

BÁO CÁO THỰC HÀNH XÂY DỰNG CHƯƠNG TRÌNH DỊCH – IT3323

BÀI 2: Phân tích cú pháp – Parser

Họ và tên: Trịnh Hữu An

MSSV: 20225593

Mã lớp: 161269

Mục lục:

I.	Kết quả thực hiện với example1.kpl không lỗi	2
II.	Kết quả thực hiện với chương trình có lỗi:	4
a.	Lỗi ERM_INVALIDCONSTANT:.....	4
b.	Lỗi ERM_INVALIDTYPE:	5
c.	Lỗi ERM_INVALIDBASICTYPE:.....	7
d.	Lỗi ERM_INVALIDPARAM:.....	9
e.	Lỗi ERM_INVALIDSTATEMENT:.....	11
f.	Lỗi ERM_INVALIDARGUMENTS:	13
g.	Lỗi ERM_INVALIDCOMPARATOR:.....	16
h.	Lỗi ERM_INVALIDEXPRESSION:.....	18
i.	Lỗi ERM_INVALIDTERM:	20
j.	Lỗi ERM_INVALIDFACTOR:	23

I. Kết quả thực hiện với example1.kpl không lỗi

- Kết quả:

```
makefile × C parser.c × example1.kpl × h error.h × C scanner.c ×
Test > example1.kpl
1 Program Example1; (* Example 1 *)
2 Begin
3 End. (* Example 1 *)

Problems Output Debug Console Terminal Ports
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/example1.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(Example1)
1-17:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
Parsing Block 4 ....
Parsing subroutines ....
Subroutines parsed ....
2-1:KW_BEGIN
3-1:KW_END
Block parsed!
3-5:SB_PERIOD
Program parsed!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Giải thích quá trình parse từng bước:

1. compileProgram() được gọi:

- In: "Parsing a Program"
- eat(KW_PROGRAM) → parse Program → in: 1-1:KW_PROGRAM
- eat(TK_IDENT) → parse Example1 → in: 1-9:TK_IDENT(Example1)
- eat(SB_SEMICOLON) → parse ; → in: 1-17:SB_SEMICOLON
- Gọi compileBlock() để parse phần block

2. compileBlock() được gọi:

- In: "Parsing a Block"
- Kiểm tra lookAhead->tokenType == KW_CONST → KHÔNG (token tiếp theo là Begin)
- Vào nhánh else → gọi compileBlock2()

3. compileBlock2() được gọi:

- In: "Parsing Block 2"
- Kiểm tra lookahead->tokenType == KW_TYPE → KHÔNG (vẫn là Begin)
- Vào nhánh else → gọi compileBlock3()

4. compileBlock3() được gọi:

- In: "Parsing Block 3"
- Kiểm tra lookahead->tokenType == KW_VAR → KHÔNG (vẫn là Begin)
- Vào nhánh else → gọi compileBlock4()

5. compileBlock4() được gọi:

- In: "Parsing Block 4"
- Gọi compileSubDecls() để kiểm tra subroutines

6. compileSubDecls() được gọi:

- In: "Parsing subroutines"
- Kiểm tra lookahead->tokenType == KW_FUNCTION || KW PROCEDURE → KHÔNG (vẫn là Begin)
- Vòng lặp while không chạy (không có subroutines)
- In: "Subroutines parsed"
- Trả về, compileBlock4() tiếp tục gọi compileBlock5()

7. compileBlock5() được gọi:

- eat(KW_BEGIN) → parse Begin → in: 2-1:KW_BEGIN
- compileStatements() → không có statements (token tiếp theo là End)
- eat(KW_END) → parse End → in: 3-1:KW_END
- Trả về

8. Quay lại compileBlock():

- In: "Block parsed!"
- Trả về

9. Quay lại compileProgram():

- eat(SB_PERIOD) → parse . → in: 3-5:SB_PERIOD
- In: "Program parsed!"
- Hoàn tất

II. Kết quả thực hiện với chương trình có lỗi:

a. Lỗi ERM_INVALIDCONSTANT:

```
Test > ERM_INVALIDCONSTANT.kpl
1 Program InvalidConstant;
2 Const c = +;
3 Begin
4 End.
5
6
Debug Console (Ctrl+Shift+Alt+Y)
Problems Output Debug Console Terminal Ports wsl
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDCONSTANT.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(InvalidConstant)
1-24:SB_SEMICOLON
Parsing a Block ....
2-1:KW_CONST
2-7:TK_IDENT(c)
2-9:SB_EQ
2-11:Invalid constant!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình InvalidConstant, và dấu ; thành công.

2. Parse block và khai báo CONST:

- Parser nhận diện Const và bắt đầu parse constant declaration.
- Parse identifier c và dấu = thành công.

