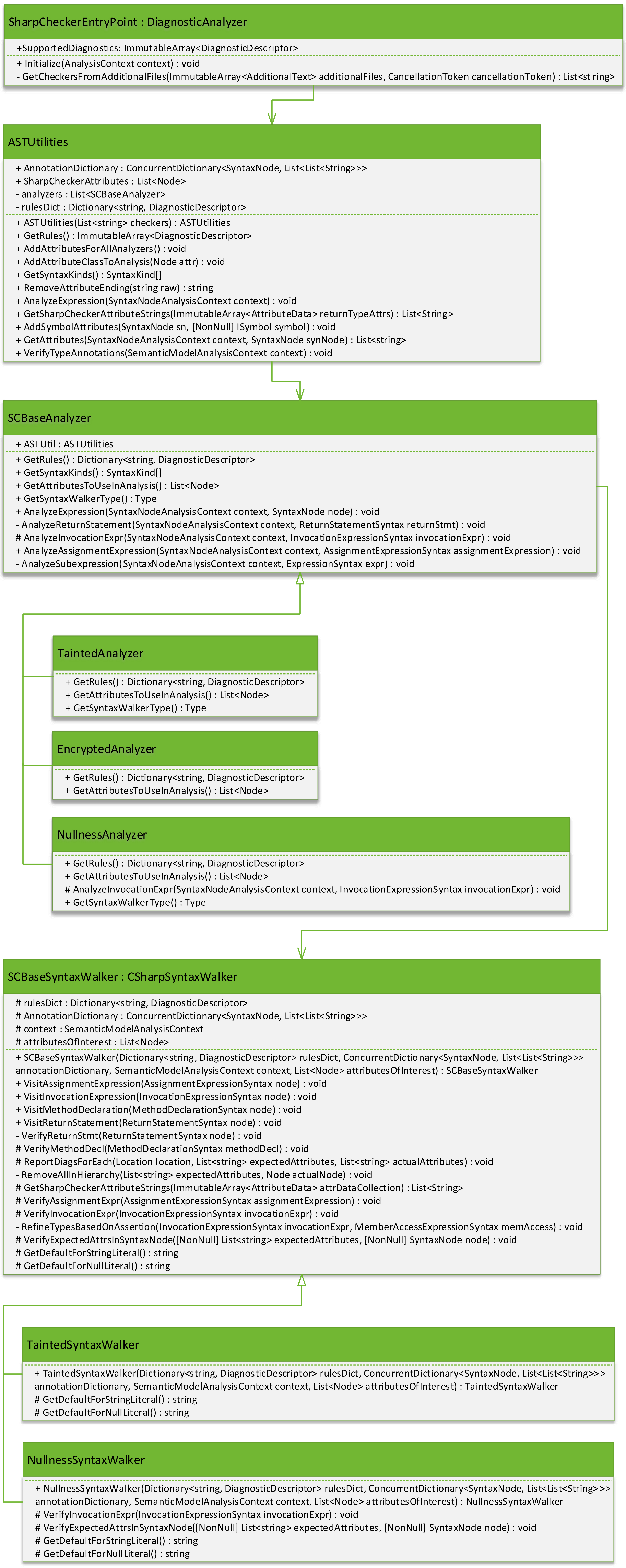
```
SharpCheckerEntryPoint : DiagnosticAnalyzer
  +SupportedDiagnostics: ImmutableArray<DiagnosticDescriptor>
  + Initialize(AnalysisContext context) : void
  - GetCheckersFromAdditionalFiles(ImmutableArray<AdditionalText> additionalFiles, CancellationToken cancellationToken) : List<st ring>

ASTUtilities
  + AnnotationDictionary : ConcurrentDictionary<SyntaxNode, List<List<String>>>
  + SharpCheckerAttributes : List<Node>
  - analyzers : List<SCBaseAnalyzer>
  - rulesDict : Dictionary<string, DiagnosticDescriptor>
  + ASTUtilities(List<string> checkers) : ASTUtilities
  + GetRules() : ImmutableArray<DiagnosticDescriptor>
  + AddAttributesForAllAnalyzers() : void
  + AddAttributeClassToAnalysis(Node attr) : void
  + GetSyntaxKinds() : SyntaxKind[]
  + RemoveAttributeEnding(string raw) : string
  + AnalyzeExpression(SyntaxNodeAnalysisContext context) : void
  + GetSharpCheckerAttributeStrings(ImmutableArray<AttributeData> returnTypeAttrs) : List<String>
  + AddSymbolAttributes(SyntaxNode sn, [NonNull] ISymbol symbol) : void
  + GetAttributes(SyntaxNodeAnalysisContext context, SyntaxNode synNode) : List<string>
  + VerifyTypeAnnotations(SemanticModelAnalysisContext context) : void

SCBaseAnalyzer
  + ASTUtil : ASTUtilities
  + GetRules() : Dictionary<string, DiagnosticDescriptor>
  + GetSyntaxKinds() : SyntaxKind[]
  + GetAttributesToUseInAnalysis() : List<Node>
  + GetSyntaxWalkerType() : Type
  + AnalyzeExpression(SyntaxNodeAnalysisContext context, SyntaxNode node) : void
  - AnalyzeReturnStatement(SyntaxNodeAnalysisContext context, ReturnStatementSyntax returnStmt) : void
  # AnalyzeInvocationExpr(SyntaxNodeAnalysisContext context, InvocationExpressionSyntax invocationExpr) : void
  + AnalyzeAssignmentExpression(SyntaxNodeAnalysisContext context, AssignmentExpressionSyntax assignmentExpression) : void
  - AnalyzeSubexpression(SyntaxNodeAnalysisContext context, ExpressionSyntax expr) : void

TaintedAnalyzer
  + GetRules() : Dictionary<string, DiagnosticDescriptor>
  + GetAttributesToUseInAnalysis() : List<Node>
  + GetSyntaxWalkerType() : Type

EncryptedAnalyzer
  + GetRules() : Dictionary<string, DiagnosticDescriptor>
  + GetAttributesToUseInAnalysis() : List<Node>

NullnessAnalyzer
  + GetRules() : Dictionary<string, DiagnosticDescriptor>
  + GetAttributesToUseInAnalysis() : List<Node>
  # AnalyzeInvocationExpr(SyntaxNodeAnalysisContext context, InvocationExpressionSyntax invocationExpr) : void
  + GetSyntaxWalkerType() : Type

