```
// Generated from C:\Users\u005Cuser\Documents\Compiler\Skeleton\src/decaf/
    DecafLexer.g4 by ANTLR 4.6
package decaf;
import org.antlr.v4.runtime.Lexer;
import org.antlr.v4.runtime.CharStream;
import org.antlr.v4.runtime.Token;
import org.antlr.v4.runtime.TokenStream;
import org.antlr.v4.runtime.*;
import org.antlr.v4.runtime.atn.*;
import org.antlr.v4.runtime.dfa.DFA;
import org.antlr.v4.runtime.misc.*;

@SuppressWarnings({"all", "warnings", "unchecked", "unused", "cast"})
public class DecafLexer extends Lexer {
        static { RuntimeMetaData.checkVersion("4.6", RuntimeMetaData.VERSION); }

        protected static final DFA[] _decisionToDFA;
        protected static final PredictionContextCache _sharedContextCache =
                new PredictionContextCache();
        public static final int
                BOOLEAN=1, BREAK=2, CALLOUT=3, CLASS=4, CONTINUE=5, ELSE=6,
                    FALSE=7, FOR=8,
                IF=9, INT=10, RETURN=11, TRUE=12, VOID=13, LCURLY=14, RCURLY=15,
                     LBRACE=16,
                RBRACE=17, LSQUARE=18, RSQUARE=19, COMMA=20, ASSIGN=21,
                    PLUSASSIGN=22,
                MINUSASSIGN=23, PLUS=24, MINUS=25, MULT=26, MOD=27, DIV=28, NOT
                    =29, AND=30,
                OR=31, EQ=32, NEQ=33, LT=34, GT=35, LTE=36, GTE=37, END=38, ID
                    =39, WS_=40,
                SL_COMMENT=41, CHAR=42, STRING=43, NUMBER=44;
        public static String[] modeNames = {
                "DEFAULT_MODE"
        };

        public static final String[] ruleNames = {
                "BOOLEAN", "BREAK", "CALLOUT", "CLASS", "CONTINUE", "ELSE", "
                    FALSE", "FOR",
                "IF", "INT", "RETURN", "TRUE", "VOID", "LCURLY", "RCURLY", "
                    LBRACE", "RBRACE",
                "LSQUARE", "RSQUARE", "COMMA", "ASSIGN", "PLUSASSIGN", "
                    MINUSASSIGN",
                "PLUS", "MINUS", "MULT", "MOD", "DIV", "NOT", "AND", "OR", "EQ",
                     "NEQ",
                "LT", "GT", "LTE", "GTE", "END", "ID", "WS_", "SL_COMMENT", "
                    CHAR", "STRING",
                "DIGIT", "HEX", "NUMBER", "ESC", "NOTESC"
        };

        private static final String[] _LITERAL_NAMES = {
                null, "'boolean'", "'break'", "'callout'", "'class'", "'continue
                    '", "'else'",
                "'false'", "'for'", "'if'", "'int'", "'return'", "'true'", "'
                    void'", "'{'",
                "'}'", "'('", "')'", "'['", "']'", "','", "'='", "'+='", "'-='",
                     "'+'",
                "'-'", "'*'", "'%'", "'/'", "'!'", "'&&'", "'||'", "'=='", "'!='
```

```
            ", "'<'",
        "'>'", "'<='", "'>='", "';'"
};
private static final String[] _SYMBOLIC_NAMES = {
        null, "BOOLEAN", "BREAK", "CALLOUT", "CLASS", "CONTINUE", "ELSE"
            , "FALSE",
        "FOR", "IF", "INT", "RETURN", "TRUE", "VOID", "LCURLY", "RCURLY"
            , "LBRACE",
        "RBRACE", "LSQUARE", "RSQUARE", "COMMA", "ASSIGN", "PLUSASSIGN",
             "MINUSASSIGN",
        "PLUS", "MINUS", "MULT", "MOD", "DIV", "NOT", "AND", "OR", "EQ",
             "NEQ",
        "LT", "GT", "LTE", "GTE", "END", "ID", "WS_", "SL_COMMENT", "
            CHAR", "STRING",
        "NUMBER"
};
public static final Vocabulary VOCABULARY = new VocabularyImpl(
    _LITERAL_NAMES, _SYMBOLIC_NAMES);

/**
 * @deprecated Use {@link #VOCABULARY} instead.
