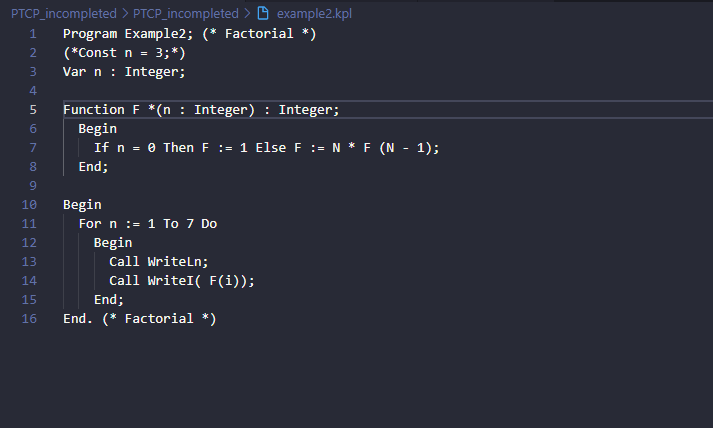
Hoàng Thanh Lâm -20194786

**Report Project 2B: Extended parser**

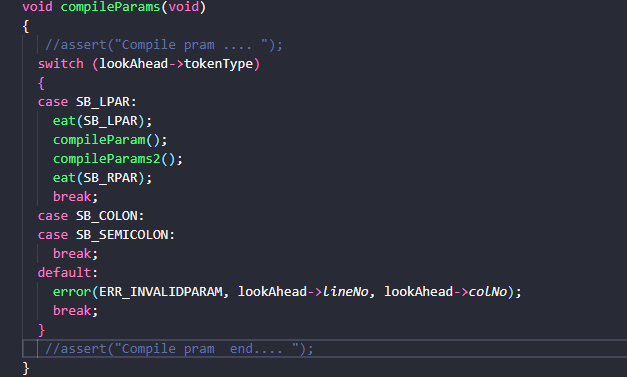
1: Modify file example2.kpl so that it causes the error "Invalid parameter!"

****

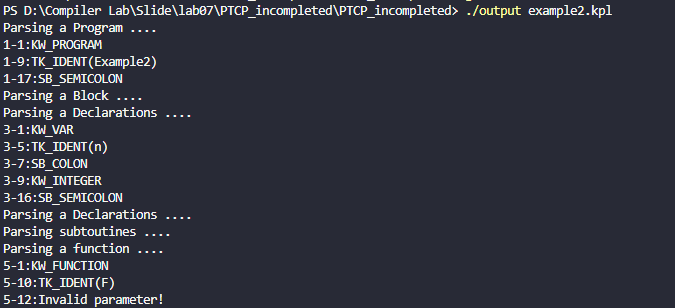
Modified code in 5th line : adding symbol \* before left parenthesis.

Then the machine run this function code compileParams in parser. We specified

that lookahead->tokenType not in any case and it switch to case default and notify error .

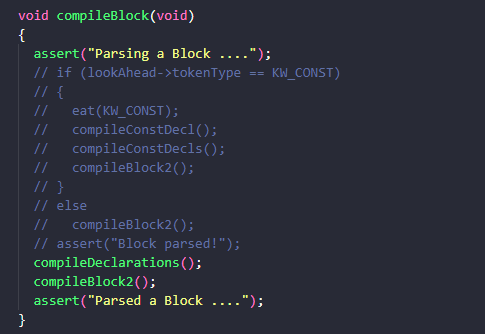


Result:

