



ALICE

*Say mê và
sáng tạo*



Giáo viên hướng dẫn

❖ Trần Phúc Minh Khoa

❖ Nguyễn Võ Phi Long

Chương trình con

1. Khái niệm chương trình con

KHÁI NIỆM: Chương trình con là một dãy lệnh mô tả một số thao tác nhất định và có thể được thực hiện (được gọi) từ nhiều vị trí trong chương trình.

Ý NGHĨA: giải quyết các bài toán lớn và phức tạp thành các bài toán nhỏ hơn và đơn giản hơn.

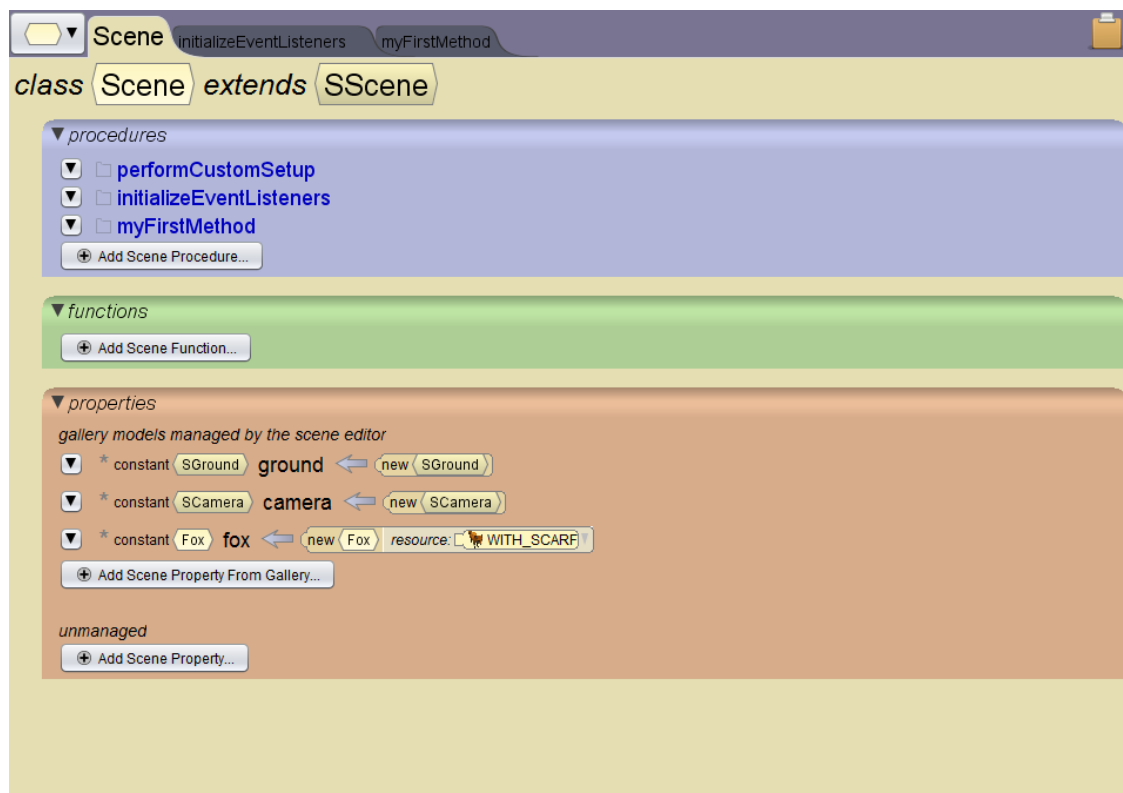
LỢI ÍCH:

- Tránh được sự lặp lại cùng một dãy lệnh. Khi cần dùng có thể gọi lại chương trình con đó.
- Hỗ trợ việc thực hiện các chương trình lớn và phức tạp.
- Mở rộng khả năng ngôn ngữ lập trình thành thư viện cho nhiều người dùng.
- Thuận tiện cho phát triển và nâng cấp chương trình.

2. Khai báo chương trình con trong Alice:

Nhắc lại bài cũ: Khi tạo thành công một project sẽ xuất hiện sẵn 3 tab: Tab Scene (Hay Lớp Scene) sẽ có màu vàng (và mặc định tab Lớp sẽ có màu vàng) và 2 tab: Tab initializeEventListeners, Tab myFirstMethod (Đây là 2 thủ tục (procedure) của Lớp Scene (và mặc định các tab thủ tục hay hàm sẽ có màu tím)

Khi nhấn vào tab Scene (Hình 1)



Hình 1: Tab Scene (Thẻ Scene)

Sẽ có 3 khung (Đối với lớp khác khi mở cũng sẽ có 3 khung tương tự như vậy): Procedures (Thủ tục), Functions (Hàm), properties (Thuộc tính)

