


Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

<div></div> <div>TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM</div> <div><u>KHOA KH & KT MÁY TÍNH</u></div>	THI CUỐI KỲ		Học kỳ / Năm học		1	2023-2024
			Ngày thi		28-12-2023	
	Môn học	Nguyên lý ngôn ngữ lập trình				
	Mã môn học	CO3005				
	Thời lượng	100 phút	Mã đề	2310		
Ghi chú: -Không được phép dùng tài liệu - Sinh viên thực hiện phần Trắc nghiệm trên giấy làm bài trắc nghiệm. - Sinh viên thực hiện phần Lập trình trên giấy làm bài. - Sinh viên nộp lại đề sau khi kiểm tra.						

PHẦN I: LÝ THUYẾT (6.0 điểm) (L.O.2)

Câu 1. Cho đoạn mã trong Python như sau:

```

1 | def outer_function():
2 |     x = 10
3 |     def inner_function():
4 |         nonlocal x
5 |         x += 5
6 |
7 |         y = x * 2
8 |         return y
9 |     res = inner_function()
10 |    return res
11 | result = outer_function()

```

Giá trị của biến `result` là

- ☐ (A) 20
 ☐ (B) 25
 ☐ (C) 30
 ☐ (D) Error: 'nonlocal' cannot be used with 'x' in inner_function
 ☐ (E) Một giá trị khác

Câu 2. Cho đoạn mã trong Python như sau:

```

1 | x = 10
2 | def outer_function():
3 |     def inner_function():
4 |         y = x + 3
5 |         def nested_function():
6 |             global x
7 |             x += y
8 |
9 |             return x - 2
10 |            return nested_function()
11 |        global x
12 |        x = 5
13 |        res = inner_function()
14 |        return res
15 | result = outer_function()

```

Giá trị của biến `result` là

- ☐ (A) 21
 ☐ (B) 11
 ☐ (C) 20
 ☐ (D) Error: 'global' cannot be used with 'x' after its first assignment in nested_function
 ☐ (E) Một giá trị khác

Đoạn mô tả sau được áp dụng cho các câu 3–6:

Cho đoạn chương trình được viết trên ngôn ngữ cấu trúc khối tựa C với việc cho phép các hàm lồng nhau.

```

1 | int x;
2 | void sub1(int y) {
3 |     int z = 1;
4 |     int sub2(int x)
5 |     { return x * z + y; }
6 |
7 |     z = 3;
8 |     x = sub2(z) - sub2(z + 1);
9 | }
10 | void main() {
11 |     x = 1;

```

```
11 | sub1(x);
12 | cout << x;
```

```
13 | }
```

Câu 3. Có bao nhiêu khai báo cục bộ trong môi trường tham khảo của sub1

- (A) 1 (B) 2 (C) 3 (D) 4

Câu 4. Các đối tượng trong bản hoạt động của sub1 khi thực thi là:

- (A) x - y - z - sub2 (B) x - y - z
(C) y - z - sub2 (D) x - sub1 - y - z - sub2

Câu 5. Tầm vực của biến x khai báo ở dòng đầu tiên là:

- (A) main, sub1, sub2 (B) main (C) sub1 (D) main, sub1

Câu 6. Giá trị của x được in ra khi thực thi hàm main là:

- (A) 1 (B) -3 (C) -1 (D) Một giá trị khác

Câu 7. Đặc điểm nào sau đây là đặc điểm của chuỗi có độ dài cố định?

- (A) Kích thước của chuỗi được xác định tại thời điểm biên dịch và giữ nguyên.
(B) Độ dài của chuỗi có thể thay đổi trong quá trình chạy chương trình.
(C) Chuỗi có độ dài cố định sử dụng bộ nhớ hiệu quả hơn so với chuỗi có độ dài biến đổi.
(D) Chúng có thể lưu trữ bất kỳ số ký tự nào mà không hạn chế.
(E) Bộ nhớ cho chuỗi có độ dài cố định được cấp phát động.

