

Cho biết trong đoạn mã sau, đối tượng nào có nhiều tên?

```
int *p = new int;
```

```
int *q = p;
```

- ☐ p
- ☐ q
- ☒ Đối tượng được tạo ra bởi new int
- ☐ int

Đúng

Đối tượng dữ liệu có nhiều tên trong đoạn mã C trên có những tên gì?

- ☐ p
- ☐ p và q
- ☐ \*p và q
- ☒ \*p và \*q

Đúng, dựa vào 2 tên này, programmer có thể truy xuất (đọc, viết) trên đối tượng được tạo ra bởi new int

Cho đoạn mã C++ sau:

```
template <class myType>
myType GetMax (myType a, myType b) {
    return (a>b?a:b);
}
```

Cho biết đoạn mã trên có thể là ví dụ minh họa cho khái niệm nào?

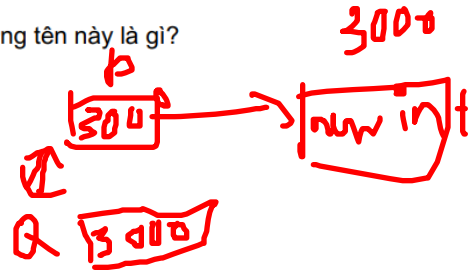
- ☐ Alias - bí danh
- ☒ Polymorphism - Đa hình
- ☐ Garbage-Rác
- ☐ Dangling reference-Tham chiếu treo

Đúng, đây là một ví dụ của đa hình (polymorphism): cũng với 1 tên là GetMax nhưng có thể tham chiếu đến những hàm khác nhau tùy theo kiểu của myType

Cho biết trong đoạn mã C++ sau, đối tượng nào có nhiều tên và những tên này là gì?

```
int *p = new int;
```

```
int *&q = p;    kiểu tham khảo của con trỏ, tức là q là alias của p
```



Chọn một:

- ☐ a. Đối tượng có nhiều tên là đối tượng được tạo ra bởi lệnh "new int" và các tên của đối tượng này là p và q
- ☒ b. Đối tượng có nhiều tên là p và các tên của nó là p và q ✓
- ☐ c. Đối tượng có nhiều tên là p và các tên của nó là p và \*q
- ☐ d. Không có câu nào đúng

Câu hỏi 3

Chính xác

Điểm 1,00 của 1,00

Cho đoạn mã C++ sau, hãy cho biết khi gọi hàm foo với thông số là 3 thì sẽ có lỗi gì (trong các lỗi tham chiếu treo, garbage hoặc alias) xảy ra?

```
int * m;
```

```
int * foo(int x) {
```

```
    static int y;
```

```
    int * z = new int;
```

```
    switch x {
```

```
        case 1: return &y;    có thể có alias + lỗi rác
```

```
        case 2: return &x;    tham chiếu tráo + tạo rác
```

```
        case 3: return z;    không gây lỗi
```

```
        case 4: return m;    lỗi rác + alias
```

```
    }
```

```
}
```



Chọn một:

- ☐ a. Gây ra lỗi tạo ra rác
- ☐ b. Gây ra alias
- ☒ c. Không gây ra lỗi gì cả ✓
- ☐ d. Gây ra lỗi tham chiếu treo (dangling reference)

Câu hỏi **4**

Chính xác

Điểm 1,00 của 1,00

Cho phát biểu C++ sau:

```
x = x + y * 10;
```

Trong biểu thức bên vế phải, phép nhân **\*** được tính toán trước phép **+**; ràng buộc này xảy ra vào thời gian nào?

Chọn một:

- ☐ a. Thời gian dịch (Compiling time)
- ☐ b. Thời gian thực thi (Runtime)
- ☐ c. Hiện thực ngôn ngữ (Language implementation time)
- ☒ d. Định nghĩa ngôn ngữ (Language design time) ✓

Câu hỏi **5**

Chính xác

Điểm 1,00 của 1,00

Cho đoạn mã C++ sau, hãy chọn các phát biểu ĐÚNG về thời gian sống (object lifetime) của các đối tượng?

```
int * m;  
int * foo(int x) {  
    static int y;  
    int * z = new int;  
    switch x {  
        case 1: return &y;  
        case 2: return &x;  
        case 3: return z;  
        case 4: return m;  
    }  
}
```

