

Đã bắt đầu vào lúc Thursday, 28 September 2017, 6:48 AM

Tình trạng Đã hoàn thành

Hoàn thành vào lúc Thursday, 28 September 2017, 7:00 AM

Thời gian thực hiện 11 phút 40 giây

Điểm 7,50 của 10,00 (75%)

Câu hỏi **1**

Chính xác

Điểm 1,00 của 1,00

Cho luật sinh của phát biểu if như sau:

ifstmt \rightarrow IF exp THEN stmt ELSE stmt

trong đó **ifstmt** là ký hiệu không kết thúc đại diện cho phát biểu if; các token **IF**, **THEN**, **ELSE** ứng với các từ khoá if, then, else; **exp** là ký hiệu không kết thúc đại diện cho một biểu thức; **stmt** là ký hiệu không kết thúc đại diện cho một phát biểu.

Nút trên cây phân tích cú pháp (parse tree) sinh bởi ANTLR ứng với ifstmt sẽ là đối tượng của lớp nào?

Chọn một:

- ☒ a. IfstmtContext ✓
- ☐ b. Ifstmt
- ☐ c. ifstmt
- ☐ d. Stmt

Câu trả lời của bạn chính xác.

Câu trả lời đúng là: IfstmtContext

Câu hỏi **2**

Không chính xác

Điểm -0,25 của
1,00

Cho luật sinh của phát biểu if như sau:

$\text{ifstmt} \rightarrow \text{IF exp THEN stmt ELSE stmt}$

trong đó **ifstmt** là ký hiệu không kết thúc đại diện cho một phát biểu if; các token **IF**, **THEN**, **ELSE** ứng với các từ khoá if, then, else; **exp** là ký hiệu không kết thúc đại diện cho một biểu thức; **stmt** là ký hiệu không kết thúc đại diện cho một phát biểu bất kỳ.

Trên AST, nút IFSTMT được dùng để diễn tả cho phát biểu if trên, hỏi nút IFSTMT sẽ có bao nhiêu nút con?

Chọn một:

- ☐ a. 2
- ☐ b. 3
- ☒ c. 6 ✖
- ☐ d. 1

Câu trả lời của bạn không chính xác.

Câu trả lời đúng là: 3

Câu hỏi **3**

Không chính xác

Điểm -0,25 của
1,00

Cho luật sinh của phát biểu for trên ngôn ngữ Pascal được viết ở dạng EBNF như sau:

`forstmt → FOR ID ASSIGN exp (TO | DOWNTO) exp DO stmt`

trong đó, **forstmt** là ký hiệu không kết thúc đại diện cho phát biểu for; các token **FOR**, **TO**, **DOWNTO**, **DO** đại diện cho các từ khoá for, to, downto, do; **ID** đại diện cho một biến; **ASSIGN** cho `:=`; **exp** là ký hiệu không kết thúc đại diện cho biểu thức; **stmt** là ký hiệu không kết thúc đại diện cho phát biểu. Khi sử dụng **TO** thì vòng lặp for sẽ tăng giá trị biến đếm ID lên 1 sau mỗi lần lặp cho đến khi giá trị của ID lớn hơn giá trị của exp; còn khi sử dụng **DOWNTO** thì giá trị của biến đếm ID sẽ giảm đi 1 sau mỗi lần lặp cho đến khi ID nhỏ hơn giá trị của exp.

Trên AST, nút FORSTMT được dùng để diễn tả cho phát biểu for nói trên, hỏi nút FORSTMT sẽ có bao nhiêu nút con?

Chọn một:

- ☐ a. 9
- ☐ b. 4
- ☐ c. 5
- ☒ d. 8 ❌

Câu trả lời của bạn không chính xác.

