**BÁO CÁO THỰC HÀNH XÂY DỰNG CHƯƠNG TRÌNH DỊCH  
GIẢNG VIÊN HƯỚNG DẪN: THẦY TRẦN VĨNH ĐỨC**

**BÁO CÁO LESSON 2: Parser**

**HỌ VÀ TÊN SINH VIÊN: NGUYỄN VĂN QUÝ**

**MÃ SỐ SINH VIÊN: 20210729**

**MÃ LỚP THỰC HÀNH: 151933**

**LINK GITHUB:** [link](https://github.com/quynv77/Thuc_Hanh_CTD.git)

1. **Hàm chính và khởi tạo**

* **compile(char \*fileName):** Hàm chính của chương trình biên dịch. Mở file đầu vào, khởi tạo các biến toàn cục currentToken và lookAhead, gọi hàm compileProgram() để bắt đầu phân tích cú pháp, giải phóng bộ nhớ và đóng file đầu vào.

A screen shot of a computer program

Description automatically generated

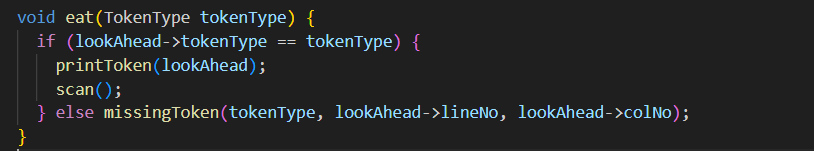
1. **HÀM XỬ LÝ TOKEN**

* **scan**(): Tiến hành đọc token tiếp theo từ luồng nhập. Token hiện tại được thay thế bởi token nhìn trước (lookAhead), và lookAhead được cập nhật bằng token hợp lệ tiếp theo từ getValidToken(). Bộ nhớ của token cũ được giải phóng.

A screen shot of a computer program

Description automatically generated

* **eat**(TokenType tokenType): Kiểm tra xem token nhìn trước có phải là loại token mong đợi (tokenType) hay không. Nếu đúng, in token ra và gọi scan() để đọc token tiếp theo. Nếu sai, báo lỗi cú pháp bằng hàm missingToken(). Ý nghĩa: Duy trì tính đúng đắn của cú pháp.



1. **Hàm phân tích cú pháp**

* **compileProgram**(): Phân tích cú pháp chương trình chính. Đảm bảo chương trình bắt đầu bằng từ khóa **PROGRAM**, theo sau là **tên**, dấu **;**, **khối lệnh**, và kết thúc bằng dấu **.**. Ý nghĩa: Đây là điểm bắt đầu phân tích cú pháp toàn bộ chương trình.

A screen shot of a computer program

Description automatically generated

* **compileBlock**(): Phân tích cú pháp cho một khối lệnh. Một khối lệnh có thể chứa khai báo hằng, kiểu, biến, và câu lệnh. Thực hiện gọi tuần tự các hàm compileConstDecls(), compileTypeDecls(), và compileVarDecls() và xử lý các câu lệnh trong khối bằng cách gọi compileStatements().

A screen shot of a computer program

Description automatically generated

* **compileBlock2**(): xử lý phần khai báo kiểu dữ liệu trong một block của chương trình nếu gặp tokenType là KW\_TYPE sau đó gọi tới compileBlock3, còn không gặp tokenType trên thì chuyển thẳng tới compileBlock3(). Đảm bảo chương trình xử lý đúng các khai báo kiểu dữ liệu trong block

A screen shot of a computer code

Description automatically generated

* **compileBlock3**(): xử lý phần khai báo biến trong block của chương trình nếu gặp tokenType là KW\_VAR sau đó xử lí phần tiếp theo của block thông qua compileBlock4, nếu không gặp tokenType trên thì chuyển thẳng tới copileBlock4().

A screen shot of a computer code

Description automatically generated

* **compileBlock4**(): xử lý phần khai báo các thủ tục và hàm con trong block của chương trình, sau đó gợi tới compileBlock5(). Đảm bảo các hàm và thủ tục được khai báo chính xác trước khi chương trình đi vào phần thân chính.

