

```
// Generated from C:\Users\u005Cuser\Documents\Compiler\Skeleton\src\decaf\
DecafParser.g4 by ANTLR 4.6
package decaf;
import org.antlr.v4.runtime.tree.ParseTreeListener;

/**
 * This interface defines a complete listener for a parse tree produced by
 * {@link DecafParser}.
 */
public interface DecafParserListener extends ParseTreeListener {
    /**
     * Enter a parse tree produced by {@link DecafParser#program}.
     * @param ctx the parse tree
     */
    void enterProgram(DecafParser.ProgramContext ctx);
    /**
     * Exit a parse tree produced by {@link DecafParser#program}.
     * @param ctx the parse tree
     */
    void exitProgram(DecafParser.ProgramContext ctx);
    /**
     * Enter a parse tree produced by {@link DecafParser#field_name}.
     * @param ctx the parse tree
     */
    void enterField_name(DecafParser.Field_nameContext ctx);
    /**
     * Exit a parse tree produced by {@link DecafParser#field_name}.
     * @param ctx the parse tree
     */
    void exitField_name(DecafParser.Field_nameContext ctx);
    /**
     * Enter a parse tree produced by {@link DecafParser#field_decl}.
     * @param ctx the parse tree
     */
    void enterField_decl(DecafParser.Field_declContext ctx);
    /**
     * Exit a parse tree produced by {@link DecafParser#field_decl}.
     * @param ctx the parse tree
     */
    void exitField_decl(DecafParser.Field_declContext ctx);
    /**
     * Enter a parse tree produced by {@link DecafParser#method_decl}.
     * @param ctx the parse tree
     */
    void enterMethod_decl(DecafParser.Method_declContext ctx);
    /**
     * Exit a parse tree produced by {@link DecafParser#method_decl}.
     * @param ctx the parse tree
     */
    void exitMethod_decl(DecafParser.Method_declContext ctx);
    /**
     * Enter a parse tree produced by {@link DecafParser#meth_name}.
     * @param ctx the parse tree
     */
    void enterMeth_name(DecafParser.Meth_nameContext ctx);
    /**
     * Exit a parse tree produced by {@link DecafParser#meth_name}.
     */
}
```

```
* @param ctx the parse tree
*/
void exitMeth_name(DecafParser.Meth_nameContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#meth_type}.
* @param ctx the parse tree
*/
void enterMeth_type(DecafParser.Meth_typeContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#meth_type}.
* @param ctx the parse tree
*/
void exitMeth_type(DecafParser.Meth_typeContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#arg_type}.
* @param ctx the parse tree
*/
void enterArg_type(DecafParser.Arg_typeContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#arg_type}.
* @param ctx the parse tree
*/
void exitArg_type(DecafParser.Arg_typeContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#block}.
* @param ctx the parse tree
*/
void enterBlock(DecafParser.BlockContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#block}.
* @param ctx the parse tree
*/
void exitBlock(DecafParser.BlockContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#var_decl}.
* @param ctx the parse tree
*/
void enterVar_decl(DecafParser.Var_declContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#var_decl}.
* @param ctx the parse tree
*/
void exitVar_decl(DecafParser.Var_declContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#var_name}.
* @param ctx the parse tree
*/
void enterVar_name(DecafParser.Var_nameContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#var_name}.
* @param ctx the parse tree
*/
void exitVar_name(DecafParser.Var_nameContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#type}.
* @param ctx the parse tree
*/
```

```
void enterType(DecafParser.TypeContext ctx);
/**
 * Exit a parse tree produced by {@link DecafParser#type}.
 * @param ctx the parse tree
 */
void exitType(DecafParser.TypeContext ctx);
/**
 * Enter a parse tree produced by the {@code Assign}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void enterAssign(DecafParser.AssignContext ctx);
/**
 * Exit a parse tree produced by the {@code Assign}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void exitAssign(DecafParser.AssignContext ctx);
/**
 * Enter a parse tree produced by the {@code MC}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void enterMC(DecafParser.MCContext ctx);
/**
 * Exit a parse tree produced by the {@code MC}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void exitMC(DecafParser.MCContext ctx);
/**
 * Enter a parse tree produced by the {@code If}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void enterIf(DecafParser.IfContext ctx);
/**
 * Exit a parse tree produced by the {@code If}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void exitIf(DecafParser.IfContext ctx);
/**
 * Enter a parse tree produced by the {@code For}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void enterFor(DecafParser.ForContext ctx);
/**
 * Exit a parse tree produced by the {@code For}
 * labeled alternative in {@link DecafParser#statement}.
 * @param ctx the parse tree
 */
void exitFor(DecafParser.ForContext ctx);
/**
 * Enter a parse tree produced by the {@code Return}
 * labeled alternative in {@link DecafParser#statement}.
```

```
* @param ctx the parse tree
*/
void enterReturn(DecafParser.ReturnContext ctx);
/**
* Exit a parse tree produced by the {@code Return}
* labeled alternative in {@link DecafParser#statement}.
* @param ctx the parse tree
*/
void exitReturn(DecafParser.ReturnContext ctx);
/**
* Enter a parse tree produced by the {@code Break}
* labeled alternative in {@link DecafParser#statement}.
