

## 1. HL.jj

Create a JavaCC file called HL.jj that JavaCC can use to generate a scanner for HL for the tokens specified below. This scanner will be marked by a program, so it is important that you follow the specifications below carefully.

- HL is case sensitive except for the keywords which are case-insensitive
- HL contains the following keywords: AND, AS, BOOL, BY, DEC, DO, ELIF, ELSE, FI, FOR, FOREACH, FROM, FUN, IF, IN, INT, ISA, NOT, NUF, OD, OR, POLYN, PRINT, RETURN, STOP, THEN, TO, VAR, WHILE, X. Each of these should be a separate token with the same name as the string.
- In the table underneath, each of the following additional tokens is defined by the string below it:

<b>Token</b>	LT	LE	GT	GE	NE	EQ	
<b>String</b>	<	<=	>	>=	<>	==	
<b>Token</b>	ADD	SUB	PROD	DIV	EXP		
<b>String</b>	+	-	*	/	^		
<b>Token</b>	ASSIGN	LPAREN	RPAREN	LSQUARE	RSQUARE	COMMA	SEMICOL
<b>String</b>	=	(	)	[	]	,	;

- IDENTIFIER tokens are strings of 1 or more letters and digits which must start with a letter.
- INTEGER tokens are strings of 1 or more digits.
- DECIMAL tokens are strings of 1 or more digits followed by a decimal point, then one or more digits, then optionally by the character E followed by a positive or negative integer.
- BOOLEAN tokens are one of the two strings: TRUE or FALSE (case insensitive)
- STRING tokens are strings of keyboard characters delimited by double-quotes ("). As in C and Java, "\" is the escape character. HL has 4 string escapes: \n for newline, \t for tab, \" for " and \\ for \
- HL allows **nested comments** which are delimited by "/\*" and "\*/" as well as inline comments in the C++ style: anything after a "/\*" up to the end of the line it occurs on is a comment.
- Your scanner should remove all comments and whitespace.

## 2. Token.java, BooleanToken.java, IntegerToken.java, DecimalToken.java, IdentifierToken.java and StringToken.java

- Following the instructions above the constructor for Token, extend the Token class for BOOLEAN, INTEGER, DECIMAL, IDENTIFIER and STRING tokens. Call them BooleanToken, IntegerToken, DecimalToken, IdentifierToken and StringToken respectively.
- These five Token classes contain an additional public variable called value. Value is of type Boolean in BooleanToken, math.BigInteger in IntegerToken, math.BigDecimal in DecimalToken, and String in IdentifierToken and StringToken. (i.e. value is an object wrapper for the value of the token).
- The value of a STRING token will be the string inside the surrounding quotes, with the escaped characters replaced by the real characters they represent.
- Overload the getValue() method of Token in the five subclasses. This returns the value of the token if there is one, or null if there isn't. Note that the only tokens which have a value are BOOLEAN, INTEGER, DECIMAL, STRING and IDENTIFIER.
- Override toString for IntegerToken and DecimalToken so that leading zeroes are not printed. Also override it in BooleanToken so that the string representation of the two Boolean tokens is always in lower case, and in StringToken so the escaped characters are replaced by the real characters they represent..