3. Parse giá trị constant:

- Parser gọi compileConstant() → compileConstant2() → compileUnsignedConstant().
- compileUnsignedConstant() chỉ chấp nhận:
- TK_NUMBER (số nguyên)

- TK_CHAR (ký tự)
- TK_IDENT (tên constant khác)
- Token hiện tại là + (SB_PLUS), không thuộc các loại trên.

4. Phát hiện và báo lỗi:

- Parser vào nhánh default trong compileUnsignedConstant().
- Gọi error(ERR_INVALID_CONSTANT, 2, 11).
- In thông báo: 2-11:Invalid constant!

b. Lỗi ERM_INVALIDTYPE:

```

Test > ERM_INVALIDTYPE.kpl
1 Program TestInvalidType;
2 Type t = +;
3 Begin
4 End.
5
6

Problems Output Debug Console Terminal Ports
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDTYPE.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(TestInvalidType)
1-24:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
2-1:KW_TYPE
2-6:TK_IDENT(t)
2-8:SB_EQ
2-10:Invalid type!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ 
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình TestInvalidType, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"

3. Parse khai báo TYPE:

- Parser nhận diện Type → in: 2-1:KW_TYPE

- Gọi compileTypeDecl() để parse type declaration.
- Parse identifier t → in: 2-6:TK_IDENT(t)
- Parse dấu = → in: 2-8:SB_EQ
- Gọi compileType() để parse giá trị type.

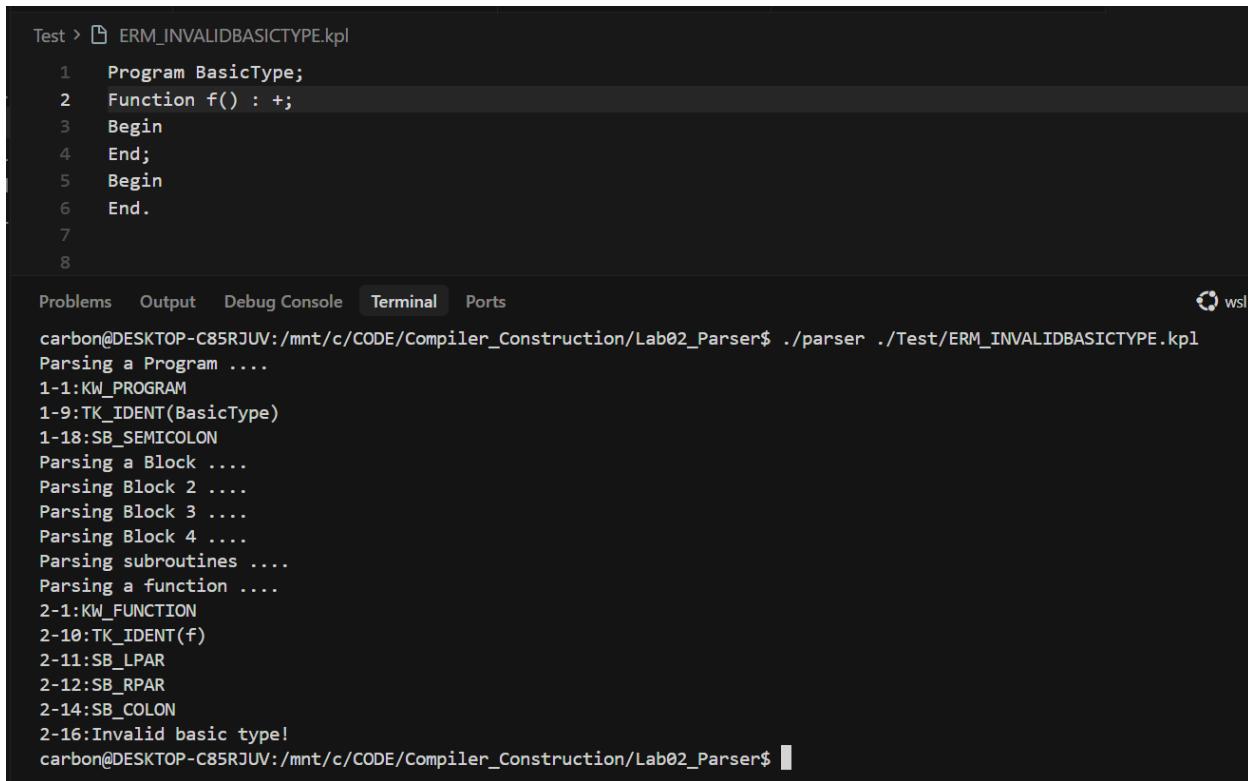
4. Parse giá trị type:

- compileType() chỉ chấp nhận:
- KW_INTEGER (kiểu số nguyên)
- KW_CHAR (kiểu ký tự)
- KW_ARRAY (kiểu mảng)
- TK_IDENT (tên type đã định nghĩa)
- Token hiện tại là + (SB_PLUS), không thuộc các loại trên.

5. Phát hiện và báo lỗi:

- Parser vào nhánh default trong compileType().
- Gọi error(ERR_INVALIDTYPE, 2, 10).
- In thông báo: 2-10:Invalid type!

c. Lỗi ERM_INVALIDBASICTYPE:



