Grammar for Compiler Project Language

⟨*Program*⟩ :== "**PROGRAM**" ⟨*Identifier*⟩ "**;**"
[⟨*Declarations*⟩] { ⟨*ProcDeclaration*⟩ } ⟨*Block*⟩ "**.**"

⟨*Declarations*⟩ :== "**VAR**" ⟨*Variable*⟩ { "**,**" ⟨*Variable*⟩ } "**;**"

⟨*ProcDeclaration*⟩ :== "**PROCEDURE**" ⟨*Identifier*⟩ [⟨*ParameterList*⟩] "**;**"
[⟨*Declarations*⟩] { ⟨*ProcDeclaration*⟩ } ⟨*Block*⟩ "**;**"

⟨*ParameterList*⟩ :== "(" ⟨*FormalParameter*⟩ { "**,**" ⟨*FormalParameter*⟩ } ")"

⟨*FormalParameter*⟩ :== ["**REF**"] ⟨*Variable*⟩

⟨*Block*⟩ :== "**BEGIN**" { ⟨*Statement*⟩ "**;**" } "**END**"

⟨*Statement*⟩ :== ⟨*SimpleStatement*⟩ | ⟨*WhileStatement*⟩ | ⟨*IfStatement*⟩ |
⟨*ReadStatement*⟩ | ⟨*WriteStatement*⟩

⟨*SimpleStatement*⟩ :== ⟨*VarOrProcName*⟩ ⟨*RestOfStatement*⟩

⟨*RestOfStatement*⟩ :== ⟨*ProcCallList*⟩ | ⟨*Assignment*⟩ | ϵ

⟨*ProcCallList*⟩ :== "(" ⟨*ActualParameter*⟩ { "**,**" ⟨*ActualParameter*⟩ } ")"

⟨*Assignment*⟩ :== ":=" ⟨*Expression*⟩

⟨*ActualParameter*⟩ :== ⟨*Variable*⟩ | ⟨*Expression*⟩

⟨*WhileStatement*⟩ :== "**WHILE**" ⟨*BooleanExpression*⟩ "**DO**" ⟨*Block*⟩

⟨*IfStatement*⟩ :== "**IF**" ⟨*BooleanExpression*⟩ "**THEN**" ⟨*Block*⟩ ["**ELSE**" ⟨*Block*⟩]

⟨*ReadStatement*⟩ :== "**READ**" "(" ⟨*Variable*⟩ { "**,**" ⟨*Variable*⟩ } ")"

⟨*WriteStatement*⟩ :== "**WRITE**" "(" ⟨*Expression*⟩ { "**,**" ⟨*Expression*⟩ } ")"

⟨*Expression*⟩ :== ⟨*CompoundTerm*⟩ { ⟨*AddOp*⟩ ⟨*CompoundTerm*⟩ }

⟨*CompoundTerm*⟩ :== ⟨*Term*⟩ { ⟨*MultOp*⟩ ⟨*Term*⟩ }

⟨*Term*⟩ :== ["−"] ⟨*SubTerm*⟩

⟨*SubTerm*⟩ :== ⟨*Variable*⟩ | ⟨*IntConst*⟩ | "(" ⟨*Expression*⟩ ")"

⟨*BooleanExpression*⟩ :== ⟨*Expression*⟩ ⟨*RelOp*⟩ ⟨*Expression*⟩

⟨*AddOp*⟩ :== "+" | "−"

⟨*MultOp*⟩ :== "∗" | "/"

⟨*RelOp*⟩ :== "=" | "<=" | ">=" | "<" | ">"

⟨*Variable*⟩ :== ⟨*Identifier*⟩

⟨*VarOrProcName*⟩ :== ⟨*Identifier*⟩

⟨*Identifier*⟩ :== ⟨*Alpha*⟩ { ⟨*AlphaNum*⟩ }

⟨*IntConst*⟩ :== ⟨*Digit*⟩ { ⟨*Digit*⟩ }

⟨*AlphaNum*⟩ :== ⟨*Alpha*⟩ | ⟨*Digit*⟩

⟨*Alpha*⟩ :== "**A**" ... "**Z**" | "**a**" ... "**z**"

⟨*Digit*⟩ :== "**0**" ... "**9**"