Chọn một hoặc nhiều hơn:

- ☐ a. Thời gian sống của đối tượng y bằng thời gian sống của đối tượng z
- ☒ b. Thời gian sống của đối tượng m bằng thời gian sống của đối tượng y ✓
- ☒ c. Thời gian sống của đối tượng z bằng thời gian sống của đối tượng x ✓
- ☐ d. Thời gian sống của đối tượng z bằng thời gian sống của đối tượng new int

Câu hỏi **7**

Chính xác

Điểm 1,00 của 1,00

Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal:

```
var x,y:integer; // line 1
```

```
  procedure foo(x:real) //line 2
```

```
    var z: integer; // line 3
```

```
    begin .... end
```

```
y//4,z//5,x//1,foo,foo1
```

```
  procedure foo1(y:integer) // line 4
```

```
    var z: real; // line 5
```

```
    begin ... end
```

```
begin ... end
```

Hãy cho biết các khai báo có hiệu lực trong procedure foo1

Chọn một hoặc nhiều hơn:

- ☐ a. Khai báo z ở line 3
- ☒ b. Khai báo z ở line 5 ✓
- ☐ c. Khai báo x ở line 2
- ☒ d. Khai báo x ở line 1 ✓
- ☒ e. Khai báo y ở line 4 ✓
- ☐ f. Khai báo y ở line 1
- ☒ g. Khai báo foo1 ở line 4 ✓
- ☒ h. Khai báo foo ở line 2 ✓

Câu hỏi 8

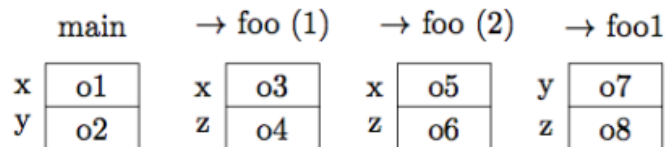
Đúng một phần

Điểm 0,47 của 1,00

Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal:

```
var x,y:integer; // line 1
procedure foo(x:real) //line 2
var z: integer; // line 3
begin .... end
procedure foo1(y:integer) // line 4
var z: real; // line 5
begin ... end
begin ... end
```

Giả sử chương trình chạy theo trình tự: main gọi foo, foo gọi đệ qui foo, foo gọi foo1 với các đối tượng được cấp phát trên các bảng ghi hoạt động như sau:



Hãy cho biết môi trường tham khảo động của foo1 có chứa các đối tượng nào?

Chọn một hoặc nhiều hơn:

- ☐ a. o6
- ☐ b. o3
- ☒ c. o8 ✓
- ☒ d. o1 ✓
- ☐ e. o4
- ☒ f. o7 ✓
- ☐ g. o2
- ☒ h. o5 ✗

Câu hỏi **6**

Đúng một phần

Điểm 0,67 của 1,00

Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal:

```
var x,y:integer; // line 1
```

```
  procedure foo(x:real) //line 2
```

```
    var z: integer; // line 3
```

```
    begin .... end
```

```
x//2,z//3,y//1,foo,foo1
```

```
  procedure foo1(y:integer) // line 4
```

```
    var z: real; // line 5
```

```
    begin ... end
```

```
begin ... end
```

Hãy cho biết các khai báo có hiệu lực trong procedure foo

Chọn một hoặc nhiều hơn:

- ☒ a. Khai báo x ở line 2 ✓
- ☐ b. Khai báo y ở line 4
- ☒ c. Khai báo z ở line 3 ✓
- ☒ d. Khai báo foo ở line 2 ✓
- ☒ e. Khai báo x ở line 1 ✗
- ☒ f. Khai báo foo1 ở line 4 ✓
- ☒ g. Khai báo y ở line 1 ✓
- ☐ h. Khai báo z ở line 5

