$$

JayJay Abstract Syntax, continued

$$
\begin{array}{rcl}
\underline{\mathit{Switch}} & \longrightarrow & \underline{\mathit{Expression}\ \mathit{StmtForSwitch}*} \\
\underline{\mathit{StmtForSwitch}} & \longrightarrow & \underline{\mathit{StmtRegular} \mid \mathit{StmtSpecial}} \\
\underline{\mathit{StmtRegular}} & \longrightarrow & \underline{\mathit{Statement}} \\
\underline{\mathit{StmtSpecial}} & \longrightarrow & \underline{\mathit{Case} \mid \mathit{Default}} \\
\underline{\mathit{Case}} & \longrightarrow & \underline{\textbf{\textsf{IntLiteral}}} \\
\underline{\mathit{Default}} & \longrightarrow & \\
\underline{\mathit{Break}} & \longrightarrow & \\
\underline{\mathit{Continue}} & \longrightarrow & \\
\mathit{Expression} & \longrightarrow & \mathit{Variable} \mid \mathit{IntLitExpr} \mid \mathit{BoolLitExpr} \mid \mathit{BinaryExpr} \mid \mathit{UnaryExpr} \\
\mathit{Variable} & \longrightarrow & \textbf{\textsf{Identifier}} \\
\mathit{IntLitExpr} & \longrightarrow & \textbf{\textsf{IntLiteral}} \\
\mathit{BoolLitExpr} & \longrightarrow & \textbf{\textsf{BoolLiteral}} \\
\mathit{BinaryExpr} & \longrightarrow & \mathit{Expression}\ \textbf{\textsf{Operator}}\ \mathit{Expression} \\
\mathit{UnaryExpr} & \longrightarrow & \textbf{\textsf{Operator}}\ \mathit{Expression}
\end{array}
$$

Switch Statements

$$\begin{array}{rcl}
\underline{\mathit{SwitchStatement}} & \longrightarrow & \underline{\textsf{switch}\ (\ \mathit{Addition})\ \text{'}\{\text{'}\ \mathit{Cases}\ \{\ \mathit{DefaultCase}\ \}_{opt}\ \text{'}\}\text{'}} \\
\underline{\mathit{Cases}} & \longrightarrow & \underline{\{\ \mathit{Case}\ \}\ *} \\
\underline{\mathit{Case}} & \longrightarrow & \underline{\textsf{case}\ \mathit{Literal}\ :\ \mathit{Statements}} \\
\underline{\mathit{DefaultCase}} & \longrightarrow & \underline{\textsf{default:}\ \mathit{Statements}}
\end{array}$$

$$\begin{array}{rcl}
\underline{\mathit{Switch}} & \longrightarrow & \underline{\mathit{Expression}\ \mathit{StmtForSwitch}*} \\
\underline{\mathit{StmtForSwitch}} & \longrightarrow & \underline{\mathit{StmtRegular}\mid\mathit{StmtSpecial}} \\
\underline{\mathit{StmtRegular}} & \longrightarrow & \underline{\mathit{Statement}} \\
\underline{\mathit{StmtSpecial}} & \longrightarrow & \underline{\mathit{Case}\mid\mathit{Default}} \\
\underline{\mathit{Case}} & \longrightarrow & \underline{\textsf{\textbf{IntLiteral}}} \\
\underline{\mathit{Default}} & \longrightarrow &
\end{array}$$