 */
@Deprecated
public static final String[] tokenNames;
static {
        tokenNames = new String[_SYMBOLIC_NAMES.length];
        for (int i = 0; i < tokenNames.length; i++) {
                tokenNames[i] = VOCABULARY.getLiteralName(i);
                if (tokenNames[i] == null) {
                        tokenNames[i] = VOCABULARY.getSymbolicName(i);
                }

                if (tokenNames[i] == null) {
                        tokenNames[i] = "<INVALID>";
                }
        }
}

@Override
@Deprecated
public String[] getTokenNames() {
        return tokenNames;
}

@Override

public Vocabulary getVocabulary() {
        return VOCABULARY;
}


public DecafLexer(CharStream input) {
        super(input);
        _interp = new LexerATNSimulator(this,_ATN,_decisionToDFA,
            _sharedContextCache);
}
```

```
@Override
public String getGrammarFileName() { return "DecafLexer.g4"; }

@Override
public String[] getRuleNames() { return ruleNames; }

@Override
public String getSerializedATN() { return _serializedATN; }

@Override
public String[] getModeNames() { return modeNames; }

@Override
public ATN getATN() { return _ATN; }

public static final String _serializedATN =
        "\3\u0430\ud6d1\u8206\uad2d\u4417\uaef1\u8d80\uaadd\2.\u012c\b
            \1\4\2\t"+
        "\2\4\3\t\3\4\4\t\4\4\5\t\5\4\6\t\6\4\7\t\7\4\b\t\b\4\t\t\t\4\n\
            t\n\4\13"+
        "\t\13\4\f\t\f\4\r\t\r\4\16\t\16\4\17\t\17\4\20\t\20\4\21\t
            \21\4\22\t\22"+
        "\4\23\t\23\4\24\t\24\4\25\t\25\4\26\t\26\4\27\t\27\4\30\t
            \30\4\31\t\31"+
        "\4\32\t\32\4\33\t\33\4\34\t\34\4\35\t\35\4\36\t\36\4\37\t\37\4
            \t \4!"+
        "\t!\4\"\t\"\4#\t#\4$\t$\4%\t%\4&\t&\4\'\t\'\4(\t(\4)\t)\4*\t
            *\4+\t+\4"+
        ",\t,\4-\t-\4.\t.\4/\t/\4\60\t\60\4\61\t
            \61\3\2\3\2\3\2\3\2\3\2\3\2\3\2"+
        "
            \3\2\3\3\3\3\3\3\3\3\3\3\3\3\3\3\4\3\4\3\4\3\4\3\4\3\4\3\4\3\4\3\4\3\5\3\5\3
            "+
        "
            \5\3\5\3\5\3\5\3\6\3\6\3\6\3\6\3\6\3\6\3\6\3\6\3\6\3\6\3\7\3\7\3\7\3\7\3\7\3\7
            "+
        "\3\b\3\b\3\b\3\b\3\b\3\b\3\t\3\t\3\t\3\t\3\n\3\n\3\n
            \3\13\3\13\3\13\3"+
        "\13\3\f\3\f\3\f\3\f\3\f\3\f\3\f\3\r\3\r\3\r\3\r\3\r
            \3\16\3\16\3\16\3\16"+
        "
            \3\16\3\17\3\17\3\20\3\20\3\21\3\21\3\22\3\22\3\23\3\23\3\24\3\24\3\25
            "+
        "
            \3\25\3\26\3\26\3\27\3\27\3\27\3\30\3\30\3\30\3\31\3\31\3\32\3\32\3\33