A screen shot of a computer screen

Description automatically generated

* **compileBlock5**(): xử lý toàn bộ phần thân chính của block, nơi các câu lệnh được thực thi.

A screen shot of a computer code

Description automatically generated

* **compileConstDecls**()/**compileConstDecl**(): Phân tích cú pháp cho khai báo hằng. Đảm bảo khai báo có tên hằng, dấu =, giá trị hằng, và kết thúc bằng dấu ;.

A computer screen shot of text

Description automatically generated

* **compileTypeDecls**()/**compileTypeDecl**(): Phân tích cú pháp cho khai báo kiểu. Đảm bảo khai báo có tên kiểu, dấu =, định nghĩa kiểu, và kết thúc bằng dấu ;. Xử lý các kiểu cơ bản như INTEGER, CHAR hoặc kiểu mảng.
* A computer screen with text and images

  Description automatically generated
* **compileVarDecls**()/**compileVarDecl**(): Phân tích cú pháp cho khai báo biến. Đảm bảo khai báo có tên biến, dấu :, kiểu dữ liệu, và kết thúc bằng dấu ;.

A screen shot of a computer code

Description automatically generated

* **compileSubDecls**(): Phân tích cú pháp cho khai báo thủ tục và hàm. Sử dụng vòng lặp while để kiểm tra liên tục xem token tiếp theo có phải là từ khóa FUNCTION hoặc PROCEDURE hay không. Nếu đúng, gọi compileFuncDecl() hoặc compileProcDecl() tương ứng để phân tích cú pháp cho khai báo hàm hoặc thủ tục. Vòng lặp tiếp tục cho đến khi gặp token không phải là FUNCTION hoặc PROCEDURE.
* A computer screen shot of text

  Description automatically generated
* **compileFuncDecl**(): Phân tích cú pháp cho khai báo hàm. Kiểm tra và xử lý theo thứ tự các thành phần của một khai báo hàm: từ khóa FUNCTION, tên hàm (TK\_IDENT), danh sách tham số (compileParams()), kiểu trả về (compileBasicType()), dấu chấm phẩy, thân hàm (compileBlock()), và dấu chấm phẩy kết thúc khai báo hàm.

A screen shot of a computer program

Description automatically generated

* **compileProcDecl**(): Phân tích cú pháp cho khai báo thủ tục. Tương tự như compileFuncDecl(), nhưng không có phần kiểu trả về. Kiểm tra và xử lý: từ khóa PROCEDURE, tên thủ tục (TK\_IDENT), danh sách tham số (compileParams()), dấu chấm phẩy, thân thủ tục (compileBlock()), và dấu chấm phẩy kết thúc khai báo thủ tục.

A screen shot of a computer code

Description automatically generated

* **compileUnsignedConstant**(): Phân tích cú pháp cho hằng số không dấu. Kiểm tra xem token tiếp theo có phải là TK\_NUMBER, TK\_IDENT hoặc TK\_CHAR không. Nếu đúng, gọi eat() để xử lý token đó. Nếu không phải, báo lỗi ERR\_INVALIDCONSTANT.
* A screen shot of a computer

  Description automatically generated
* **compileConstant**()/**compileConstant2**(): Phân tích cú pháp cho hằng số.

A screen shot of a computer program

Description automatically generated

A computer screen with text

Description automatically generated

* **compileType**(): Phân tích cú pháp cho kiểu dữ liệu. Kiểm tra token tiếp theo để xác định kiểu dữ liệu: KW\_INTEGER, KW\_CHAR, TK\_IDENT (kiểu do người dùng định nghĩa), hoặc KW\_ARRAY. Nếu là kiểu mảng, xử lý phần khai báo kích thước mảng và kiểu phần tử.

A screen shot of a computer program

Description automatically generated

* **compileBasicType**(): Phân tích cú pháp cho kiểu dữ liệu cơ bản (integer, char). Kiểm tra xem token tiếp theo có phải là KW\_INTEGER hoặc KW\_CHAR không. Nếu đúng, gọi eat() để xử lý token đó

A screen shot of a computer code

Description automatically generated

* **compileParams()/compileParams2**(): Phân tích cú pháp cho danh sách tham số của thủ tục/hàm.