Câu trả lời đúng là: 5

Câu hỏi **4**

Chính xác

Điểm 1,00 của 1,00

Cho luật sinh của phát biểu for như sau:

`forstmt → LB exp SEMI exp SEMI exp RB stmt`

trong đó, **forstmt** là ký hiệu không kết thúc biểu diễn phát biểu for; các ký hiệu kết thúc **LB**, **RB** và **SEMI** tương ứng với '(', ')' và ';'; **exp** là ký hiệu không kết thúc biểu diễn biểu thức; **stmt** là ký hiệu không kết thúc biểu diễn một phát biểu bất kỳ.

Cho các khai báo cấu trúc dữ liệu của AST hiện có như sau:

`trait Stmt // cấu trúc lưu trữ cho phát biểu`

`trait Exp // cấu trúc lưu trữ cho biểu thức`

Chọn khai báo thích hợp cho cấu trúc dữ liệu của nút ForStmt trên AST biểu diễn cho phát biểu for trên?

Chọn một:

- ☒ a. `case class ForStmt(e1:Exp, e2:Exp, e3:Exp, s:Stmt) extends Stmt` ✓
- ☐ b. `case class ForStmt(e1:Exp, e2:Exp, e3:Exp, s:Stmt) extends Exp`
- ☐ c. `case class ForStmt(e:Exp, s:Stmt) extends Exp`
- ☐ d. `case class ForStmt(e:Exp, s:Stmt) extends Stmt`

Câu trả lời của bạn chính xác.

Câu trả lời đúng là: `case class ForStmt(e1:Exp, e2:Exp, e3:Exp, s:Stmt) extends Stmt`

Câu hỏi **5**

Chính xác

Điểm 1,00 của 1,00

Cho văn phạm có các luật sinh sau:

$\text{program} \rightarrow \text{vardecls}$

$\text{vardecls} \rightarrow \text{vardecl vardecls} \mid \text{vardecl}$

$\text{vardecl} \rightarrow \text{type ids}$

$\text{type} \rightarrow \text{INTTYPE} \mid \text{FLOATTYPE}$

$\text{ids} \rightarrow \text{ID COMMA ids} \mid \text{ID}$

Và AST tương ứng với văn phạm trên được định nghĩa như sau:

trait AST

case class Program(decls:List[VarDecl]) extends AST

case class VarDecl(typ:Type,id:List[String]) extends AST

trait Type extends AST

object IntType extends Type

object FloatType extends Type

Phương thức visitVardecls cần phải trả về đối tượng gì?

Chọn một:

- ☐ a. Program
- ☒ b. List[VarDecl] ✓
- ☐ c. IntType
- ☐ d. VarDecl

Câu trả lời của bạn chính xác.

Câu trả lời đúng là: List[VarDecl]

Câu hỏi **6**

Chính xác

Điểm 1,00 của 1,00

Cho luật sinh của phát biểu if như sau:

`ifstmt` \rightarrow `IF exp THEN stmt ELSE stmt`

trong đó **ifstmt** là ký hiệu không kết thúc đại diện cho phát biểu if; các token **IF**, **THEN**, **ELSE** ứng với các từ khoá if, then, else; **exp** là ký hiệu không kết thúc đại diện cho một biểu thức; **stmt** là ký hiệu không kết thúc đại diện cho một phát biểu.

Giả sử ctx là biến đang cất giữ nút ứng với ifstmt và đang viết cho một visitor để tạo AST, để tạo AST cho nút con stmt sau ELSE thì cần phải viết như thế nào?

Chọn một:

- ☐ a. `ctx.stmt().accept(this)`
- ☒ b. `ctx.stmt(1).accept(this)` ✓
- ☐ c. `ctx.stmt.accept(this)`
- ☐ d. `ctx.stmt(1)`

Câu trả lời của bạn chính xác.

