

PPL_GK211

1. (Harmony cho BTL1)

Khi nhập học tại trường Đại học Bách Khoa, sinh viên được yêu cầu đặt một tên tài khoản gọi là **BKNetID**, gồm ba thành phần theo thứ tự: **tên**, **họ** và **chuỗi tự chọn**. Giữa tên và họ, sinh viên phải đặt một dấu chấm (.). Tên và họ là chuỗi chỉ bao gồm các ký tự chữ thường với độ dài tối thiểu là 1. Chuỗi tự chọn là một chuỗi có chiều dài từ 1 đến 5 ký tự bao gồm chữ thường, ký tự số, dấu chấm, dấu gạch dưới nhưng không được kết thúc bằng dấu chấm.

Ví dụ: *duy.tran2903*, *duy.tran.3_12* là các chuỗi BKNetID hợp lệ nhưng *duy.tran2903.* hoặc *duy2.tran2903* là BKNetID không hợp lệ.

Hãy sử dụng ANTLR để viết biểu thức chính quy cho **BKNetID** nói trên, Sinh viên phải sử dụng **fragment** để nhận trọn điểm.

2. (Harmony cho BTL1)

Trong ngôn ngữ **MiniPHP**, chương trình bao gồm các khai báo biến. Một *khai báo biến* được gắn liền với lần đầu tiên biến đó được gán giá trị. Phép gán trong MiniPHP bao gồm các thành phần theo thứ tự tên biến **VARNAME**, dấu bằng **EQ**, một biểu thức và kết thúc bởi dấu chấm phẩy **SEMI**.

Biểu thức trong MiniPHP là tổ hợp của các toán hạng và các toán tử được viết theo trung thứ tự (infix expression).

- Các toán hạng bao gồm: tên biến, hằng số nguyên **INTLIT**, hằng số thực **FLOATLIT**, hằng chuỗi **STRINGLIT**, hoặc một mảng. Có hai loại mảng trong MiniPHP là mảng chỉ số (indexed array) và mảng phối hợp tên (associative array).
 - Mảng chỉ số bắt đầu bằng từ khóa array **ARRAY** tiếp theo là một danh sách có thể rỗng các biểu thức được phân cách bởi một dấu phẩy **COMMA** và được bao lại bằng một cặp ngoặc tròn **LP** và **RP**.
 - Mảng phối hợp tên bắt đầu bằng từ khóa array **ARRAY** tiếp theo là một danh sách có thể rỗng các *cặp kết hợp* (associative pair) được phân cách bởi một dấu phẩy **COMMA** và được bao lại bằng một cặp ngoặc tròn **LP** và **RP**. Một cặp kết hợp bao gồm một tên cặp **PAIRNAME**, tiếp theo là một dấu mũi tên **ARROW** và sau đó là một biểu thức.
- Các toán tử được liệt kê theo độ ưu tiên từ cao xuống thấp (các toán tử được mô tả trên cùng một dòng sẽ cùng một độ ưu tiên) và chỉ rõ tính kết hợp:
 - Toán tử **** DSTAR** kết hợp phải
 - Toán tử **. DOT** kết hợp trái
 - Toán tử *** MUL**, **/ DIV**, **% MOD** kết hợp trái
 - Toán tử **+ ADD**, **- SUB** kết hợp phải
 - Toán tử **?? DQUES** không có tính kết hợp
- Để thay đổi được độ ưu tiên và tính kết hợp, người ta có thể sử dụng cặp ngoặc tròn để tạo biểu thức con.

Sử dụng ANTLR4 để mô tả ngôn ngữ nói trên. Biết rằng các từ in đậm và nghiêng là tên các từ vựng trong ngôn ngữ đã đặt, sinh viên trực tiếp sử dụng các tên này.

Ví dụ:

```
abc = 1 + 2 ?? 3;  
u = array( a1 => 3.4, a2 => 3 + (u2 % 5));
```

3.

Trình bày về **thừa kế (inheritance)** trong các ngôn ngữ lập trình hướng đối tượng? Dựa vào khái niệm này, các ngôn ngữ lập trình hướng đối tượng được chia thành bao nhiêu dạng?

Bằng các ngôn ngữ lập trình đã học, hãy cho ví dụ để thấy rõ các vấn đề gặp phải khi thiết kế thừa kế trong các dạng trên.

4.

Yêu cầu: Sinh viên sử dụng lập trình hàm để thực hiện câu hỏi sau đây. Nếu không sử dụng lập trình hàm, sinh viên chỉ nhận tối đa 1.5 điểm cho câu hỏi này.

Cho một đoạn ngữ pháp mô tả ngôn ngữ X như sau:

```
program: funcdecl+ EOF;
funcdecl: FUNC ID paramlist typ SM;
paramlist: param typ (CM param typ)*;
param: REF? ID;
typ: 'int' | 'float';
CM: ','; REF: '&';
FUNC: 'func';
SM: ';';
ID: [a-zA-Z]+;
WS: [ \r\n] -> skip;
```

Mô tả: chương trình dùng để khai báo hàm, trong đó tham số có thể được truyền theo kiểu giá trị/tham chiếu.

Ví dụ: `func add(a int, &b int) int;` là một lệnh phù hợp trong ngôn ngữ đã mô tả, trong đó a (kiểu int), b (kiểu int) là hai tham số của hàm add (có kiểu trả về là int) được truyền lần lượt theo giá trị và tham chiếu.

Cho các lớp AST như sau:

```
class AST(ABC)
class Program(AST) # decl:List[Decl]
class Decl(AST)
class FuncDecl(AST) # id:str, ret:Type, param:ParamDecl = None
class ParamDecl(AST) # id:str, isRef:bool, typ:Type, tail:ParamDecl = None
class Type(AST)
class IntType(Type)
class FloatType(Type)
```

Hãy **viết các phương thức** của lớp ASTGeneration để sinh cây AST cho trường hợp ngôn ngữ X nói trên.

Ví dụ: `func add(a int, &b int) int;`

```
Program(
  FuncDecl(
    "add",
    IntType(),
    ParamDecl("a", False, IntType(),
    ParamDecl("b", True, IntType(), None))
  )
)
```

✓ y olse