```
Test > ERM_INVALIDBASICTYPE.kpl
1 Program BasicType;
2 Function f() : +;
3 Begin
4 End;
5 Begin
6 End.
7
8

Problems Output Debug Console Terminal Ports wsl
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDBASICTYPE.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(BasicType)
1-18:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
Parsing Block 4 ....
Parsing subroutines ....
Parsing a function ....
2-1:KW_FUNCTION
2-10:TK_IDENT(f)
2-11:SB_LPAR
2-12:SB_RPAR
2-14:SB_COLON
2-16:Invalid basic type!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình BasicType, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"
- Không có VAR → gọi compileBlock4().
- In: "Parsing Block 4"
- Gọi compileSubDecls() để kiểm tra subroutines.

3. Parse function declaration:

- Parser nhận diện Function → in: "Parsing subroutines" và "Parsing a function"

- eat(KW_FUNCTION) → parse Function → in: 2-1:KW_FUNCTION
- eat(TK_IDENT) → parse f → in: 2-10:TK_IDENT(f)
- Gọi compileParams() → parse () (params rỗng):
- eat(SB_LPAR) → in: 2-11:SB_LPAR
- Không có params (gặp SB_RPAR ngay) → không gọi compileParam()
- eat(SB_RPAR) → in: 2-12:SB_RPAR
- eat(SB_COLON) → parse : → in: 2-14:SB_COLON
- Gọi compileBasicType() để parse kiểu trả về.

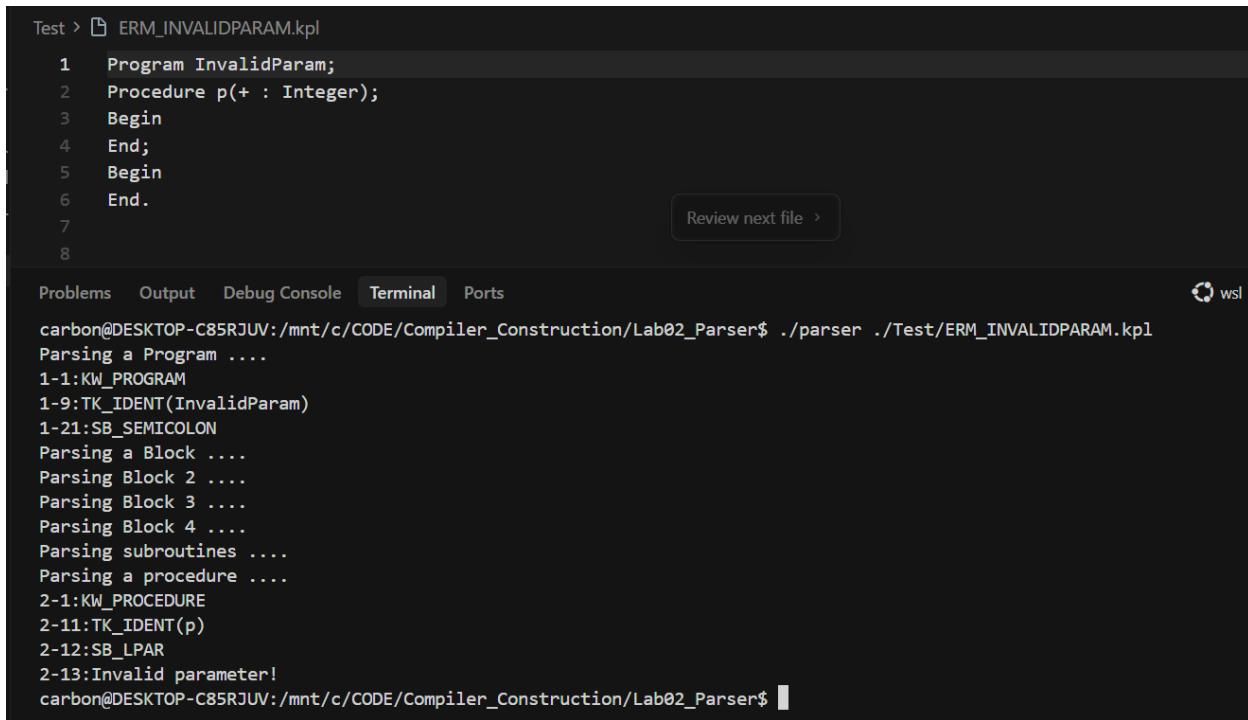
4. Parse basic type:

- compileBasicType() chỉ chấp nhận:
- KW_INTEGER (kiểu số nguyên)
- KW_CHAR (kiểu ký tự)
- Token hiện tại là + (SB_PLUS), không thuộc các loại trên.