Câu trả lời đúng là: `ctx.stmt(1).accept(this)`

Câu hỏi **7**

Chính xác

Điểm 1,00 của 1,00

Cho văn phạm có các luật sinh sau:

$\text{program} \rightarrow \text{vardecls}$

$\text{vardecls} \rightarrow \text{vardecl vardecls} \mid \text{vardecl}$

$\text{vardecl} \rightarrow \text{type ids}$

$\text{type} \rightarrow \text{INTTYPE} \mid \text{FLOATTYPE}$

$\text{ids} \rightarrow \text{ID COMMA ids} \mid \text{ID}$

Và AST tương ứng với văn phạm trên được định nghĩa như sau:

trait AST

case class Program(decls:List[VarDecl]) extends AST

case class VarDecl(typ:Type,id:List[String]) extends AST

trait Type extends AST

object IntType extends Type

object FloatType extends Type

Đối tượng IntType được tạo ra trong phương thức nào?

Chọn một:

- ☒ a. visitType ✓
- ☐ b. visitVardecl
- ☐ c. visitINTTYPE
- ☐ d. visitIds

Câu trả lời của bạn chính xác.

Câu trả lời đúng là: visitType

Câu hỏi **8**

Chính xác

Điểm 1,00 của 1,00

Cho luật sinh của phát biểu if như sau:

`ifstmt` \rightarrow `IF exp THEN stmt ELSE stmt`

trong đó **ifstmt** là ký hiệu không kết thúc đại diện cho phát biểu if; các token **IF**, **THEN**, **ELSE** ứng với các từ khoá if, then, else; **exp** là ký hiệu không kết thúc đại diện cho một biểu thức; **stmt** là ký hiệu không kết thúc đại diện cho một phát biểu.

Giả sử `ctx` là biến đang cất giữ nút ứng với `ifstmt`, để truy xuất nút con `stmt` sau `THEN` thì cần phải viết như thế nào?

Chọn một:

- ☒ a. `ctx.stmt(0)` ✓
- ☐ b. `ctx.stmt()`
- ☐ c. `ctx.stmt`
- ☐ d. `ctx.StmtContext()`

Câu trả lời của bạn chính xác.

Câu trả lời đúng là: `ctx.stmt(0)`

Câu hỏi **9**

Chính xác

Điểm 1,00 của 1,00

Cho văn phạm có các luật sinh sau:

$\text{vardecls} \rightarrow \text{vardecl} \text{ vardecls} \mid \text{vardecl}$

$\text{vardecl} \rightarrow \text{type ids}$

$\text{type} \rightarrow \text{INTTYPE} \mid \text{FLOATTYPE}$

$\text{ids} \rightarrow \text{ID COMMA ids} \mid \text{ID}$

Một visitor sinh bởi ANTLR cho văn phạm sẽ có phương thức visit nào trong các phương thức dưới đây?

Chọn một:

- ☐ a. visitVardeclContext
- ☒ b. visitIds ✓
- ☐ c. visitID
- ☐ d. visittype

Câu trả lời của bạn chính xác.

Câu trả lời đúng là: visitIds

Câu hỏi **10**

Chính xác

Điểm 1,00 của 1,00

Cho cấu trúc dữ liệu của AST được khai báo như sau:

trait Exp

case class Add(e1:Exp,e2:Exp) extends Exp // cho phép cộng 2 ngôi

case class Minus(e1:Exp,e2:Exp) extends Exp // cho phép trừ 2 ngôi

case class Mul(e1:Exp,e2:Exp) extends Exp // cho phép nhân

case class Lit(i:Int) extends Exp

Hãy viết AST thích hợp của biểu thức $(10 - 20) * 5 + 7$ (độ ưu tiên và kết hợp như thông lệ)?

Qui ước viết:

- Viết đúng cả chữ thường và hoa, đúng thứ tự toán hạng (trái hoặc phải), trình tự tính toán
- Không viết khoảng trắng
- = Không viết dư thừa

Trả lời: ✓

Câu trả lời đúng là: Add(Mul(Minus(Lit(10),Lit(20)),Lit(5)),Lit(7))