Câu hỏi 9

Không chính xác

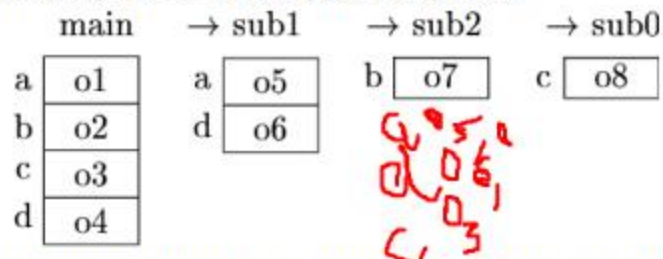
Điểm 0,00 của 1,00

Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal:

```
var a,b,c,d: integer; // line 1
procedure sub0
  var c: integer;      // line 2
begin .. end;
procedure sub1(a:real) //line 3
  var d:real;          //line 4
  procedure sub2(b:char) //line 5
  begin ... end
begin ... end
begin ... end
```

Đây là nguyên tắc cấu trúc khối tầm vực tính

Giả sử chương trình chạy theo trình tự: main gọi sub1, sub1 gọi sub2, sub2 gọi sub0 với các đối tượng được cấp phát trên các bảng ghi hoạt động như sau:



Hãy cho biết môi trường tham khảo động của sub0 có chứa các đối tượng nào?

Chọn một hoặc nhiều hơn:

- ☒ a. o6 ✗
- ☒ b. o7 ✗
- ☐ c. o3
- ☒ d. o4 ✓
- ☒ e. o5 ✗
- ☒ f. o1 ✓
- ☒ g. o2 ✓
- ☒ h. o8 ✓

**Câu hỏi 10**

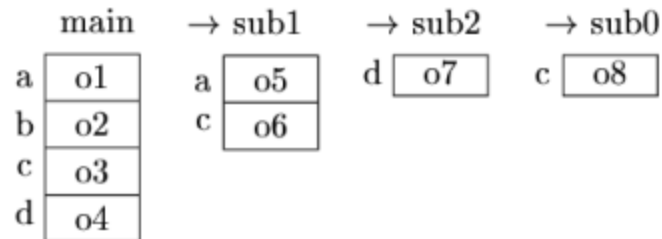
Chính xác

Điểm 1,00 của 1,00

Cho đoạn chương trình sau viết trên một ngôn ngữ tầm vực động:

```
program main;  
var a,b,c,d: integer; // line 1  
begin ... end  
procedure sub0  
var c: integer; // line 2  
begin .. end;  
procedure sub1(a:real) //line 3  
var c:real; //line 4  
begin ... end  
procedure sub2(d:char) //line 5  
begin ... end
```

Giả sử chương trình chạy theo trình tự: main gọi sub1, sub1 gọi sub2, sub2 gọi sub0 với các đối tượng được cấp phát trên các bảng ghi hoạt động như sau:



Hãy cho biết môi trường tham khảo động của sub0 có chứa các đối tượng nào?

Chọn một hoặc nhiều hơn:

- ☐ a. o1
- ☐ b. o6
- ☒ c. o8 ✓
- ☒ d. o7 ✓
- ☐ e. o4
- ☒ f. o5 ✓
- ☐ g. o3
- ☒ h. o2 ✓



Câu hỏi 4

Hoàn thành

Điểm 0,67 của 1,00

Cho đoạn chương trình sau viết trên ngôn ngữ C++ (các con số đầu dòng là thứ tự hàng):

```
1. int * x;
2. int foo(int *&y) {
3.     int m = 0;
4.     y = new int;
5.     return &m;
6. }
7. int main() {
8.     int *n = foo(x);
9.     x = new int;
10.    return *n;
}
```

khi gọi new int tại đây thì liên kết đến đối tượng new int trong hàm foo sẽ bị hủy không thể truy xuất đc nữa nên gây ra lỗi rác

Hãy chọn các phát biểu ĐÚNG trong các phát biểu sau?