            "+
        "\3\33\3\34\3\34\3\35\3\35\3\36\3\36\3\37\3\37\3\37\3 \3 \3
            \3!\3!\3!\3"+
        "\"\3\"\3\"\3#\3#\3$\3$\3%\3%\3&\3&\3&\3\'\3\'\3(\3(\3(\7(\u00ec
            \n(\f("+
        "\16(\u00ef\13(\3)\3)\3)\3)\3*\3*\3*\3*\7*\u00f9\n*\f*\16*\u00fc
            \13*\3"+
        "*\3*\3*\3*\3+\3+\3+\5+\u0105\n+\3+\3+\3,\3,\3,\7,\u010c\n,\f
            ,\16,\u010f"+
        "\13,\3,\3,\3-\3-\3.\3.\5.\u0117\n.\3/\6/\u011a\n/\r/\16/\u011b
            \3/\3/\3"+
        "/\3/\6/\u0122\n/\r/\16/\u0123\5/\u0126\n
```

```
    /\3\60\3\60\3\60\3\61\3\61\2\2"+
"\62\3\3\5\4\7\5\t\6\13\7\r\b\17\t\21\n\23\13\25\f\27\r
    \31\16\33\17\35"+
"\20\37\21!\22#\23%\24\'\25)
    \26+\27-\30/\31\61\32\63\33\65\34\67\359\36"+
";\37= ?!A\"C#E$G%I&K\'M(O)Q*S+U,W-Y\2[\2]._\2a\2\3\2\n\5\2C\\
    aac|\6\2"+
"\62;C\\aac|\4\2\13\f\"\"\3\2\f\f\3\2\62;\4\2CHch\7\2$$))^^ppvv
    \6\2\13"+
"\f$$))^^\u0130\2\3\3\2\2\2\2\5\3\2\2\2\2\7\3\2\2\2\2\t
    \3\2\2\2\2\13\3"+
"\2\2\2\2\r
    \3\2\2\2\2\17\3\2\2\2\2\21\3\2\2\2\2\23\3\2\2\2\2\25\3\2\2\2"
    +
"
    \2\27\3\2\2\2\2\31\3\2\2\2\2\33\3\2\2\2\2\35\3\2\2\2\2\37\3\2\2\2\2!\3
    "+
"\2\2\2\2#\3\2\2\2\2%\3\2\2\2\2\'\3\2\2\2\2)
    \3\2\2\2\2+\3\2\2\2\2-\3\2"+
"
    \2\2\2/\3\2\2\2\2\61\3\2\2\2\2\63\3\2\2\2\2\65\3\2\2\2\2\67\3\2\2\2\2
    "+
"9\3\2\2\2\2;\3\2\2\2\2=\3\2\2\2\2?\3\2\2\2\2A\3\2\2\2\2C
    \3\2\2\2\2E\3"+
"\2\2\2\2G\3\2\2\2\2I\3\2\2\2\2K\3\2\2\2\2M\3\2\2\2\2O\3\2\2\2\2
    Q\3\2\2"+
"\2\2S\3\2\2\2\2U\3\2\2\2\2W\3\2\2\2\2]\3\2\2\2\3c\3\2\2\2\5k
    \3\2\2\2\7"+
"q\3\2\2\2\ty\3\2\2\2\13\177\3\2\2\2\r\u0088\3\2\2\2\17\u008d
    \3\2\2\2\21"+
"\u0093\3\2\2\2\23\u0097\3\2\2\2\25\u009a\3\2\2\2\27\u009e
    \3\2\2\2\31\u00a5"+
"\3\2\2\2\33\u00aa\3\2\2\2\35\u00af\3\2\2\2\37\u00b1\3\2\2\2!\
    u00b3\3\2"+
"\2\2#\u00b5\3\2\2\2%\u00b7\3\2\2\2\'\u00b9\3\2\2\2)\u00bb
    \3\2\2\2+\u00bd"+
"\3\2\2\2-\u00bf\3\2\2\2/\u00c2\3\2\2\2\61\u00c5\3\2\2\2\63\
    u00c7\3\2\2"+
"\2\65\u00c9\3\2\2\2\67\u00cb\3\2\2\29\u00cd\3\2\2\2;\u00cf
    \3\2\2\2=\u00d1"+
"\3\2\2\2?\u00d4\3\2\2\2A\u00d7\3\2\2\2C\u00da\3\2\2\2E\u00dd
    \3\2\2\2G"+
"\u00df\3\2\2\2I\u00e1\3\2\2\2K\u00e4\3\2\2\2M\u00e7\3\2\2\2O\
    u00e9\3\2"+
"\2\2Q\u00f0\3\2\2\2S\u00f4\3\2\2\2U\u0101\3\2\2\2W\u0108
    \3\2\2\2Y\u0112"+
"\3\2\2\2[\u0116\3\2\2\2]\u0125\3\2\2\2_\u0127\3\2\2\2a\u012a
    \3\2\2\2c"+
"d\7d\2\2de\7q\2\2ef\7q\2\2fg\7n\2\2gh\7g\2\2hi\7c\2\2ij\7p\2\2j
    \4\3\2"+
"\2\2kl\7d\2\2lm\7t\2\2mn\7g\2\2no\7c\2\2op\7m\2\2p\6\3\2\2\2qr