SCBaseSyntaxWalker : CSharpSyntaxWalker
  # rulesDict : Dictionary<string, DiagnosticDescriptor>
  # AnnotationDictionary : ConcurrentDictionary<SyntaxNode, List<List<String>>>
  # context : SemanticModelAnalysisContext
  # attributesOfInterest : List<Node>
  + SCBaseSyntaxWalker(Dictionary<string, DiagnosticDescriptor> rulesDict, ConcurrentDictionary<SyntaxNode, List<List<String>>>
  annotationDictionary, SemanticModelAnalysisContext context, List<Node> attributesOfInterest) : SCBaseSyntaxWalker
  + VisitAssignmentExpression(AssignmentExpressionSyntax node) : void
  + VisitInvocationExpression(InvocationExpressionSyntax node) : void
  + VisitMethodDeclaration(MethodDeclarationSyntax node) : void
  + VisitReturnStatement(ReturnStatementSyntax node) : void
  - VerifyReturnStmt(ReturnStatementSyntax node) : void
  # VerifyMethodDecl(MethodDeclarationSyntax methodDecl) : void
  # ReportDiagsForEach(Location location, List<string> expectedAttributes, List<string> actualAttributes) : void
  - RemoveAllInHierarchy(List<string> expectedAttributes, Node actualNode) : void
  # GetSharpCheckerAttributeStrings(ImmutableArray<AttributeData> attrDataCollection) : List<String>
  # VerifyAssignmentExpr(AssignmentExpressionSyntax assignmentExpression) : void
  # VerifyInvocationExpr(InvocationExpressionSyntax invocationExpr) : void
  - RefineTypesBasedOnAssertion(InvocationExpressionSyntax invocationExpr, MemberAccessExpressionSyntax memAccess) : void
  # VerifyExpectedAttrsInSyntaxNode([NonNull] List<string> expectedAttributes, [NonNull] SyntaxNode node) : void
  # GetDefaultForStringLiteral() : string
  # GetDefaultForNullLiteral() : string

TaintedSyntaxWalker
  + TaintedSyntaxWalker(Dictionary<string, DiagnosticDescriptor> rulesDict, ConcurrentDictionary<SyntaxNode, List<List<String>>>
  annotationDictionary, SemanticModelAnalysisContext context, List<Node> attributesOfInterest) : TaintedSyntaxWalker
  # GetDefaultForStringLiteral() : string
  # GetDefaultForNullLiteral() : string

NullnessSyntaxWalker
  + NullnessSyntaxWalker(Dictionary<string, DiagnosticDescriptor> rulesDict, ConcurrentDictionary<SyntaxNode, List<List<String>>>
  annotationDictionary, SemanticModelAnalysisContext context, List<Node> attributesOfInterest) : NullnessSyntaxWalker
  # VerifyInvocationExpr(InvocationExpressionSyntax invocationExpr) : void
  # VerifyExpectedAttrsInSyntaxNode([NonNull] List<string> expectedAttributes, [NonNull] SyntaxNode node) : void
  # GetDefaultForStringLiteral() : string
  # GetDefaultForNullLiteral() : string
```