Câu 8. Trong biểu diễn 32-bit theo chuẩn IEEE-754 (phần mũ là 8 bit) của số thực 1.78, chuỗi 4 bit (theo thứ tự) nào sau đây chỉ xuất hiện duy nhất 1 lần:

- (A) 1110 (B) 1011 (C) 1100 (D) 1000

Câu 9. Cho một hàm được định nghĩa như sau trên một ngôn ngữ dùng suy diễn kiểu có cú pháp tựa Python:

```
def foo(x, y, z): return reduce(x(z), y, [])
```

Giả sử trên ngôn ngữ này, `reduce` có kiểu $(List[T] \times T \rightarrow List[T]) \times List[T] \times List[T] \rightarrow List[T]$ (nhắc lại trong biểu thức kiểu, phép \rightarrow có độ ưu tiên thấp hơn \times) và kiểu của một hằng số nguyên là `int`. Hãy cho biết kiểu của biến `a` là gì để lệnh gọi `foo(a, [2, 3, 4, 5], 4)` không bị sai kiểu?

- (A) $List[int] \times int \rightarrow List[int]$ (B) $int \rightarrow (List[int] \times int \rightarrow List[int])$
(C) $(int \times List[int]) \times int \rightarrow List[int]$ (D) $int \rightarrow (List[T] \times T \rightarrow List[T])$

Câu 10. Cho một hàm được định nghĩa như sau trên một ngôn ngữ dùng suy diễn kiểu có cú pháp tựa Python:

```
def foo(x, y): return y(y(x))
```

Giả định rằng, các ký hiệu bắt đầu bằng ký tự T đều là các biến kiểu. Biểu thức kiểu biểu diễn kiểu của hàm `foo` sử dụng ít biến kiểu nhất là:

- (A) $(T_1 \times (T_1 \rightarrow T_2)) \rightarrow T_2$ (B) $(T_1 \times (T_1 \rightarrow T_1)) \rightarrow T_1$
(C) $(T_1 \times (T_2 \rightarrow T_1)) \rightarrow T_1$ (D) $(T_1 \times (T_1 \rightarrow T_2)) \rightarrow T_3$

Mô tả sau được áp dụng cho các câu 11–12:

Cho một struct được khai báo trên C như sau:

```
1 | struct X {
2 |     char a;
3 |     int b;
4 |     char c;
5 |     float d;
```

```
6 | union {
7 |     char e[3];
8 |     float f;
9 | } g;
```

Cho biết kích thước của các kiểu dữ liệu `char`, `int` và `float` tương ứng là 1, 2 và 4.

Câu 11. Hãy xác định kích thước của struct khi áp dụng alignment?

- (A) 11 (B) 16 (C) 12 (D) Một giá trị khác

Câu 12. Hãy chọn một trình tự khai báo các thành phần của struct trên để có kích thước nhỏ nhất khi áp dụng alignment?

- (A) a-b-c-d-g (B) b-d-a-c-g (C) g-b-a-d-c (D) a-c-b-d-g

Câu 13. Cho các phát biểu sau:

- (a) Phải khởi động trị cho biến kiểu tham khảo ngay khi khai báo
 - (b) Có thể buộc biến kiểu tham khảo chuyển sang tham khảo biến khác
 - (c) Không có tác vụ dereference đối với biến kiểu tham khảo
 - (d) Kiểu tham khảo cũng có các tác vụ số học như kiểu con trỏ
- Có bao nhiêu phát biểu Đúng trong các phát biểu trên:

- (A) 4 (B) 3 (C) 2 (D) 1

Câu 14. Đoạn mã C++ nào sau đây có thể được dùng làm ví dụ cho khái niệm Ad hoc polymorphism?

- (A)

```
1 class A {
2     int foo(int a);
3     char foo(char a, char b);
4 }
```
- (B)

```
1 template<typename T> T swap(T &a, T
    &b);
```
- (C)

```
1 class A{ ... }
2 class B:A {...}
```
- (D)

```
1 class A{ int foo(int a); }
2 class B::A { override int foo(int
    a); }
```

Câu 15. Hãy cho biết ngôn ngữ Python áp dụng quy tắc tầm vực và kiểm tra kiểu loại gì?

- (A) tầm vực tĩnh-không kiểm tra kiểu (B) tầm vực động-kiểm tra kiểu động
- (C) tầm vực tĩnh-kiểm tra kiểu động (D) tầm vực động-kiểm tra kiểu tĩnh
- (E) tầm vực tĩnh-kiểm tra kiểu tĩnh

Câu 16. Phát biểu nào sau đây là đúng khi nói về tính kết hợp của toán tử?

