ĐỀ THI CUỐI KỲ ONLINE THEO HÌNH THỨC MỞ

IT 4182 - THỰC HÀNH CHƯƠNG TRÌNH DỊCH

(Thời gian 90 phút – từ 13h đến 14h30 ngày 8/9/2021,

Được phép dùng tài liệu in và viết code trên máy tính)

**Sinh viên chép tay ra giấy, chụp bài làm và gửi ảnh lên Assignment “Thi cuối kỳ”.**

**KHÔNG LÀM BÀI, VIẾT CODE BẰNG MÁY TÍNH!**

**Bài 1:** Ngôn ngữ KPL có sự thay đổi: ***không đánh dấu chú thích bằng cặp (\* và \*)*** mà dùng hai cách

* Bắt đầu bằng // và kết thúc khi hết dòng **hoặc**
* Bắt đầu bằng /\* và kết thúc bằng \*/, có thể viết trên nhiều dòng.

Chú ý là phép chia vẫn sử dụng ký tự /

Hãy viết những thay đồi cần thiết cho các hàm của project scanner để đáp ứng với yêu cầu trên. Chú ý nhắc đến cả những đoạn code bị xóa bỏ, được thêm vào và bị thay đổi. Những đoạn không thay đổi không cần chép vào bài.

**Bài 2:** Cú pháp của lệnh CALL trong KPL có sự thay đổi như các luật 56-58 bên dưới. Từ khóa CALL không còn được dùng nữa. Như vậy cần xử lý theo kiểu văn phạm LL(2) với lệnh gán và lệnh CALL.

Hãy viết lại hàm compileAssignSt, compileCallSt và các hàm liên quan của project Parser để thực hiện sự thay đổi này. ***Chú ý xử lý trường hợp thủ tục được gọi không có tham số***

56) <AssignSt> ::= <Variable> SB\_ASSIGN <Expression>  
57) <AssignSt> ::= TK\_IDENT SB\_ASSIGN <Expression>  
**58) <CallSt> ::= TK\_IDENT <Arguments>**

**Bài 3:** Hàm **compileArguments** của bạn trong project Semantics4 có kiểm soát được:

- Danh sách tham số hình thức (params) và danh sách tham số thực sự (arguments) **phù hợp về số lượng**. Làm rõ chương trình đã xử lý trường hợp danh sách tham số thực sự có **ít** hoặc **nhiều** tham số hơn so với khai báo.

- Mỗi tham số thực sụ phải **phù hợp về kiểu với kiểu** của tham số hình thức trong khai báo. Chú ý phân biệt 2 loại:tham biến (param\_reference) và tham trị (param\_value)

**Chứng minh điều đó bằng các đoạn code minh họa kèm với lời giải thích.**

\*, Note:

- param\_reference: phải check Lvalue để còn trả về giá trị trong hàm bằng assign statement.

- gây lỗi semantic ít hỏi k hay hỏi trong bài cuối kỳ nhưng parser thì hay hỏi

\*, các câu hỏi về phần semantics:

- **Bài 1: symbol table**

+ 1 biến có kiểu mảng thì bảng ký hiệu ghi như nào? 1 chiều 2 chiều khác nhau như nào?

+ return type của hàm nó ghi như nào?

+ duplicateConstantValue thì làm gì?

+ khi có dấu – đằng trước khai báo hàm thì?

+ thêm 1 tham số ở đằng trước khai báo hằng thì sao?

+ phân biệt Param\_kind

+ Param\_list khi tính được thì thuộc về bảng ký hiệu nào

* **Bài 2: Scope Management**

+ sử dụng tên hàm với tên tham số giống nhau trong cùng 1 hàm hoặc 1 thủ tục thì sao?

+ vế trái của lệnh gán checkDeclaredLValue, vế phải checkDeclared trong factor.

+ Khi lệnh gán cho 1 hàm thì vế trái phải trùng với tên của hàm hiện hành. VD thì tạo hàm trong hàm.

+ vế phải của lệnh gán thì khó có thể kiểm tra được hàm đã được khai báo hay chưa? Tại sao?

+ Dùng 1 định danh tên là WriteC thì có được không? Tại sao không?

* **Bài 3: Type Checker**

+ vế phải của lệnh gán thì có khi nào phải kiểm tra như khai báo hằng k?

+ sử dụng nhiều chiều hơn khai báo? Ít chiều hơn khai báo?

+ chỉ số của mảng có thể dùng được ký tự k? (Có nhưng ngoài phạm vi compiler KPL chỉ dùng integer)

+ Comparator có thể so sánh được 2 ký tự k? (Được nhưng với điều kiện expession 2 bên phải k có toán tử +, -, \*, /, …)

+ Có thể dùng while(1) hay không? (KPL không cho phép 1 vì parser yêu cầu 2 bên phải là expression và có 1 op ở giữa, thứ 2 là do KPL không có lệnh nhảy ra ngoài như break)