Chọn một hoặc nhiều hơn: Hàng 8: truyền vào x cho hàm foo, do truyền tham khảo của con trỏ, nên x và y là alias, khi y trỏ đến đối tượng new int thì cũng như x trỏ đến đối tượng new int

- ☒ a. Hàng 8 tạo ra đối tượng có nhiều tên (alias) ✓
- ☐ b. Hàng 5 gây ra rác vì đối tượng new int ở hàng 4 không thể truy xuất
- ☒ c. Hàng 9 gây ra rác vì có đối tượng còn tồn tại nhưng không thể truy xuất được
- ☐ d. Hàng 8 tạo ra rác vì có đối tượng còn tồn tại nhưng không thể truy xuất được
- ☒ e. Hàng 10 gây ra lỗi tham chiếu treo vì truy xuất một đối tượng đã bị hủy bỏ
- ☐ f. Hàng 9 tạo ra đối tượng có nhiều tên (alias)

Câu hỏi 3

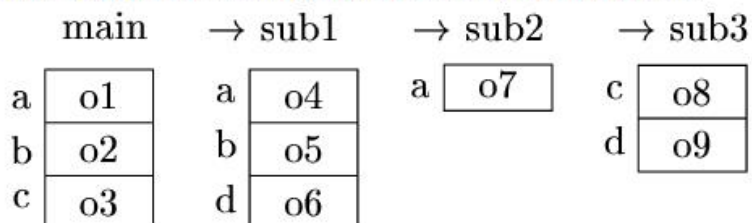
Hoàn thành

Điểm 1,00 của 1,00

Cho chương trình sau được viết trên một ngôn ngữ **tầm vực động (dynamic-scope)**:

```
program main;
var a,b,c:integer; //1
begin ... end // body of main
procedure sub1(a:real) //2
  var b,d:real;    //3
  begin ... end // body of sub1
procedure sub2(a:char) //4
  begin ... end //body of sub2
procedure sub3(c:real) //5
  var d:integer;   //6
  begin ... end // body of sub3
```

Giả sử khi thực thi, main sẽ gọi sub1, sub1 gọi sub2, và sub2 gọi sub3 với các đối tượng được cấp phát trên các bản hoạt động của các chương trình như sau:



Hãy viết danh sách các đối tượng trên các bản hoạt động mà sub3 có thể truy cập được?

Để bảo đảm trùng khớp, hãy viết các đối tượng theo thứ tự tăng dần của số sau chữ o, (ví dụ o3,o4,o9); các đối tượng được viết cách nhau bằng 1 dấu phẩy ',' và viết liên tục không có khoảng trắng nào.

Trả lời:

Câu hỏi 2

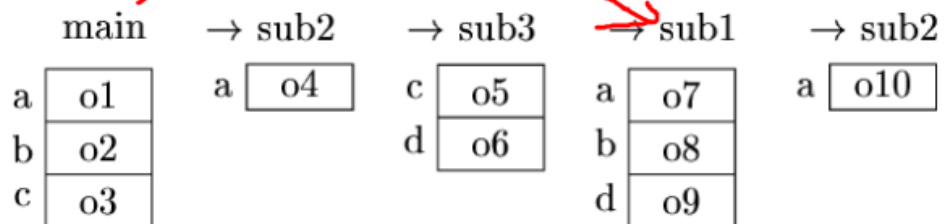
Hoàn thành

Điểm 0,00 của 1,00

Cho chương trình sau được viết trên một ngôn ngữ cấu trúc khối:

```
var a,b,c:integer; //1
procedure sub1(a:real) //2
    var b,d:real;    //3
    begin ... end // body of sub1
procedure sub2(a:char) //4
    procedure sub3(c:real) //5
        var d:integer; //6
        begin ... end // body of sub3
    begin ... end //body of sub2
begin ... end // body of main
```

Giả sử khi thực thi, main sẽ gọi sub2, sub2 gọi sub3, sub3 gọi sub1 và sub1 gọi sub2 với các đối tượng được cấp phát trên các bản hoạt động của các chương trình như sau:



Hãy viết danh sách các đối tượng trên các bản hoạt động mà sub2 ở lần chạy thứ hai có thể truy cập được?

Để bảo đảm trùng khớp, hãy viết các đối tượng theo thứ tự tăng dần của số sau chữ o, (ví dụ o3,o4,o9); các đối tượng được viết cách nhau bằng 1 dấu phẩy ',' và viết liên tục không có khoảng trắng nào.