- (A) Kết hợp của toán tử đề cập đến thứ tự mà các toán tử có độ ưu tiên giống nhau được thực hiện trong một biểu thức. Nó có thể là trái sang phải hoặc phải sang trái.
- (B) Kết hợp của toán tử là một thuật ngữ được sử dụng để mô tả cách các toán tử tương tác với các toán hạng trong một biểu thức. Nó đảm bảo rằng các phép toán được thực hiện theo một thứ tự cụ thể, tránh sự mơ hồ.
- (C) Kết hợp của toán tử không liên quan đến ngôn ngữ lập trình; tất cả các toán tử đều được thực hiện đồng thời trong một biểu thức.
- (D) Kết hợp của toán tử là một tính năng cho phép trình biên dịch thay đổi thứ tự của các toán tử động trong thời gian chạy.

Câu 17. Một chương trình trong Python đang thực hiện có ba biến a, b, c lần lượt đang mang giá trị 3, 2, 5. Cho biết giá trị của result sau khi thực hiện phát biểu sau?

```
result = (a * b) + (c - b) if a < b else (c / b) - (a + b)
```

- (A) 3.5 (B) 9 (C) -2.5 (D) 8.5
- (E) Một giá trị khác

Câu 18. Cho đoạn mã sau viết trên C++:

```
1 || if (p != NULL && p->val != 5) p->val = 5;
```

Chọn đoạn mã tương đương trong trường hợp C++ không thực hiện rút ngắn tính toán (short-circuit evaluation) trên biểu thức luận lý:

(A)

```
1 || bool c1 = p != NULL;  
2 || bool c2 = p->val != 5;  
3 || if (c1 && c2) p->val = 5;
```

(B)

```
1 || if (p != NULL) p->val = 5;  
2 || else if (p->val != 5) p->val = 5;
```

(C)

```
1 || if (p != NULL)  
2 ||     if (p->val != 5) p->val = 5;
```

(D)

```
1 || bool check = false;  
2 || if (p != NULL) check = true;  
3 || if (p->val != 5) check = true;  
4 || if (check) p->val = 5;
```

Câu 19. Trong các biểu thức C++ sau, biểu thức nào có thể có các giá trị khác nhau?

(a) $a + (b = 4)$

(b) $a + (a = 4)$

(c) $a + b * c$

(A) Chỉ (b)

(B) Chỉ (a)

(C) (a) và (b)

(D) Cả 3 biểu thức

(E) Không có biểu thức nào

Câu 20. Cho một chương trình trong Python3:

```
1 || def foo(lst, target):  
2 ||     for item in lst:  
3 ||         if item == target:  
4 ||             print(f"{target} found!")  
5 ||             break  
6 ||     else:  
7 ||         print(f"{target} not found!")  
8 ||  
9 || # Consider the following function call:  
10 || lst = [1, 2, 3, 4, 5]  
11 || foo(lst, 3)
```

Sau khi thực hiện xong, giá trị nào được in ra màn hình:

(A) 3 found!

(B) 3 not found!

(C) 3 found! 3 not found!

(D) Không có gì được in ra

Câu 21. Lý do nào sau đây làm cho vấn đề dangling-else không xuất hiện trong Python?

(A) Python không hỗ trợ cú pháp "else" trong các câu lệnh điều kiện.

(B) Khối lệnh trong Python được xác định bởi thụt đầu dòng để xác định phạm vi

(C) Python tự động thụt đầu dòng vào mỗi câu lệnh điều kiện để tránh vấn đề "dangling else".

(D) Python sử dụng từ khóa "elif" thay vì "else" để loại bỏ khả năng xuất hiện "dangling else"

Câu 22. Cho đoạn mã sau viết trên Python:

```
1 || n = 10  
2 || for i in range(0, n):  
3 ||     i = i - 1  
4 ||     print(i)
```

Hỏi số lần lặp của các phát biểu trong thân vòng for trên là:

(A) 1 lần

(B) 9 lần

(C) 10 lần

(D) vô hạn

Mô tả sau dùng cho các câu hỏi 23–24:

Cho một chương trình được viết trong ngôn ngữ có cú pháp tựa Pascal:

```

1 function phil(int a, int b, int c)
2 begin
3     b := b + 5;
4     b := a + c + 4;
5     print a, b, c;
6 end
7

```

```

8 function main
9 begin
10     int j := 10;
11     int k := 15;
12     phil(j, j, j + k);
13     print j, k;
14 end

```

Biết rằng hàm `main` được thực thi đầu tiên khi chương trình khởi chạy.