5. Phát hiện và báo lỗi:

- Parser vào nhánh default trong compileBasicType().
- Gọi error(ERR_INVALIDBASICTYPE, 2, 16).
- In thông báo: 2-16:Invalid basic type!

d. Lỗi ERM_INVALIDPARAM:



```
Test > ERM_INVALIDPARAM.kpl
1 Program InvalidParam;
2 Procedure p(+ : Integer);
3 Begin
4 End;
5 Begin
6 End.
7
8
Review next file >

Problems Output Debug Console Terminal Ports
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDPARAM.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(InvalidParam)
1-21:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
Parsing Block 4 ....
Parsing subroutines ....
Parsing a procedure ....
2-1:KW PROCEDURE
2-11:TK_IDENT(p)
2-12:SB_LPAR
2-13:Invalid parameter!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình InvalidParam, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"
- Không có VAR → gọi compileBlock4().
- In: "Parsing Block 4"
- Gọi compileSubDecls() để kiểm tra subroutines.

3. Parse procedure declaration:

- Parser nhận diện Procedure → in: "Parsing subroutines" và "Parsing a procedure"

- eat(KW PROCEDURE) → parse Procedure → in: 2-1:KW PROCEDURE
- eat(TK IDENT) → parse p → in: 2-11:TK IDENT(p)
- Gọi compileParams() để parse parameters:
- eat(SB_LPAR) → parse (→ in: 2-12:SB_LPAR
- Kiểm tra lookAhead->tokenType != SB_RPAR → **ĐÚNG** (token là +)
- Gọi compileParam() để parse parameter

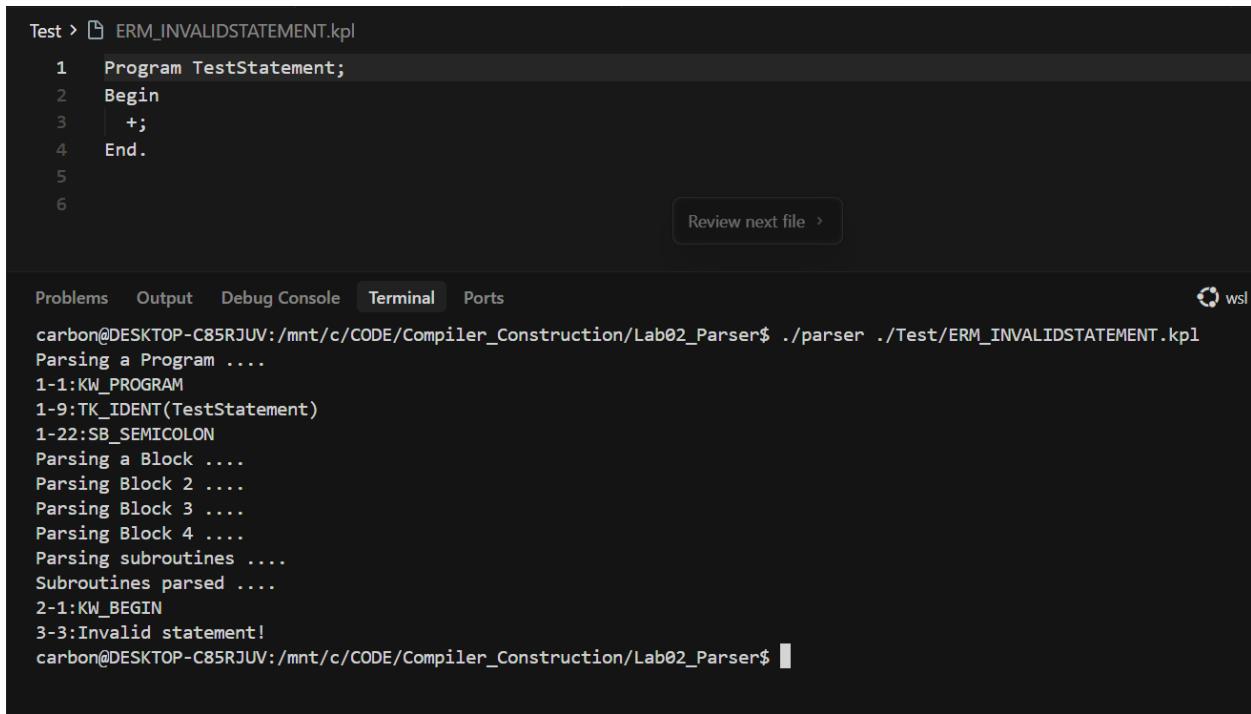
4. Parse parameter:

- compileParam() kiểm tra lookAhead->tokenType == KW_VAR → **KHÔNG**
- Kiểm tra lookAhead->tokenType != TK_IDENT → **ĐÚNG** (token là +, không phải TK_IDENT)
- Token hiện tại là + (SB_PLUS), không phải identifier hợp lệ

5. Phát hiện và báo lỗi:

- Parser phát hiện token không phải TK_IDENT trong compileParam().
- Gọi error(ERR_INVALIDPARAM, 2, 13).
- In thông báo: 2-13:Invalid parameter!

e. Lỗi ERM_INVALIDSTATEMENT:



