未使用的变量检查器

1. 功能名称
   未使用的变量检查器 `findUnusedVariables`
2. 功能概述
   本功能旨在静态分析用户提供的仓颉源代码，以**识别并报告在函数作用域内已声明但未被后续代码有效使用的局部变量**。
3. 功能范围
   a. 包含的功能
      i. 对用户提供的单个仓颉源代码文件（以字符串形式）进行分析。
      ii. 在未使用的局部变量前添加`` `/* UNUSED VAR*/` ``。
   b. 不包含的功能
      i. 跨多个文件或模块的变量使用情况分析。
      ii. 对类成员变量（属性）、接口成员或全局作用域变量的未使用情况进行分析（除非它们在函数内部被重新声明为同名局部变量且该局部变量未使用）。
      iii. 自动删除任何被识别为未使用的代码。
      iv. 对函数参数的未使用情况进行分析。
4. 功能性需求
   a. 输入 (Input)
      i. 合法输入
         一个字符串，其中包含符合仓颉语言语法规范的源代码。输入的代码应能被仓颉标准库的 std.ast 包成功解析为抽象语法树 (AST)。
      ii. 数据规范：
         使用仓颉 std.ast 包解析之后的语法树只存在这些类型的节点：`AssignExpr`，`BinaryExpr`，`Block`，`Body`，`ClassDecl`，`Decl`，`Expr`，`ForInExpr`，`FuncDecl`，`FuncParam`，`IfExpr`，`ImportContent`，`ImportList`，`LitConstExpr`，`Modifier`，`Node`，`PackageHeader`，`ParenExpr`，`Pattern`，`PrimitiveType`，`RangeExpr`，`RefExpr`，`RefType`，`ReturnExpr`，`Token`，`Tokens`，`TypeNode`，`UnaryExpr`，`VarDecl`，`VarPattern`
   b. 处理行为 (Processing Behavior)
      i. 未使用变量的定义：一个在函数内部声明的局部变量，如果在其声明语句之后，直至该变量所属作用域（即其所在的函数体）结束，其标识符都未在任何可执行语句或表达式中被读取或引用，则该变量被视为“未使用”。变量的声明本身不构成“使用”。
      ii. 检测范围：检测应严格限制在**函数内部声明的局部变量**。不检测函数参数列表中的参数是否未使用。分析应针对代码的静态结构，不执行代码。
   c. 输出 (Output)
      i. 输出增加注释后的完成代码，对于不符合数据限制的输入，返回字符串 **`''ILLEGAL INPUT''`**