Câu 23. Hãy cho biết các giá trị lần lượt được in ra khi trong hàm `phil` có tham số `a`, `b` được truyền bằng tham khảo (pass by reference) và `c` được truyền bằng giá trị (pass by value)?

- (A) 10 44 25 10 15 (B) 44 44 25 10 15 (C) 44 39 25 44 15 (D) 44 44 25 44 15

Câu 24. Hãy cho biết các giá trị lần lượt được in ra khi trong hàm `phil` có tham số `a`, `b` được truyền bằng giá trị-kết quả (pass by value-result) và `c` được truyền bằng giá trị (pass by value)?

- (A) 10 39 25 39 15 (B) 44 44 25 44 15 (C) 44 39 25 44 15 (D) 44 44 25 10 15

Câu 25. Cho hai phát biểu sau:

- (a) khi truyền tham khảo (pass-by-reference), thông số thực được tính toán trước khi truyền
(b) khi truyền bằng tên (pass-by-name), thông số thực được truyền mà không cần tính toán

- (A) Chỉ phát biểu (a) đúng (B) Chỉ phát biểu (b) đúng (C) Cả 2 phát biểu đều sai
(D) Cả 2 phát biểu đều đúng

Câu 26. Generator của Python có thể được xem là minh họa cho cơ chế gọi chương trình con nào?

- (A) Gọi-Trở về đơn giản (B) Đệ quy (C) Biến cố (D) Song hành

Câu 27. Cho `a` là biến nguyên có chỉ số 2 và `b` là biến mảng các số nguyên có chỉ số 5 trong dãy các biến cục bộ (local variables array). Biểu thức C++ nào có thể được dịch thành chuỗi các lệnh Java bytecode sau:

```

1 aload 5
2 aload 5
3 iconst_5
4 iaload

```

```

1 iload_2
2 bipush 10
3 imul
4 iastore

```

- (A) `b[b[5]] = a * 10;` (B) `b[5] = a * 10;`
(C) `a = b[5] * 10;` (D) `a = b[b[5]] * 10;`

Câu 28. Trong các lệnh Java bytecode sau, có bao nhiêu lệnh sau khi thực thi sẽ làm kích thước operand stack giảm đi 2 so với trước khi thực thi?

`bipush`, `baload`, `iadd`, `iconst_1`, `fastore`, `fstore_3`, `goto`, `if_icmpne`.

- (A) 1 (B) 0 (C) 2 (D) 3

Đoạn mô tả sau áp dụng cho các câu 29–30:

Cho lớp `DoWhile` (tương ứng phát biểu do-while của C) trong các lớp biểu diễn AST được mô tả như sau:

```

1 class DoWhile(Stmt): #body:List[Stmt],expr:Expr

```

Hãy điền vào các chỗ trống trong đoạn mã Python sau để sinh mã cho phát biểu do-while của C:

```

1 def visitDoWhile(self,ctx:DoWhile,o:Object):
2     o.frame.enterLoop()
3     cntLabel,brkLabel=o.frame.getContinueLabel(),o.frame.getBreakLabel()
4     label1 = o.frame.getNewLabel()
5     self.emit.printout(self.emit.emitLABEL(label1,o.frame))
6     [self.visit(x,o) for x in ctx.body]
7     self.emit.printout(_____)
8     self.emit.printout(self.visit(ctx.expr,Access(o.frame,o.sym,False,True))[0])
9     self.emit.printout(self.emit._____.label1,o.frame))
10    self.emit.printout(self.emit.emitLABEL(brkLabel,o.frame))
11    o.frame.exitLoop()

```

Câu 29. Chỗ trống ở dòng 7:

(A) self.emit.emitLABEL(label1,o.frame)

(B) self.emit.emitLABEL(bkrLabel1,o.frame)

(C) Không cần mã nào ở vị trí này

(D) self.emit.emitLABEL(cntLabel,o.frame)

Câu 30. Chỗ trống ở dòng 9:

(A) emitGOTO(

(B) emitIFFALSE(

(C) emitIFTRUE(

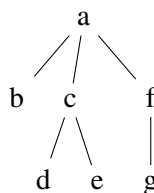
(D) emitRELOP(">",

PHẦN II: LẬP TRÌNH (cũng dùng để tính điểm phần B của BTL 2 và 3) (4.0 điểm)