```
Test > ERM_INVALIDSTATEMENT.kpl
1 Program TestStatement;
2 Begin
3   ;
4 End.
5
6
Review next file >

Problems Output Debug Console Terminal Ports wsl
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDSTATEMENT.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(TestStatement)
1-22:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
Parsing Block 4 ....
Parsing subroutines ....
Subroutines parsed ....
2-1:KW_BEGIN
3-3:Invalid statement!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình TestStatement, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"
- Không có VAR → gọi compileBlock4().
- In: "Parsing Block 4"
- Gọi compileSubDecls() → không có subroutines → in: "Subroutines parsed"
- Gọi compileBlock5() để parse phần BEGIN...END.

3. Parse statements:

- eat(KW_BEGIN) → parse Begin → in: 2-1:KW_BEGIN

- Gọi compileStatements() để parse các statements:
- Gọi compileStatement() để parse statement đầu tiên

4. *Parse statement:*

- compileStatement() kiểm tra lookAhead->tokenType:
- Không phải TK_IDENT (assign statement)
- Không phải KW_CALL (call statement)
- Không phải KW_BEGIN (group statement)
- Không phải KW_IF (if statement)
- Không phải KW WHILE (while statement)
- Không phải KW_FOR (for statement)
- Không phải SB_SEMICOLON, KW_END, KW_ELSE (empty statement)
- Token hiện tại là + (SB_PLUS), không khớp với bất kỳ loại statement nào

5. *Phát hiện và báo lỗi:*

- Parser vào nhánh default trong compileStatement().
- Gọi error(ERR_INVALIDSTATEMENT, 3, 3).
- In thông báo: 3-3:Invalid statement!

f. Lỗi ERM_INVALIDARGUMENTS:

```
Test > ERM_INVALIDARGUMENTS.kpl

1 Program TestArguments;
2 Procedure p(i : Integer);
3 Begin
4 End;
5 Begin
6 | Call p(+);
7 End.

Problems Output Debug Console Terminal Ports Ctrl+K to generate command
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDARGUMENTS.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(TestArguments)
1-22:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
Parsing Block 4 ....
Parsing subroutines ....
Parsing a procedure ....
2-1:KW_PROCEDURE
2-11:TK_IDENT(p)
2-12:SB_LPAR
2-13:TK_IDENT(i)
2-15:SB_COLON
2-17:KW_INTEGER
2-24:SB_RPAR
2-25:SB_SEMICOLON
```

```
Test > ERM_INVALIDARGUMENTS.kpl

1 Program TestArguments;
2 Procedure p(i : Integer);
3 Begin
4 End;
5 Begin
6 | Call p(+);
7 End.

Problems Output Debug Console Terminal Ports Ctrl+K to generate command
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDARGUMENTS.kpl
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
Parsing Block 4 ....
Parsing subroutines ....
Subroutines parsed ....
3-1:KW_BEGIN
4-1:KW_END
Block parsed!
4-4:SB_SEMICOLON
Procedure parsed ....
Subroutines parsed ....
5-1:KW_BEGIN
Parsing a call statement ....
6-3:KW_CALL
6-8:TK_IDENT(p)
6-9:SB_LPAR
6-10:Invalid arguments!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình TestArguments, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"
- Không có VAR → gọi compileBlock4().
- In: "Parsing Block 4"
- Gọi compileSubDecls() để kiểm tra subroutines.

3. Parse procedure declaration:

- Parser nhận diện Procedure → in: "Parsing subroutines" và "Parsing a procedure"
- Parse Procedure p(i : Integer); thành công:
 - 2-1:KW PROCEDURE
 - 2-11:TK IDENT(p)
 - 2-12:SB_LPAR
 - 2-13:TK IDENT(i)
 - 2-15:SB_COLON
 - 2-17:KW_INTEGER
 - 2-24:SB_RPAR
 - 2-25:SB_SEMICOLON
- Parse block của procedure thành công → in: "Procedure parsed"

4. Parse main block:

- eat(KW_BEGIN) → parse Begin → in: 5-1:KW BEGIN

- Gọi compileStatements() để parse statements

5. *Parse call statement:*

- Parser nhận diện Call → in: "Parsing a call statement"
- eat(KW_CALL) → parse Call → in: 6-3:KW_CALL
- eat(TK_IDENT) → parse p → in: 6-8:TK_IDENT(p)
- Gọi compileArguments() để parse arguments:
- eat(SB_LPAR) → parse (→ in: 6-9:SB_LPAR
- Kiểm tra lookAhead->tokenType != SB_RPAR → **ĐÚNG** (token là +)
- Kiểm tra token hợp lệ cho expression:
- Token hiện tại là + (SB_PLUS)
- Không phải TK_NUMBER, TK_CHAR, TK_IDENT, SB_LPAR, hoặc SB_MINUS
- Token không hợp lệ để bắt đầu expression

6. *Phát hiện và báo lỗi:*

- Parser phát hiện token không hợp lệ trong compileArguments().
- Gọi error(ERR_INVALIDARGUMENTS, 6, 10).
- In thông báo: 6-10:Invalid arguments!

g. Lỗi ERM_INVALIDCOMPARATOR:

The screenshot shows a terminal window with the following content:

```
Test > C:\CODE\Compiler_Construction\Lab02_Parser\Test\ERM_INVALIDSTATEMENT.kpl
1 Program TestComparator;
2 Var i : Integer;
3 Begin
4   If i + 5 Then
5     i := 1;
6 End.
7
8
```

Terminal tab selected:

```
Problems Output Debug Console Terminal Ports
```

Output:

```
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDCOMPARATOR.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(TestComparator)
1-23:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
2-1:KW_VAR
2-5:TK_IDENT(i)
2-7:SB_COLON
2-9:KW_INTEGER
2-16:SB_SEMICOLON
Parsing Block 4 ....
Parsing subroutines ....
Subroutines parsed ....
3-1:KW_BEGIN
Parsing an if statement ....
4-3:KW_IF
Parsing an expression
4-6:TK_IDENT(i)
4-8:SB_PLUS
4-10:TK_NUMBER(5)
Expression parsed
4-12:Invalid comparator!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình TestComparator, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"
- Parser nhận diện Var → in: 2-1:KW_VAR
- Parse variable declaration thành công:
 - 2-5:TK_IDENT(i)
 - 2-7:SB_COLON
 - 2-9:KW_INTEGER

- 2-16:SB_SEMICOLON
- Gọi compileBlock4() → in: "Parsing Block 4"
- Gọi compileSubDecls() → không có subroutines → in: "Subroutines parsed"
- Gọi compileBlock5() để parse phần BEGIN...END.

3. Parse if statement:

- eat(KW_BEGIN) → parse Begin → in: 3-1:KW_BEGIN
- Gọi compileStatements() → gọi compileStatement()
- Parser nhận diện If → in: "Parsing an if statement"
- eat(KW_IF) → parse If → in: 4-3:KW_IF
- Gọi compileCondition() để parse điều kiện

4. Parse condition:

- compileCondition() gọi compileExpression() để parse expression đầu tiên:
- In: "Parsing an expression"
- Parse i + 5 thành công:
 - 4-6:TK_IDENT(i)
 - 4-8:SB_PLUS
 - 4-10:TK_NUMBER(5)
- In: "Expression parsed"
- Gọi compileCondition2() để parse comparator và expression thứ hai

5. Parse comparator:

- compileCondition2() kiểm tra lookAhead->tokenType:
- Không phải SB_EQ (=)
- Không phải SB_NEQ (!=)
- Không phải SB_LT (<)
- Không phải SB_LE (<=)
- Không phải SB_GT (>)

- Không phải SB_GE (\geq)
- Token hiện tại là KW_THEN, không phải comparator hợp lệ

6. Phát hiện và báo lỗi:

- Parser vào nhánh default trong compileCondition2().
- Gọi error(ERR_INVALIDCOMPARATOR, 4, 12).
- In thông báo: 4-12:Invalid comparator!