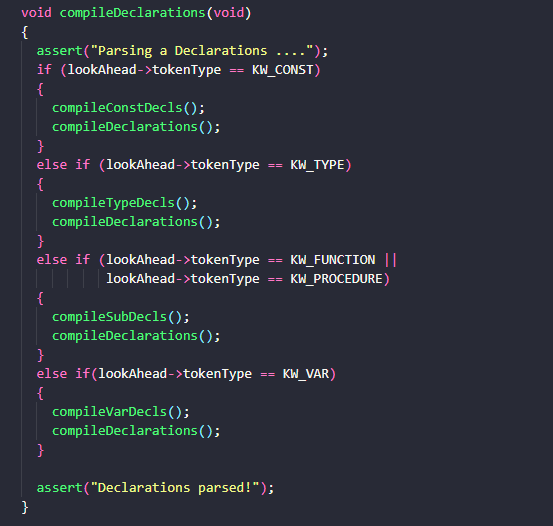


2: Suppose syntax of declarations in KPL follows the following rules

Block ::= Declarations Block2



Declarations::= ConstDecls Declarations|TypeDecls Declarations VarDecls Declarations|SubDecls Declarations|?



Constdecls ::= KW\_CONST ConstDecl ConstDecls2

ConstDecls2::= ConstDecl ConstDecls2

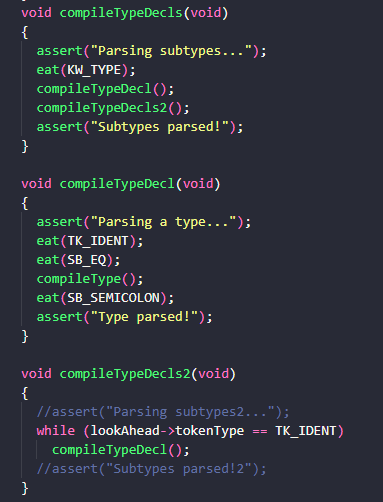
ConstDecls2::= eps



TypeDecls ::= KW\_TYPE TypeDecl TypeDecls2

TypeDecls2 ::= TypeDecl TypeDecls2

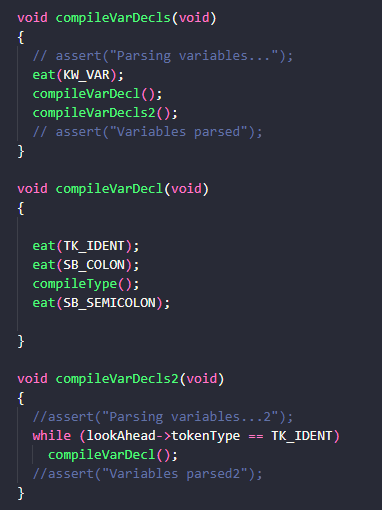
TypeDecls ::= eps



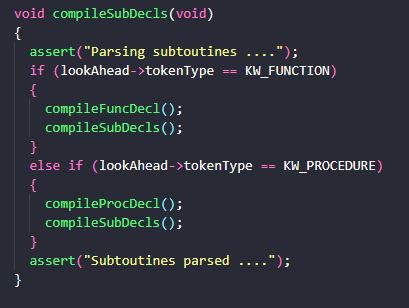
VarDecls ::= KW\_VAR VarDecl VarDeclrs2

VarDecls2 ::= VarDecl VarDecls

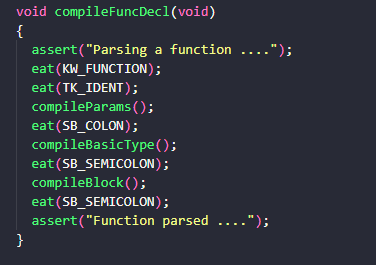
VarDecls2 ::= eps



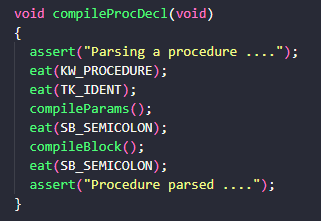
SubDecls ::= FunDecl SubDecls|ProcDecls SubDecls|?



FunDecl ::= KW\_FUNCTION TK\_IDENT Params SB\_COLON BasicType SB\_SEMICOLON Block SB\_SEMICOLON



ProcDecl ::= KW\_PROCEDURE TK\_IDENT Params SB\_SEMICOLON Block SB\_SEMICOLON



Block2 ::= KW\_BEGIN Statements KW\_END