    \7e\2\2"+
"rs\7c\2\2st\7n\2\2tu\7n\2\2uv\7q\2\2vw\7w\2\2wx\7v\2\2x\b
    \3\2\2\2yz\7"+
"e\2\2z{\7n\2\2{|\7c\2\2|}\7u\2\2}~\7u\2\2~\n\3\2\2\2\177\u0080
    \7e\2\2"+
"\u0080\u0081\7q\2\2\u0081\u0082\7p\2\2\u0082\u0083\7v\2\2\u0083
```

```
    \u0084"+
"\7k\2\2\u0084\u0085\7p\2\2\u0085\u0086\7w\2\2\u0086\u0087\7g
    \2\2\u0087"+
"\f\3\2\2\2\u0088\u0089\7g\2\2\u0089\u008a\7n\2\2\u008a\u008b\7u
    \2\2\u008b"+
"\u008c\7g\2\2\u008c\16\3\2\2\2\u008d\u008e\7h\2\2\u008e\u008f\7
    c\2\2\u008f"+
"\u0090\7n\2\2\u0090\u0091\7u\2\2\u0091\u0092\7g\2\2\u0092
    \20\3\2\2\2\u0093"+
"\u0094\7h\2\2\u0094\u0095\7q\2\2\u0095\u0096\7t\2\2\u0096
    \22\3\2\2\2\u0097"+
"\u0098\7k\2\2\u0098\u0099\7h\2\2\u0099\24\3\2\2\2\u009a\u009b\7
    k\2\2\u009b"+
"\u009c\7p\2\2\u009c\u009d\7v\2\2\u009d\26\3\2\2\2\u009e\u009f\7
    t\2\2\u009f"+
"\u00a0\7g\2\2\u00a0\u00a1\7v\2\2\u00a1\u00a2\7w\2\2\u00a2\u00a3
    \7t\2\2"+
"\u00a3\u00a4\7p\2\2\u00a4\30\3\2\2\2\u00a5\u00a6\7v\2\2\u00a6\
    u00a7\7"+
"t\2\2\u00a7\u00a8\7w\2\2\u00a8\u00a9\7g\2\2\u00a9\32\3\2\2\2\
    u00aa\u00ab"+
"\7x\2\2\u00ab\u00ac\7q\2\2\u00ac\u00ad\7k\2\2\u00ad\u00ae\7f
    \2\2\u00ae"+
"\34\3\2\2\2\u00af\u00b0\7}\2\2\u00b0\36\3\2\2\2\u00b1\u00b2
    \7\177\2\2"+
"\u00b2 \3\2\2\2\u00b3\u00b4\7*\2\2\u00b4\"\3\2\2\2\u00b5\u00b6
    \7+\2\2"+
"\u00b6$\3\2\2\2\u00b7\u00b8\7]\2\2\u00b8&\3\2\2\2\u00b9\u00ba\7
    _\2\2\u00ba"+
"(\3\2\2\2\u00bb\u00bc\7.\2\2\u00bc*\3\2\2\2\u00bd\u00be\7?\2\2\
    u00be,"+
"\3\2\2\2\u00bf\u00c0\7-\2\2\u00c0\u00c1\7?\2\2\u00c1.\3\2\2\2\
    u00c2\u00c3"+
"\7/\2\2\u00c3\u00c4\7?\2\2\u00c4\60\3\2\2\2\u00c5\u00c6\7-\2\2\
    u00c6\62"+
"\3\2\2\2\u00c7\u00c8\7/\2\2\u00c8\64\3\2\2\2\u00c9\u00ca
    \7,\2\2\u00ca"+
"\66\3\2\2\2\u00cb\u00cc\7\'\2\2\u00cc8\3\2\2\2\u00cd\u00ce
    \7\61\2\2\u00ce"+
":\3\2\2\2\u00cf\u00d0\7#\2\2\u00d0<\3\2\2\2\u00d1\u00d2\7(\2\2\
    u00d2\u00d3"+
"\7(\2\2\u00d3>\3\2\2\2\u00d4\u00d5\7~\2\2\u00d5\u00d6\7~\2\2\
    u00d6@\3"+
"\2\2\2\u00d7\u00d8\7?\2\2\u00d8\u00d9\7?\2\2\u00d9B\3\2\2\2\
    u00da\u00db"+
"\7#\2\2\u00db\u00dc\7?\2\2\u00dcD\3\2\2\2\u00dd\u00de\7>\2\2\
    u00deF\3"+
"\2\2\2\u00df\u00e0\7@\2\2\u00e0H\3\2\2\2\u00e1\u00e2\7>\2\2\
    u00e2\u00e3"+
"\7?\2\2\u00e3J\3\2\2\2\u00e4\u00e5\7@\2\2\u00e5\u00e6\7?\2\2\
    u00e6L\3"+
"\2\2\2\u00e7\u00e8\7=\2\2\u00e8N\3\2\2\2\u00e9\u00ed\t\2\2\2\
    u00ea\u00ec"+
"\t\3\2\2\u00eb\u00ea\3\2\2\2\u00ec\u00ef\3\2\2\2\u00ed\u00eb
    \3\2\2\2\u00ed"+
