

Lazy Evaluation

Bài tập 1

Cho một khai báo hàm như sau:

```
def foo(x:Boolean,y:Int,z:Int) = if (x) y else z
```

Hỏi khai báo trên sẽ gây ra lỗi gì?

- ☐ Không gây ra lỗi gì
- ☐ Lỗi biên dịch vì không khai báo kiểu trả về
- ☐ Có thể gây ra lỗi khi thực thi tùy theo cách gọi hàm
- ☐ Luôn gây ra lỗi runtime

Đúng nhưng chưa đủ. Khai báo hàm như trên không gây ra lỗi từ vựng, cú pháp hay ngữ nghĩa. Tuy nhiên, khi gọi hàm như là: `foo(a==0,1,b/a)` thì các biểu thức `a==0`, `1`, `b/a` sẽ được tính để có giá trị truyền cho các thông số hình thức `a,b,c` nên sẽ gây ra lỗi chia cho 0 khi `a` bằng 0.

Chưa đúng, vì Scala có thể suy diễn được kiểu trả về là `Int` do cả 2 vế của `if` đều trả về `Int`

Đúng, tùy theo cách gọi hàm, có thể xảy ra lỗi khi thực thi. Ví dụ `foo(a==0,1,b/a)` sẽ gây ra lỗi chia cho 0 khi `a` bằng 0. Hoặc `foo(x == null,null,x.m())` sẽ gây ra lỗi Null pointer reference.

Chưa đúng, ví dụ `foo(a==0,1,2)` sẽ không gây ra lỗi.

Solution

1. Wrong
2. Wrong
3. Correct Option
4. Wrong

Bài tập 2

Để khắc phục lỗi về ngữ dụng trong bài tập 1, cần phải sửa khai báo hàm như thế nào?

- ☐ `def foo(x:Boolean)(y:Int)(z:Int) = if (x) y else z`
- ☐ `def foo(x:Boolean,y:Int) = (z:Int) => if (x) y else z`

- ☐ `def foo(x:Boolean, y: => Int, z: => Int) = if (x) y else z`
- ☐ `def foo(x:Boolean,y: Int, z:Int) = {
 lazy val m = y
 lazy val n = z
 if (x) m else n
}`

Chưa đúng, khai báo hàm này sẽ cho phép gọi hàm mà không cần đủ các thông số (currying functions) nhưng vẫn gây ra lỗi thực thi như đã nêu ở bài tập 1

Chưa đúng, cách khai báo này là một hình thức khác của khai báo dạng curry, cho phép gọi hàm với 2 thông số nhưng vẫn gây ra lỗi như đã nêu ở bài tập 1

Đúng, khi viết `=>` trước kiểu của thông số, thì thông số sẽ được truyền theo tên (pass-by-name). Khi đó các biểu thức thông số chỉ được tính khi thông số hình thức được sử dụng trong thân hàm.

Chưa đúng, mặc dù dung các biến `m,n` là lazy nhưng do các thông số `y, z` là eager nên khi gọi hàm `foo(a==0,1,b/a)` thì các thông số `a==0, 1, b/a` được tính trước khi truyền nên vẫn gây ra lỗi chia cho 0

Solution

1. Wrong
 2. Wrong
 3. Correct Option
 4. Wrong
-