* @param ctx the parse tree
*/
void enterBreak(DecafParser.BreakContext ctx);
/**
* Exit a parse tree produced by the {@code Break}
* labeled alternative in {@link DecafParser#statement}.
* @param ctx the parse tree
*/
void exitBreak(DecafParser.BreakContext ctx);
/**
* Enter a parse tree produced by the {@code Continue}
* labeled alternative in {@link DecafParser#statement}.
* @param ctx the parse tree
*/
void enterContinue(DecafParser.ContinueContext ctx);
/**
* Exit a parse tree produced by the {@code Continue}
* labeled alternative in {@link DecafParser#statement}.
* @param ctx the parse tree
*/
void exitContinue(DecafParser.ContinueContext ctx);
/**
* Enter a parse tree produced by the {@code Bl}
* labeled alternative in {@link DecafParser#statement}.
* @param ctx the parse tree
*/
void enterBl(DecafParser.BlContext ctx);
/**
* Exit a parse tree produced by the {@code Bl}
* labeled alternative in {@link DecafParser#statement}.
* @param ctx the parse tree
*/
void exitBl(DecafParser.BlContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#assign_op}.
* @param ctx the parse tree
*/
void enterAssign_op(DecafParser.Assign_opContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#assign_op}.
* @param ctx the parse tree
*/
void exitAssign_op(DecafParser.Assign_opContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#math_assign}.
```

```
* @param ctx the parse tree
*/
void enterMath_assign(DecafParser.Math_assignContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#math_assign}.
* @param ctx the parse tree
*/
void exitMath_assign(DecafParser.Math_assignContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#method_call}.
* @param ctx the parse tree
*/
void enterMethod_call(DecafParser.Method_callContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#method_call}.
* @param ctx the parse tree
*/
void exitMethod_call(DecafParser.Method_callContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#method_name}.
* @param ctx the parse tree
*/
void enterMethod_name(DecafParser.Method_nameContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#method_name}.
* @param ctx the parse tree
*/
void exitMethod_name(DecafParser.Method_nameContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#location}.
* @param ctx the parse tree
*/
void enterLocation(DecafParser.LocationContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#location}.
* @param ctx the parse tree
*/
void exitLocation(DecafParser.LocationContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#expr}.
* @param ctx the parse tree
*/
void enterExpr(DecafParser.ExprContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#expr}.
* @param ctx the parse tree
*/
void exitExpr(DecafParser.ExprContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#callout_arg}.
* @param ctx the parse tree
*/
void enterCallout_arg(DecafParser.Callout_argContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#callout_arg}.
* @param ctx the parse tree
*/
```

```
void exitCallout_arg(DecafParser.Callout_argContext ctx);
/**
 * Enter a parse tree produced by {@link DecafParser#bin_op}.
 * @param ctx the parse tree
 */
void enterBin_op(DecafParser.Bin_opContext ctx);
/**
 * Exit a parse tree produced by {@link DecafParser#bin_op}.
 * @param ctx the parse tree
 */
void exitBin_op(DecafParser.Bin_opContext ctx);
/**
 * Enter a parse tree produced by {@link DecafParser#arith_op}.
 * @param ctx the parse tree
 */
void enterArith_op(DecafParser.Arith_opContext ctx);
/**
 * Exit a parse tree produced by {@link DecafParser#arith_op}.
 * @param ctx the parse tree
 */
void exitArith_op(DecafParser.Arith_opContext ctx);
/**
 * Enter a parse tree produced by {@link DecafParser#rel_op}.
 * @param ctx the parse tree
 */
void enterRel_op(DecafParser.Rel_opContext ctx);
/**
 * Exit a parse tree produced by {@link DecafParser#rel_op}.
 * @param ctx the parse tree
 */
void exitRel_op(DecafParser.Rel_opContext ctx);
/**
 * Enter a parse tree produced by {@link DecafParser#eq_op}.
 * @param ctx the parse tree
 */
void enterEq_op(DecafParser.Eq_opContext ctx);
/**
 * Exit a parse tree produced by {@link DecafParser#eq_op}.
 * @param ctx the parse tree
 */
void exitEq_op(DecafParser.Eq_opContext ctx);
/**
 * Enter a parse tree produced by {@link DecafParser#cond_op}.
 * @param ctx the parse tree
 */
void enterCond_op(DecafParser.Cond_opContext ctx);
/**
 * Exit a parse tree produced by {@link DecafParser#cond_op}.
 * @param ctx the parse tree
 */
void exitCond_op(DecafParser.Cond_opContext ctx);
/**
 * Enter a parse tree produced by {@link DecafParser#literal}.
 * @param ctx the parse tree
 */
void enterLiteral(DecafParser.LiteralContext ctx);
/**
```

```
* Exit a parse tree produced by {@link DecafParser#literal}.
* @param ctx the parse tree
*/
void exitLiteral(DecafParser.LiteralContext ctx);
/**
* Enter a parse tree produced by {@link DecafParser#bool_literal}.
* @param ctx the parse tree
*/
void enterBool_literal(DecafParser.Bool_literalContext ctx);
/**
* Exit a parse tree produced by {@link DecafParser#bool_literal}.
* @param ctx the parse tree
*/
void exitBool_literal(DecafParser.Bool_literalContext ctx);
}
```