Câu 31. Cho một đoạn ngữ pháp của ngôn ngữ U để mô tả một cấu trúc cây (tree) được viết trong ANTLR4 như sau:

```
tree: node EOF;
node: LS info node* RS;
info: NAME (DOT COLOR)?;
NAME: [a-zA-Z]+; COLOR: [RGB]; LS: '['; RS: ']'; DOT: '.';
```

Ví dụ: [a.R [b] [c.G [d.B] [e]] [f.R [g.Y]]] để mô tả một cây tương ứng có hình vẽ sau:



Cho các lớp AST:

```
1 class Tree(ABC)
2 class MultiwayTree(Tree)
3 # name: str, color: str, first: Tree, middles: List[Tree], last: Tree
4 class BinaryTree(Tree) # name: str, color: str, left: Tree, right: Tree
5 class Leaf(Tree) # name: str, color: str
```

Hãy viết lớp ASTGeneration (lớp con của Visitor sinh ra bởi ANTLR) để sinh cây cú pháp từ biểu tượng với đoạn ngữ pháp nêu trên theo qui tắc sau:

- MultiwayTree dùng cho node có từ 3 node con trở lên. Các thành phần ngoại trừ color phải khác None.
- BinaryTree dùng cho node có từ 1 đến 2 node. Khi có 1 node con thì thành phần right là None.
- Leaf dành cho node không có node con.

Với cây cho trong ví dụ, thì cây cú pháp từ biểu tượng được sinh ra là:

```
1 MultiwayTree('a', 'R',
2     Leaf('b', ''),
3     [BinaryTree('c', 'G', Leaf('d', 'B'), Leaf('e', ''))],
4     BinaryTree('f', 'R', Leaf('g', 'Y'), None))
```

Câu 32. (L.O.3) Giả sử một chương trình trên ngôn ngữ X chỉ gồm có các khai báo và các biểu thức. Có 2 loại khai báo trên ngôn ngữ này: khai báo biến và khai báo lớp. Khai báo biến gồm danh hiệu và kiểu (kiểu nguyên và kiểu lớp). Một khai báo lớp gồm tên lớp, tên lớp cha (None nếu không có lớp cha) và một danh sách các khai báo thành phần trong đó mỗi thành phần tương tự khai báo biến (tức gồm danh hiệu - tên thành phần và kiểu). Các biểu thức chỉ có hai loại là biến và truy xuất thành phần của lớp. Các lớp của AST mô tả chương trình trên ngôn ngữ X gồm:

```
1 class AST(ABC)
2 class Program(AST) #decl:List[Decl];exps:List[Exp]
3 class Decl(AST)
4 class ClassDecl(Decl) #name:str;parent:str;mem:List[VDecl]
5 class VDecl(Decl): #name:str;typ:Type
6 class Exp(ABC)
```

```

7 | class Id(Exp) #name:str
8 | class FieldAccess(Exp) #exp:Exp;field:str
9 | class Type(ABC)
10 | class IntType(Type)
11 | class ClassType(Type) #name:str

```

trong đó các class AST, Decl, Exp, Type là các lớp trừu tượng, các lớp còn lại là các lớp cụ thể và có các thành phần được thể hiện bởi các chú thích kể bên tên lớp.

Ví dụ:

```

1 | Program(
2 |   [VDecl("a", IntType()),
3 |     VDecl("b", ClassType("c")),
4 |     ClassDecl("c", None,
5 |       [ VDecl("a", IntType()), VDecl("b", ClassType("c")) ] )
6 |   ],
7 |   [ Id("a"),
8 |     FieldAccess(FieldAccess(Id("b"), "b"), "a")
9 |   ])

```

thể hiện chương trình có 2 khai báo biến a (kiểu nguyên) và biến b (kiểu lớp c) cùng với một khai báo lớp c (không có lớp cha và có 2 thành phần a kiểu nguyên và thành phần b kiểu lớp c). Chương trình có 2 biểu thức: biến a và truy xuất thành phần a của thành phần b của biến b.