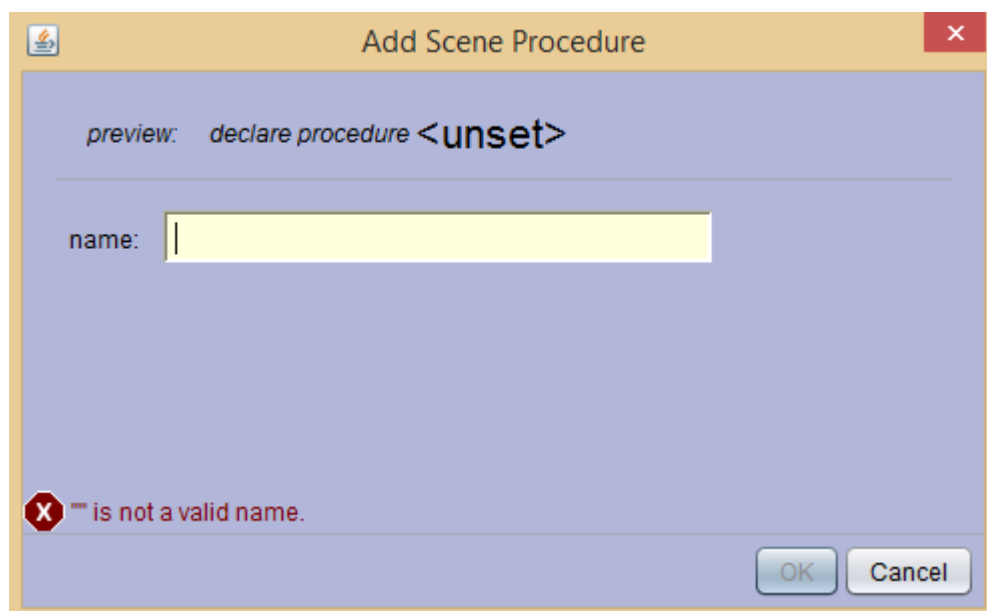
2.1. Procedure (Thủ tục)

Tạo một thủ tục mới: Chọn Add Scene Procedure để tạo một thủ tục mới cho lớp Scene (Hình 2)



Hình 2: Thủ tục của lớp Scene

Khai báo thủ tục trong Alice: **declare procedure <tên thủ tục>** (Hình 3)



Hình 3: Bảng khai báo thủ tục

Trong đó: <tên thủ tục>: do người lập trình nhập tại (2)

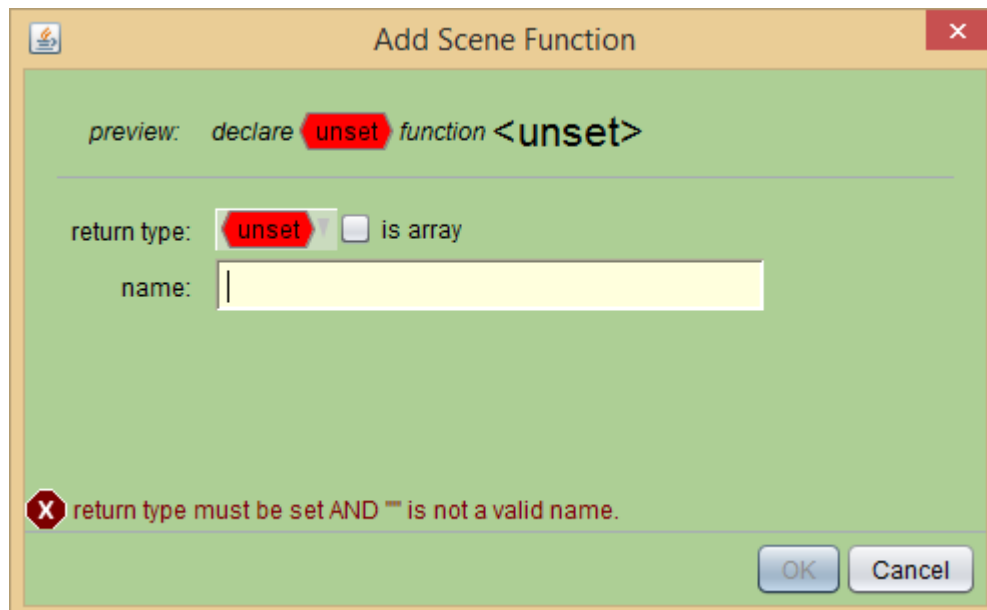
2.2. Function(Hàm):

Tạo một hàm mới: Chọn Add Scene Function để tạo một thủ tục mới cho lớp Scene (Hình 4)



Hình 4: Hàm của lớp Scene

Khai báo hàm trong Alice: **declare** <kiểu dữ liệu trả về> **function** <tên hàm> (Hình 5)



Hình 5: Bảng khai báo hàm

Trong đó:

<kiểu dữ liệu trả về>: người lập trình sẽ lựa chọn tại (2)

<tên hàm>: người lập trình sẽ khai báo tại (3)

Lưu ý: Thủ tục sẽ không có giá trị trả về tuy nhiên hàm sẽ có giá trị trả về

3. Vùng ảnh hưởng của một biến

Sẽ có 2 loại biến:

