



🏠 Trang chủ

Trang của tôi » Học kỳ I năm học 2018-2019 (Semester 1 - Academic year 2018-2019) » Chương trình Kỹ sư Tài năng (Training Program of Talent Engineers) » Khoa Khoa học và Kỹ thuật Máy tính (Faculty of Computer Science and Engineering) » Nguyên lý ngôn ngữ lập trình (CO3005)_Nguyễn Hứa Phùng (TN_HK181) » Điều khiển trình tự » Bài kiểm tra Điều khiển trình tự 2

Đã bắt đầu vào lúc Tuesday, 20 November 2018, 2:19 PM
Tình trạng Đã hoàn thành
Hoàn thành vào lúc Tuesday, 20 November 2018, 2:30 PM
Thời gian thực hiện 10 phút 47 giây
Điểm 20,00 của 20,00 (100%)

Câu hỏi 1
Hoàn thành
Điểm 1,00 của 1,00

Làm thế nào, trong cơ chế Gọi Trở về đơn giản (Simple Call-Return), lệnh **return** không chứa địa chỉ cần chuyển đến nhưng vẫn có thể thực hiện được việc chuyển điều khiển về chương trình gọi khi kết thúc thực thi chương trình con?

Chọn một:

- ☐ a. Nhờ địa chỉ quay trở về đã được xác định và ghi vào stack bởi compiler
- ☒ b. Nhờ địa chỉ quay trở về đã được ghi vào bản hoạt động của chương trình con bởi sự thực thi của một lệnh khác
- ☐ c. Nhờ địa chỉ là một giá trị ẩn trong lệnh return.
- ☐ d. Nhờ địa chỉ quay trở về đã được xác định và ghi vào bản hoạt động của chương trình con bởi compiler

Câu trả lời đúng là: Nhờ địa chỉ quay trở về đã được ghi vào bản hoạt động của chương trình con bởi sự thực thi của một lệnh khác

Câu hỏi 2
Hoàn thành
Điểm 3,00 của 3,00

Cho chương trình con sau viết bằng Scala:

```
object Timer {  
  def apply(interval: Int,  
            repeats: Boolean = true)  
    (op: => Unit) {  
    val timeOut = new javax.swing.AbstractAction() {  
      def actionPerformed  
        (e: java.awt.event.ActionEvent) = op  
    }  
    val t = new javax.swing.Timer(interval, timeOut)  
    t.setRepeats(repeats)  
    t.start()  
  }  
}
```

Trong đoạn chương trình trên có khai báo tường minh của phương thức có tên là (nếu có nhiều tên thì các tên viết theo thứ tự xuất hiện trong chương trình, cách nhau bằng dấu ',' và không có khoảng trắng) với các tên thông số là (các thông số xuất hiện theo thứ tự xuất hiện trong chương trình, cách nhau bằng dấu ',', các nhóm thông số của các chương trình con khác nhau cách nhau bằng dấu ';' và không có khoảng trắng).

Câu hỏi 3
Hoàn thành
Điểm 1,00 của 1,00

Trong các cơ chế truyền thông số, cơ chế nào làm thay đổi thông số thực ngay khi thông số hình thức bị thay đổi mà không chờ đến khi chương trình được gọi kết thúc?

Chọn một hoặc nhiều hơn:

- ☒ a. Pass by name
- ☐ b. Pass by value-result
- ☒ c. Pass by reference
- ☐ d. Pass by result

Câu trả lời đúng là: Pass by reference, Pass by name

Câu hỏi 4
Hoàn thành
Điểm 3,00 của 3,00

Cho đoạn chương trình sau viết bằng Pascal:

var X:real; **toàn cục** **x = 4**

procedure SUB2(X,Y:real;function F(U:real):real);

var Z:real;

begin **CT con 1**

 Z := abs(X - Y);

 Z := (F(X) + F(Y)) * Z / 2;

 write(Z);

procedure SUB1;

var Y:real; **y biết Y = 2**

function FUNC(V:real):real;

begin **4*V+2: static lấy x của toàn cục.**

4*V+2 deep lấy x lại chương trình chính

1*V+2: shallow lấy x tại ct con

FUNC := X * V + Y; **X trong cục bộ ko định nghĩa**

begin **CT con 2**

 Y := 2;

 SUB2(1,2,FUNC);

begin **CT chính**

 X := 4;

 SUB1;

end.

