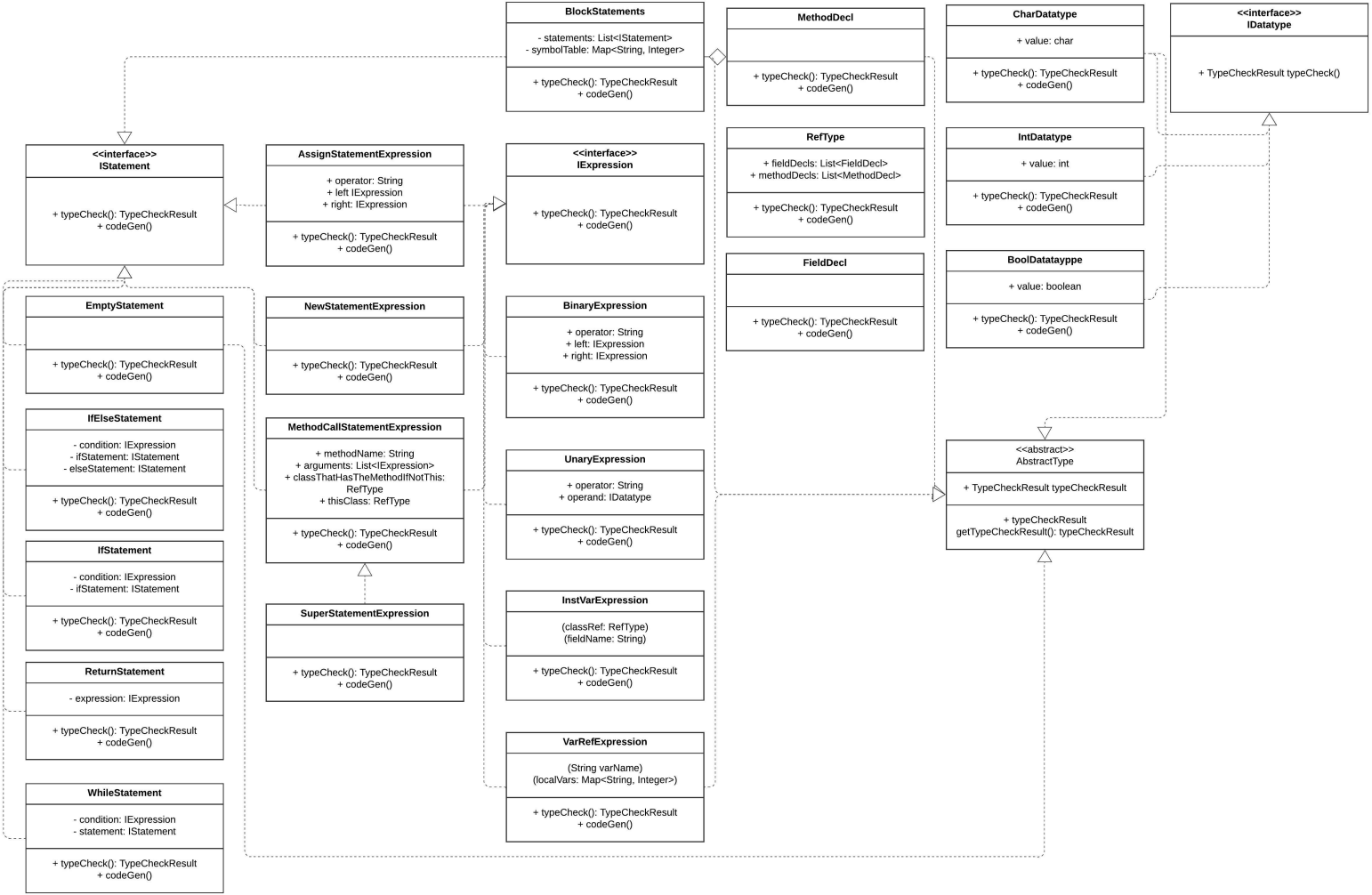
BlockStatements
- statements: List<IStatement>
- symbolTable: Map<String, Integer>
+ typeCheck(): TypeCheckResult
+ codeGen()
MethodDecl
+ typeCheck(): TypeCheckResult
+ codeGen()
CharDatatype
+ value: char
+ typeCheck(): TypeCheckResult
+ codeGen()
<<interface>>
IDatatype
+ TypeCheckResult typeCheck()
<<interface>>
IStatement
+ typeCheck(): TypeCheckResult
+ codeGen()
AssignStatementExpression
+ operator: String
+ left IExpression
+ right: IExpression
+ typeCheck(): TypeCheckResult
+ codeGen()
<<interface>>
IExpression
+ typeCheck(): TypeCheckResult
+ codeGen()
RefType
+ fieldDecls: List<FieldDecl>
+ methodDecls: List<MethodDecl>
+ typeCheck(): TypeCheckResult
+ codeGen()
IntDatatype
+ value: int
+ typeCheck(): TypeCheckResult
+ codeGen()
FieldDecl
+ typeCheck(): TypeCheckResult
+ codeGen()
BoolDatatayppe
+ value: boolean
+ typeCheck(): TypeCheckResult
+ codeGen()
EmptyStatement
+ typeCheck(): TypeCheckResult
+ codeGen()
NewStatementExpression
+ typeCheck(): TypeCheckResult
+ codeGen()
BinaryExpression
+ operator: String
+ left: IExpression
+ right: IExpression
+ typeCheck(): TypeCheckResult
+ codeGen()
IfElseStatement
- condition: IExpression
- ifStatement: IStatement
- elseStatement: IStatement
+ typeCheck(): TypeCheckResult
+ codeGen()
MethodCallStatementExpression
+ methodName: String
+ arguments: List<IExpression>
+ classThatHasTheMethodIfNotThis: RefType
+ thisClass: RefType
+ typeCheck(): TypeCheckResult
+ codeGen()
UnaryExpression
+ operator: String
+ operand: IDatatype
+ typeCheck(): TypeCheckResult
+ codeGen()
<<abstract>>
AbstractType
+ TypeCheckResult typeCheckResult
+ typeCheckResult
getTypeCheckResult(): typeCheckResult
IfStatement
- condition: IExpression
- ifStatement: IStatement
+ typeCheck(): TypeCheckResult
+ codeGen()
SuperStatementExpression
+ typeCheck(): TypeCheckResult
+ codeGen()
InstVarExpression
(classRef: RefType)
(fieldName: String)
+ typeCheck(): TypeCheckResult
+ codeGen()
ReturnStatement
- expression: IExpression
+ typeCheck(): TypeCheckResult
+ codeGen()
VarRefExpression
(String varName)
(localVars: Map<String, Integer>)
+ typeCheck(): TypeCheckResult
+ codeGen()
WhileStatement
- condition: IExpression
- statement: IStatement
+ typeCheck(): TypeCheckResult
+ codeGen()