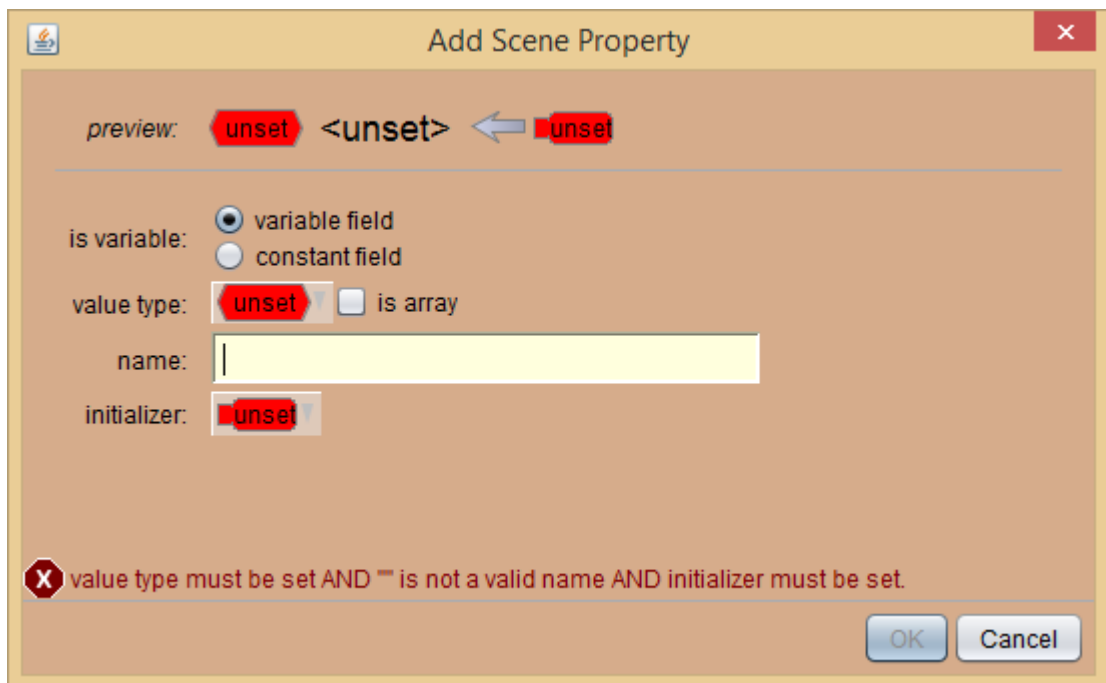
1. **Biến ảnh hưởng trong một lớp:** Khi biến này được gọi các thủ tục hay hàm đều sử dụng được

Tạo một biến mới: Chọn Add Scene Property để tạo một biến mới cho lớp Scene (Hình 6)



Hình 6: Thuộc tính của lớp Scene

Khai báo biến: <kiểu của biến><kiểu dữ liệu của biến><tên biến> ⇔ <giá trị ban đầu> (Hình 7)



Hình 7: Bảng khai báo biến

Trong đó:

<kiểu của biến>: sẽ được chọn trong (1)

variable field: là một biến bình thường, có thể thay đổi được giá trị trong quá trình lập trình

constant field: khi chọn vào ô này biến sẽ trở thành hằng, hằng không thể thay đổi được trong quá trình lập trình

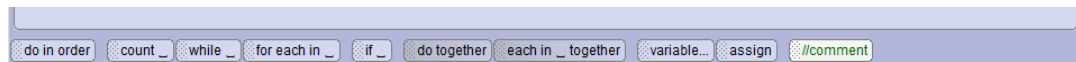
<kiểu dữ liệu>: sẽ được chọn trong (3)

<tên biến>: sẽ được người dùng tự đặt tên ở (4)

<giá trị ban đầu> sẽ được người dùng chọn trong (5)

2. **Biến ảnh hưởng trong một hàm hoặc thủ tục:** Khi biến này được gọi thì chỉ duy nhất thủ tục hay hàm đó sử dụng mà hàm khác không thể sử dụng

Tạo một biến mới trong hàm hay thủ tục: Kéo variable vào Bảng lập trình (Hình 8)



Hình 8: Available trên thanh điều khiển

Khai báo biến trong thủ tục hay hàm: <kiểu của biến><kiểu dữ liệu của biến><tên biến> ⇐ <giá trị ban đầu> (Hình 9)

Hình 9: Bảng khai báo biến trong thủ tục hoặc hàm

Trong đó:

<kiểu của biến>: sẽ được chọn trong (1)

variable: là một biến bình thường, có thể thay đổi được giá trị trong quá trình lập trình

constant: khi chọn vào ô này biến sẽ trở thành hằng, hằng không thể thay đổi được trong quá trình lập trình

<kiểu dữ liệu>: sẽ được chọn trong (3)

<tên biến>: sẽ được người dùng tự đặt tên ở (4)

<giá trị ban đầu> sẽ được người dùng chọn trong (5)