Cho visitor để thao tác trên các lớp của AST trên và các lớp để báo lỗi như sau:

<pre> 1 class BaseVisitor(Visitor, Utils): 2 def visitProgram(self, ast, c) 3 def visitClassDecl(self, ast, c) 4 def visitVDecl(self, ast, c) 5 def visitId(self, ast, c) 6 def visitFieldAccess(self, ast, c) 7 def visitIntType(self, ast, c) 8 def visitClassType(self, ast, c) </pre>	<pre> 1 class Exc(Exception) 2 class Redeclared(Exc) #kind:Kind;name:str 3 class Undeclared(Exc) #kind:Kind;name:str 4 class TypeMismatchInExp(Exc) #exp:Exp 5 class Kind(ABC) 6 class Variable(Kind) 7 class Field(Kind) 8 class Class(Kind) </pre>
---	--

Hãy viết một (hoặc nhiều) lớp con của lớp BaseVisitor để kiểm tra các ràng buộc sau trên một chương trình của ngôn ngữ X:

- Kiểm tra các biến trùng tên hoặc các lớp trùng tên hoặc các thành phần của cùng một lớp có trùng tên. Nếu có thì phải ném ra Redeclared(Variable(), <tên biến trùng tên>)- đối với biến trùng tên hoặc Redeclared(Field(), <tên thành phần trùng tên>)-đối với thành phần trùng tên hoặc Redeclared(Class(), <tên lớp trùng tên>)- đối với lớp trùng tên. Hai thành phần trùng tên nhưng thuộc các lớp khác nhau không gây ra lỗi trùng tên. Một biến với một lớp trùng tên không gây ra lỗi trùng tên.
- Trong một FieldAccess thì kiểu của thành phần exp phải là ClassType và field phải có trong thành phần hoặc được thừa kế từ các lớp cha/tổ tiên của lớp tương ứng, nếu không thì ném ra TypeMismatchInExp(<biểu thức>). Ví dụ nếu có các biểu thức FiledAccess(Id("a"), "a") hoặc FieldAccess(Id("b"), "c") thì phải ném ra lỗi này vì biến a không có kiểu ClassType, còn biến b có kiểu ClassType nhưng lớp này (không có lớp cha) không có thành phần nào có tên là c.

Có thể hiện thực các câu trên trong cùng 1 đoạn code nhưng phải ghi tiêu đề là các câu đã thực hiện (ví dụ a,b). Môi trường tham khảo dùng trong câu (b) phải được tạo ra trong câu (a). Được phép sử dụng hàm lookup của lớp Utils và lớp Symbol đã được cho trong bài tập lớn 3.

HẾT

PHẦN I: LÝ THUYẾT (6.0 điểm) (L.O.2)

Câu 1. C

Câu 2. B

Câu 3. C

Câu 4. C

Câu 5. D

Câu 6. B

Câu 7. A

Câu 8. B

Câu 9. B

Câu 10. B

Câu 11. B

Câu 12. D

Câu 13. C

Câu 14. A

Câu 15. C

Câu 16. A

Câu 17. C

Câu 18. C

Câu 19. A

Câu 20. A

Câu 21. B

Câu 22. C

Câu 23. D

Câu 24. A

Câu 25. D

Câu 26. D

Câu 27. A

Câu 28. A

Câu 29. D

Câu 30. C

PHẦN II: LẬP TRÌNH (cũng dùng để tính điểm phần B của BTL 2 và 3) (4.0 điểm)

Câu 31. Answer: Hướng dẫn chấm:

Câu 32. Answer: (a) (1 điểm)

- Đảm bảo việc duyệt các node cần thiết (Program,Decl,StructType) (0.25)
- Đảm bảo việc truyền đúng biến môi trường để kiểm tra việc trùng tên (0.25)
- Đảm bảo việc kiểm tra trùng tên đúng (0.25)
- Đảm bảo ném đúng biến cố và thông số (Variable, Field) (0.25)

(b) (0.5 điểm)

- Đảm bảo việc duyệt các node cần thiết (Program, StructAccess, Id) và truyền đúng môi trường tham khảo (0.25)

- Đảm bảo việc kiểm tra đúng và ném đúng biến cố cùng thông số (0.25)

(c) (0.5 điểm)

- Đảm bảo việc duyệt các node cần thiết (Program, StructAccess, Id) và truyền đúng môi trường tham khảo (0.1)

- Đảm bảo việc kiểm tra đúng và ném đúng biến cố cùng thông số (0.4)