h. Lỗi ERM_INVALIDEXPRESSION:

```

Test > ERM_INVALIDEXPRESSION.kpl
1 Program TestExpression;
2 Var i : Integer;
3 Begin
4   i := 5 ;
5 End.
6
7

Problems Output Debug Console Terminal Ports
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDEXPRESSION.kpl
Parsing a Program ....
1-1:TK_PROGRAM
1-9:TK_IDENTIFIER(TestExpression)
1-23:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
2-1:TK_VAR
2-5:TK_IDENTIFIER(i)
2-7:SB_COLON
2-9:KW_INTEGER
2-16:SB_SEMICOLON
Parsing Block 4 ....
Parsing subroutines ....
Subroutines parsed ....
3-1:KW_BEGIN
Parsing an assign statement ....
4-3:TK_IDENTIFIER(i)
4-6:SB_ASSIGN
Parsing an expression
4-8:TK_NUMBER(5)
4-10:Invalid expression!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ 
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình TestExpression, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"

- Parser nhận diện Var → in: 2-1:KW_VAR
- Parse variable declaration thành công:
- 2-5:TK_IDENT(i)
- 2-7:SB_COLON
- 2-9:KW_INTEGER
- 2-16:SB_SEMICOLON
- Gọi compileBlock4() → in: "Parsing Block 4"
- Gọi compileSubDecls() → không có subroutines → in: "Subroutines parsed"
- Gọi compileBlock5() để parse phần BEGIN...END.

3. Parse assignment statement:

- eat(KW_BEGIN) → parse Begin → in: 3-1:KW_BEGIN
- Gọi compileStatements() → gọi compileStatement()
- Parser nhận diện TK_IDENT → gọi compileAssignSt()
- In: "Parsing an assign statement"
- eat(TK_IDENT) → parse i → in: 4-3:TK_IDENT(i)
- compileIndexes() → không có indexes
- eat(SB_ASSIGN) → parse := → in: 4-6:SB_ASSIGN
- Gọi compileExpression() để parse expression

4. Parse expression:

- In: "Parsing an expression"
- compileExpression() gọi compileTerm() → compileFactor()
- Parse 5 thành công → in: 4-8:TK_NUMBER(5)
- Gọi compileExpression2() → compileExpression3() để kiểm tra các toán tử tiếp theo

5. Kiểm tra expression continuation:

- compileExpression3() kiểm tra lookAhead->tokenType:
- Không phải SB_PLUS (+)

- Không phải SB_MINUS (-)
- Không phải các token trong FOLLOW set (KW_TO, KW_DO, SB_RPAR, SB_COMMMA, SB_NEQ, SB_LE, SB_LT, SB_GE, SB_GT, SB_RSEL, SB_SEMICOLON, KW_END, KW_ELSE, KW_THEN)
- Token hiện tại là SB_EQ (=)
- SB_EQ không phải toán tử hợp lệ trong expression (chỉ có + và -)
- SB_EQ cũng không nằm trong FOLLOW set của expression trong context này

6. Phát hiện và báo lỗi:

- Parser vào case SB_EQ trong compileExpression3() (đã được sửa để báo lỗi).
- Gọi error(ERR_INVALIDEXPRESSION, 4, 10).
- In thông báo: 4-10:Invalid expression!

i. Lỗi ERM_INVALIDTERM:

```

Test > ERM_INVALIDTERM.kpl
1 Program TestInvalidTerm;
2 Var i : Integer;
3 Begin
4 | i := 5 * =;
5 End.
6
7

Problems Output Debug Console Terminal Ports
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDTERM.kpl
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser ./Test/ERM_INVALIDTERM.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(TestInvalidTerm)
1-24:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
2-1:KW_VAR
2-5:TK_IDENT(i)
2-7:SB_COLON
2-9:KW_INTEGER
2-16:SB_SEMICOLON
Parsing Block 4 ....
Parsing subroutines ....
Subroutines parsed ....
3-1:KW_BEGIN
Parsing an assign statement ....
4-3:TK_IDENT(i)
4-6:SB_ASSIGN
Parsing an expression
4-8:TK_NUMBER(5)
4-10:SB_TIMES
4-12:Invalid term!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ 

```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình TestInvalidTerm, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"
- Parser nhận diện Var → in: 2-1:KW_VAR
- Parse variable declaration thành công:
- 2-5:TK_IDENT(i)
- 2-7:SB_COLON
- 2-9:KW_INTEGER
- 2-16:SB_SEMICOLON
- Gọi compileBlock4() → in: "Parsing Block 4"
- Gọi compileSubDecls() → không có subroutines → in: "Subroutines parsed"
- Gọi compileBlock5() để parse phần BEGIN...END.

3. Parse assignment statement:

- eat(KW_BEGIN) → parse Begin → in: 3-1:KW_BEGIN
- Gọi compileStatements() → gọi compileStatement()
- Parser nhận diện TK_IDENT → gọi compileAssignSt()
- In: "Parsing an assign statement"
- eat(TK_IDENT) → parse i → in: 4-3:TK_IDENT(i)
- compileIndexes() → không có indexes
- eat(SB_ASSIGN) → parse := → in: 4-6:SB_ASSIGN
- Gọi compileExpression() để parse expression

4. Parse expression và term:

- In: "Parsing an expression"
- compileExpression() gọi compileTerm() để parse term đầu tiên

- compileTerm() gọi compileFactor() → parse 5 thành công → in: 4-8:TK_NUMBER(5)
- Gọi compileTerm2() để kiểm tra các toán tử nhân/chia tiếp theo

5. *Parse term continuation:*

- compileTerm2() kiểm tra lookAhead->tokenType:
- Gặp SB_TIMES () → in: 4-10:SB_TIMES
- eat(SB_TIMES) thành công
- Kiểm tra token hợp lệ cho factor:
- Token hiện tại là = (SB_EQ)
- Không phải TK_NUMBER, TK_CHAR, TK_IDENT, SB_LPAR, hoặc SB_MINUS
- Token không hợp lệ để bắt đầu factor

6. *Phát hiện và báo lỗi:*

- Parser phát hiện token không hợp lệ trong compileTerm2() (sau khi đã kiểm tra trước khi gọi compileFactor()).
- Gọi error(ERR_INVALIDTERM, 4, 12).
- In thông báo: 4-12:Invalid term!

j. Lỗi ERM_INVALIDFACTOR:

```
Test > ERM_INVALIDFACTOR.kpl
1 Program InvalidFactor;
2 Var i : Integer;
3 Begin
4 | i := =;
5 End.
6
7

Problems Output Debug Console Terminal Ports
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$ ./parser . /Test/ERM_INVALIDFACTOR.kpl
Parsing a Program ....
1-1:KW_PROGRAM
1-9:TK_IDENT(InvalidFactor)
1-22:SB_SEMICOLON
Parsing a Block ....
Parsing Block 2 ....
Parsing Block 3 ....
2-1:KW_VAR
2-5:TK_IDENT(i)
2-7:SB_COLON
2-9:KW_INTEGER
2-16:SB_SEMICOLON
Parsing Block 4 ....
Parsing subroutines ....
Subroutines parsed ....
3-1:KW_BEGIN
Parsing an assign statement ....
4-3:TK_IDENT(i)
4-6:SB_ASSIGN
Parsing an expression
4-8:Invalid factor!
carbon@DESKTOP-C85RJUV:/mnt/c/CODE/Compiler_Construction/Lab02_Parser$
```

Quá trình parse và phát hiện lỗi

1. Parse chương trình:

- Parser nhận diện Program, tên chương trình InvalidFactor, và dấu ; thành công.

2. Parse block:

- Parser vào compileBlock() → không có CONST → gọi compileBlock2().
- In: "Parsing Block 2"
- Không có TYPE → gọi compileBlock3().
- In: "Parsing Block 3"
- Parser nhận diện Var → in: 2-1:KW_VAR
- Parse variable declaration thành công:
 - 2-5:TK_IDENT(i)
 - 2-7:SB_COLON
 - 2-9:KW_INTEGER
 - 2-16:SB_SEMICOLON

- Gọi compileBlock4() → in: "Parsing Block 4"
- Gọi compileSubDecls() → không có subroutines → in: "Subroutines parsed"
- Gọi compileBlock5() để parse phần BEGIN...END.

3. Parse assignment statement:

- eat(KW_BEGIN) → parse Begin → in: 3-1:KW_BEGIN
- Gọi compileStatements() → gọi compileStatement()
- Parser nhận diện TK_IDENT → gọi compileAssignSt()
- In: "Parsing an assign statement"
- eat(TK_IDENT) → parse i → in: 4-3:TK_IDENT(i)
- compileIndexes() → không có indexes
- eat(SB_ASSIGN) → parse := → in: 4-6:SB_ASSIGN
- Gọi compileExpression() để parse expression

4. Parse expression và factor:

- In: "Parsing an expression"
- compileExpression() gọi compileTerm() → compileFactor() để parse factor đầu tiên
- compileFactor() kiểm tra lookAhead->tokenType:
 - Không phải TK_NUMBER (số)
 - Không phải TK_CHAR (ký tự)
 - Không phải TK_IDENT (identifier)
 - Không phải SB_LPAR (dấu ngoặc mở)
 - Không phải SB_MINUS (dấu trừ)
 - Token hiện tại là = (SB_EQ), không thuộc các loại trên

5. Phát hiện và báo lỗi:

- Parser vào nhánh default trong compileFactor().
- Gọi error(ERR_INVALIDFACTOR, 4, 8).
- In thông báo: 4-8:Invalid factor!