Đề đảm bảo trùng khớp, kết quả phải viết ở dạng D.D (một số trước và một số sau dấu chấm).

a. Pascal là ngôn ngữ static-scoping, vậy giá trị z được in ra trong chương trình con SUB2 là

b. Giả sử chương trình trên được viết trên một ngôn ngữ dynamic-scoping dùng shallow binding thì giá trị z được in ra khi SUB2 được thực thi là

c. Giả sử chương trình trên được viết trên một ngôn ngữ dynamic-scoping dùng deep binding thì giá trị z được in ra khi SUB2 được thực thi là

Câu hỏi 5
Hoàn thành
Điểm 3,00 của 3,00

1. Khi lập trình cho các giao diện Windows, người lập trình sẽ định nghĩa các phương thức sẽ được thực thi khi một nút nhấn được chọn, một văn bản được nhập,... Cơ chế là cơ chế gọi chương trình con nào chủ yếu được sử dụng trong trường hợp này.

2. Khi lập trình cho một hệ thống đòi hỏi các đáp ứng phải xảy ra đúng theo các mốc thời gian bất kể máy chạy nhanh hoặc chậm, cơ chế là cơ chế gọi chương trình con cần được sử dụng trong trường hợp này.

3. Hiện tượng tương tranh (race condition) hoặc khóa chết (deadlock) có thể xảy ra khi dùng cơ chế gọi chương trình con .

4. Cơ chế thường được sử dụng để việc xử lý lỗi được tách bạch khỏi logic của chương trình.

Câu hỏi 6
Hoàn thành
Điểm 4,00 của 4,00

Trong các cơ chế truyền thông số,

1. Cơ chế không có thông số thực

2. Cơ chế không tính toán thông số thực mà truyền mã của thông số thực cho thông số hình thức

3. Cơ chế buộc chương trình được gọi không thể thay đổi giá trị thông số hình thức

4. Cơ chế ngăn chặn những thay đổi trên thông số hình thức dẫn đến thay đổi trên thông số thực sau khi kết thúc chương trình được gọi

Câu hỏi **7**
Hoàn thành
Điểm 4,00 của 4,00

Trong các cơ chế gọi chương trình con,

1. cơ chế cho phép có thể chuyển điều khiển từ chương trình gọi sang một điểm không phải là điểm bắt đầu của chương trình được gọi
2. cơ chế cho phép có thể chuyển điều khiển từ chương trình gọi sang chương trình được gọi mà không có lệnh gọi tương minh?
3. cơ chế cho phép chương trình gọi thực hiện tiếp lệnh kế tiếp sau lệnh gọi mà không chờ chương trình được gọi kết thúc?
4. Cơ chế gọi chương trình con khi hiện thực sẽ tạo ra nhiều bản hoạt động của cùng một chương trình con khi thực thi?

Câu hỏi **8**
Hoàn thành
Điểm 1,00 của 1,00

Hiện tượng tương tranh (race condition) hoặc khoá chết (deadlock) có thể xảy ra khi dùng cơ chế gọi chương trình con nào?

Chọn một:

- ☐ a. Biến cố (Exception)
- ☐ b. Đệ qui (Recursive Call)
- ☐ c. Định thời (Scheduled)
- ☐ d. Song hành (Coroutine)
- ☒ e. Công tác (Task)

Câu trả lời đúng là: Công tác (Task)

Copyright 2007-2014 BKĐT-Đại Học Bách Khoa Tp.HCM. All Rights Reserved.

Địa chỉ: Nhà A1- 268 Lý Thường Kiệt, Phường 14, Quận 10, Tp.HCM. Email: elearning@hcmut.edu.vn

Phát triển dựa trên hệ thống Moodle