

Cho các lớp của AST được khai báo như sau:

```
class AST(ABC): pass

class Decl(AST): pass
class Type(AST): pass

class TypeDecl(Decl): # name: str, rhs: Type
    def __init__(self, name, rhs): self.name = name; self.rhs = rhs
    def accept(self, v, o): return v.visitTypeDecl(self, o)

class VarDecl(Decl): # name: str, typ: Type
    def __init__(self, name, typ): self.name = name; self.typ = typ
    def accept(self, v, o): return v.visitVarDecl(self, o)

class Block(Decl): # ele: List[Decl]
    def __init__(self, ele): self.ele = ele
    def accept(self, v, o): return v.visitBlock(self, o)

class IntType(Type):
    def accept(self, v, o): return v.visitIntType(self, o)

class FloatType(Type):
    def accept(self, v, o): return v.visitFloatType(self, o)

class Id(Type): # name: str
    def __init__(self, name): self.name = name
    def accept(self, v, o): return v.visitId(self, o)

class Visitor(ABC):
    def visit(self, ctx, o): return ctx.accept(self, o)

class UndeclaredType(Exception):
    def __init__(self, name): self.name = name
    def __str__(self): return "Undeclared type: " + str(self.name)

class Symbol:
    def __init__(self, name, typ): self.name = name; self.typ = typ
```

Yêu cầu: Viết lớp `StaticCheck` để kiểm tra xem tên kiểu có được khai báo hợp lệ hay không. Một tên kiểu chỉ hợp lệ nếu nó đã được khai báo bởi `TypeDecl` trong cùng `Block` hoặc một `Block` bao bên ngoài. Nếu không thỏa, ném lỗi `UndeclaredType`.

Ví dụ:

```
Block([
    VarDecl("a", IntType()),
    TypeDecl("vd", FloatType()),
    Block([
        VarDecl("b", FloatType()),
        VarDecl("c", Id("vd")) # Hợp lệ
    ]),
    Block([
        VarDecl("d", IntType()),
        TypeDecl("vd1", IntType())
    ]),
    VarDecl("e", Id("vd")) # Hợp lệ
    # Nếu dùng Id("vd1") ở đây thì sai
])
```

và lớp `StaticChecker` được hiện thực một phần như sau:

```

class StaticCheck(Visitor):
    def visitBlock(self, ast: Block, o):
        def reducer(acc, decl):
            res = self.visit(decl, acc)
            return acc if res is None else ____(1)___

        reduce(
            reducer,
            ast.ele,
            ____(2)___
        )

    def visitTypeDecl(self, ast: TypeDecl, o):
        return [__(3)___] + o[1:]

    def visitVarDecl(self, ast: VarDecl, o):
        if type(ast.typ) is ____(4)___:
            self.visit(ast.typ, o)
        return None

    def visitId(self, ast: Id, o):
        if next(
            filter(
                lambda sym: ast.name == sym.name,
                [sym for scope in o for sym in scope]
            ),
            ____(5)___
        ):
            return
        raise UndeclaredType(ast.name)

```

Câu hỏi 1

Chưa trả lời

Đạt điểm 1,00

Trong đoạn mã `visitId`, mục đích của dòng lệnh sau là gì?

```
if next(filter(lambda sym: ast.name == sym.name, [sym for scope in o for sym in scope]), _____):
```

Select one:

- ☐ a. Kiểm tra xem `ast.name` có được khai báo ở block hiện tại hay không
- ☒ b. Kiểm tra xem `ast.name` có phải là tên biến hợp lệ không
- ☐ c. Tìm trong tất cả các kiểu đã khai báo và trả về kiểu phù hợp với `ast.name`
- ☐ d. Kiểm tra xem tên kiểu `ast.name` có được khai báo ở bất kỳ block nào bên ngoài hay không

[Clear my choice](#)



Câu hỏi 2

Chưa trả lời

Đạt điểm 1,00

Trong `visitBlock`, tại sao lại truyền vào `[[]] + o` làm tham số `o` mới khi gọi `reduce`?

Select one:

- ☐ a. Để xóa tất cả các kiểu đã khai báo trước đó
- ☒ b. Để thêm một scope mới (một block mới) vào môi trường tham khảo
- ☐ c. Để gom tất cả các scope lại thành một list duy nhất
- ☐ d. Để tạo ra một bản sao của `o` tránh sửa đổi trực tiếp

[Clear my choice](#)

Câu hỏi 3

Chưa trả lời

Đạt điểm 1,00

Điền vào chỗ trống trong hàm `visitTypeDecl` để thêm kiểu mới vào block hiện tại:

```
def visitTypeDecl(self, ast: TypeDecl, o):  
    return [_____] + o[1:]
```

Select one:

- ☐ a. `o[1:] + [Symbol(ast.name, ast.rhs)]`
- ☒ b. `o[0] + [Symbol(ast.name, ast.rhs)]`
- ☐ c. `[Symbol(ast.name, self.visit(ast.rhs))] + o[0]`
- ☐ d. `o + [Symbol(ast.name, ast.rhs)]`

[Clear my choice](#)

Câu hỏi 4

Chưa trả lời

Đạt điểm 1,00

Trong trường hợp khai báo `VarDecl("e", Id("vd1"))` xuất hiện sau `TypeDecl("vd1", IntType())` ở một block con trước đó, thì điều gì sẽ xảy ra?

Select one:

- ☒ a. Có lỗi vì vd1 không thuộc cùng block hoặc block cha
- ☐ b. Không có lỗi vì vd1 đã được khai báo
- ☐ c. Không có lỗi vì tên không quan trọng, chỉ cần đúng kiểu
- ☐ d. Có lỗi vì vd1 không phải là kiểu hợp lệ

[Clear my choice](#)

Câu hỏi 5

Chưa trả lời

Đạt điểm 1,00

Trong `visitVarDecl`, việc gọi `self.visit(ast.typ, o)` là cần thiết khi:

`if type(ast.typ) is _____:`

Select one:

- ☐ a. Symbol
- ☐ b. IntType
- ☒ c. Id
- ☐ d. FloatType

[Clear my choice](#)

Câu hỏi 6

Chưa trả lời

Đạt điểm 1,00

Điền vào chỗ trống để đảm bảo môi trường mới được khởi tạo chính xác khi bắt đầu xử lý một **Block**:

```
reduce(  
    reducer,  
    ast.ele,  
    _____  
)
```

Select one:

- ☐ a. `[] + o`
- ☒ b. `[] if o is None else [] + o`
- ☐ c. `[o]`
- ☐ d. `o + []`

[Clear my choice](#)

Câu hỏi 7

Chưa trả lời

Đạt điểm 1,00

Hoàn chỉnh phần kiểm tra trong `visitId` để đảm bảo chỉ ném lỗi khi tên không có trong bất kỳ scope nào:

```
if next(  
    filter(lambda sym: ast.name == sym.name, [sym for scope in o for sym in scope]),  
    _____  
):
```

Select one:

- ☐ a. `[]`
- ☐ b. `0`
- ☐ c. `False`
- ☒ d. `None`

[Clear my choice](#)

Câu hỏi 8

Chưa trả lời

Đạt điểm 1,00

Trong `visitBlock`, mỗi lần `visit(decl, acc)` trả về kết quả khác `None`, ta gán lại `acc` bằng:

```
def reducer(acc, decl):  
    res = self.visit(decl, acc)  
    return acc if res is None else _____
```

Select one:

- ☐ a. `decl`
- ☒ b. `res`
- ☐ c. `[]`
- ☐ d. `o`

[Clear my choice](#)