A screen shot of a computer code

Description automatically generated

A screen shot of a computer code

Description automatically generated

* **compileStatement**(): Phân tích cú pháp cho một câu lệnh. Kiểm tra token tiếp theo để xác định loại câu lệnh: gán (TK\_IDENT), gọi thủ tục/hàm (KW\_CALL), ghép (KW\_BEGIN), if (KW\_IF), while (KW\_WHILE), for (KW\_FOR). Gọi hàm phân tích cú pháp tương ứng cho từng loại câu lệnh.

A screen shot of a computer

Description automatically generated

* **compileAssignSt**(): Phân tích cú pháp cho câu lệnh gán. Xử lý tên biến (hoặc phần tử mảng nếu có chỉ số), dấu gán (:=), và biểu thức được gán.

A computer screen shot of code

Description automatically generated

* **compileCallSt**(): Phân tích cú pháp cho câu lệnh gọi thủ tục/hàm. Xử lý từ khóa CALL, tên thủ tục/hàm, và danh sách đối số (nếu có).

A screen shot of a computer code

Description automatically generated

* **compileGroupSt**(): Phân tích cú pháp cho câu lệnh ghép (begin ... end). Xử lý từ khóa BEGIN, danh sách câu lệnh, và từ khóa END.

A screen shot of a computer program

Description automatically generated

* **compileIfSt**(): Phân tích cú pháp cho câu lệnh if. Xử lý từ khóa IF, điều kiện, từ khóa THEN, câu lệnh được thực hiện nếu điều kiện đúng, và phần else (nếu có).

A screen shot of a computer program

Description automatically generated

* **compileElseSt**(): Phân tích cú pháp cho phần else của câu lệnh if. Xử lý từ khóa ELSE và câu lệnh được thực hiện nếu điều kiện sai.

A computer code on a black background

Description automatically generated

* **compileWhileSt**(): Phân tích cú pháp cho câu lệnh while. Xử lý từ khóa WHILE, điều kiện, từ khóa DO, và câu lệnh được lặp lại.

A screen shot of a computer code

Description automatically generated

* **compileForSt**(): Phân tích cú pháp cho câu lệnh for. Xử lý từ khóa FOR, biến đếm, giá trị bắt đầu, giá trị kết thúc, từ khóa DO, và câu lệnh được lặp lại.

A screen shot of a computer program

Description automatically generated

* **compileArguments()/compileArguments2**(): Phân tích cú pháp cho danh sách đối số của một lời gọi thủ tục/hàm.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

* **compileCondition()/compileCondition2**(): Phân tích cú pháp cho điều kiện.

A screen shot of a computer program

Description automatically generated

* **compileExpression()/compileExpression2()/compileExpression3**(): Phân tích cú pháp cho biểu thức. Bao gồm các thao tác ử lý các toán tử cộng, trừ, nhân, chia; xác định các thành phần cơ bản như số, biến, hoặc biểu thức con.

A screen shot of a computer program

Description automatically generated

A black screen with white text

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen with white text

Description automatically generated

* **compileTerm()/compileTerm2**(): Phân tích cú pháp cho term.

A screen shot of a computer code

Description automatically generated

A screen shot of a computer

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

* **compileFactor**(): Phân tích cú pháp cho factor. Kiểm tra xem factor là số (TK\_NUMBER), ký tự (TK\_CHAR), biểu thức trong ngoặc đơn, biến (TK\_IDENT), hoặc lời gọi hàm.

A screen shot of a computer program

Description automatically generated

- **compileIndexes**(): Phân tích cú pháp cho chỉ tiêu của mảng. Xử lý các chỉ số nằm trong cặp dấu ngoặc vuông, mỗi chỉ số là một biểu thức. Sử dụng đệ quy để xử lý mảng nhiều chiều.

A computer screen with text

Description automatically generated

1. **Kết quả chạy code**

File example, test, result có trong tệp test của Week 2

Nhận xét:

* Code chạy đúng các test case 1, 2, 3, 4, 7
* Test case 6 chưa giải quyết được lỗi missing “;” mà trả về lỗi khác là Invalid.
* Có vẻ file result của test case 5 không khớp với example 5 do example 5 không có lỗi mà tại file test lại bắt được lỗi