"\u00ee\3\2\2\2\u00eeP\3\2\2\2\u00ef\u00ed\3\2\2\2\u00f0\u00f1\t
    \4\2\2"+
```

```
            "\u00f1\u00f2\3\2\2\2\u00f2\u00f3\b)\2\2\u00f3R\3\2\2\2\u00f4\
                u00f5\7\61"+
            "\2\2\u00f5\u00f6\7\61\2\2\u00f6\u00fa\3\2\2\2\u00f7\u00f9\n
                \5\2\2\u00f8"+
            "\u00f7\3\2\2\2\u00f9\u00fc\3\2\2\2\u00fa\u00f8\3\2\2\2\u00fa\
                u00fb\3\2"+
            "\2\2\u00fb\u00fd\3\2\2\2\u00fc\u00fa\3\2\2\2\u00fd\u00fe\7\f
                \2\2\u00fe"+
            "\u00ff\3\2\2\2\u00ff\u0100\b*\2\2\u0100T\3\2\2\2\u0101\u0104\7)
                \2\2\u0102"+
            "\u0105\5_\60\2\u0103\u0105\5a\61\2\u0104\u0102\3\2\2\2\u0104\
                u0103\3\2"+
            "\2\2\u0105\u0106\3\2\2\2\u0106\u0107\7)\2\2\u0107V\3\2\2\2\
                u0108\u010d"+
            "\7$\2\2\u0109\u010c\5_\60\2\u010a\u010c\5a\61\2\u010b\u0109
                \3\2\2\2\u010b"+
            "\u010a\3\2\2\2\u010c\u010f\3\2\2\2\u010d\u010b\3\2\2\2\u010d\
                u010e\3\2"+
            "\2\2\u010e\u0110\3\2\2\2\u010f\u010d\3\2\2\2\u0110\u0111\7$
                \2\2\u0111"+
            "X\3\2\2\2\u0112\u0113\t\6\2\2\u0113Z\3\2\2\2\u0114\u0117\t
                \7\2\2\u0115"+
            "\u0117\5Y-\2\u0116\u0114\3\2\2\2\u0116\u0115\3\2\2\2\u0117
                \\\3\2\2\2\u0118"+
            "\u011a\5Y-\2\u0119\u0118\3\2\2\2\u011a\u011b\3\2\2\2\u011b\
                u0119\3\2\2"+
            "\2\u011b\u011c\3\2\2\2\u011c\u0126\3\2\2\2\u011d\u011e
                \7\62\2\2\u011e"+
            "\u011f\7z\2\2\u011f\u0121\3\2\2\2\u0120\u0122\5[.\2\u0121\u0120
                \3\2\2"+
            "\2\u0122\u0123\3\2\2\2\u0123\u0121\3\2\2\2\u0123\u0124\3\2\2\2\
                u0124\u0126"+
            "\3\2\2\2\u0125\u0119\3\2\2\2\u0125\u011d\3\2\2\2\u0126
                ^\3\2\2\2\u0127"+
            "\u0128\7^\2\2\u0128\u0129\t\b\2\2\u0129`\3\2\2\2\u012a\u012b\n\
                t\2\2\u012b"+
            "b\3\2\2\2\f\2\u00ed\u00fa\u0104\u010b\u010d\u0116\u011b\u0123\
                u0125\3"+
            "\b\2\2";
        public static final ATN _ATN =
                new ATNDeserializer().deserialize(_serializedATN.toCharArray());
        static {
                _decisionToDFA = new DFA[_ATN.getNumberOfDecisions()];
                for (int i = 0; i < _ATN.getNumberOfDecisions(); i++) {
                        _decisionToDFA[i] = new DFA(_ATN.getDecisionState(i), i)
                            ;
                }
        }
}
```