Trả lời:

o2,o3,o10

~~o3,o8,o9,o10~~

**Câu hỏi 1**

Hoàn thành

Điểm 0,00 của 1,00

Cho chương trình sau được viết trên một ngôn ngữ cấu trúc khối:

```
var a,b,c:integer; //1
procedure sub1(a:real) //2
    var b,d:real;    //3
    begin ... end // body of sub1
procedure sub2(a:char) //4
    procedure sub3(c:real) //5
        var d:integer;    //6
        begin ... end // body of sub3
    begin ... end //body of sub2
begin ... end // body of main
```

Hãy viết môi trường tham khảo (referencing environment) của hàm sub3?

Để bảo đảm trùng khớp, hãy viết các danh hiệu theo thứ tự alphabet, sau mỗi danh hiệu (ứng với biến) có thêm //line, ví dụ a//1 hoặc c//5; các danh hiệu được viết cách nhau bằng 1 dấu phẩy ',' và viết liên tục không có khoảng trắng nào.

Trả lời:  [a//4,b//1,c//5,d//6,sub1,sub2,sub3](#)

**Câu hỏi 1**

Hoàn thành

Điểm 0,50 của 1,00

Cho đoạn mã C++ sau, hãy cho biết khi gọi hàm foo với thông số là 2 thì sẽ có lỗi gì (trong các lỗi tham chiếu treo, garbage hoặc alias) xảy ra?

```
int * m;
int * foo(int x) {
    static int y;
    int * z = new int;
    switch x {
        case 1: return &y;
        case 2: return &x;
        case 3: return z;
        case 4: return m;
    }
}
```

Chọn một hoặc nhiều hơn:

- ☒ a. Gây ra lỗi tham chiếu treo (dangling reference)
- ☒ b. Gây ra lỗi tạo ra rác
- ☐ c. Không gây ra lỗi gì cả
- ☐ d. Gây ra alias

Câu trả lời đúng là: Gây ra lỗi tham chiếu treo (dangling reference), Gây ra lỗi tạo ra rác

**Câu hỏi 2**

Hoàn thành

Điểm -0,33 của  
1,00

Cho đoạn mã Python sau:

`x = "abcdef"``y = 1.25``z = x + y`Biểu thức `x + y` sẽ gây ra lỗi kiểu, cho biết thời gian xảy ra lỗi kiểu?

Chọn một:

- ☐ a. Thời gian dịch (Compiling time)
- ☒ b. Định nghĩa ngôn ngữ (Language design time)
- ☐ c. Hiện thực ngôn ngữ (Language implementation time)
- ☐ d. Thời gian thực thi (Runtime)

Câu trả lời đúng là: Thời gian thực thi (Runtime)

**Câu hỏi 3**

Hoàn thành

Điểm 1,50 của 2,00

Cho đoạn mã C++ sau,

```
int * m;  
int * foo(int x) {  
    static int y;  
    int * z = new int;  
    switch x {  
        case 1: return &y;  
        case 2: return &x;  
        case 3: return z;  
        case 4: return m;  
    }  
}
```

Hãy điền tên của biến thích hợp vào các chỗ trống sau:

Chú ý:

- Chỉ điền tên 1 biến vào ô trống ngay cả khi có nhiều tên biến thích hợp để điền
- Nếu cần điền đối tượng được tạo ra bởi `new int` thì ghi là **new int** (7 ký tự bao gồm 1 khoảng trắng)

Đối tượng `x` có thời gian sống bằng với biến `z`.Đối tượng ~~`x`~~ `z` có thời gian sống bằng thời gian thực thi của hàm `foo`.Đối tượng `y` có thời gian sống bằng với biến `m`.Đối tượng `new int` có thời gian sống dài hơn thời gian thực thi của hàm `foo` nhưng không bằng thời gian thực thi của chương trình chính.**Câu hỏi 4**

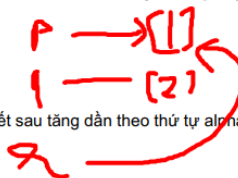
Hoàn thành

Điểm 1,00 của 1,00

Cho biết trong đoạn mã Python sau, đối tượng nào có nhiều tên và những tên này là gì?

`p = [1]``q = [2]``r = p`

Để đảm bảo so trùng, viết đối tượng trước, các tên viết sau tăng dần theo thứ tự alphabet; các đối tượng và tên cách nhau bằng dấu phẩy, không có khoảng trắng.

Trả lời: `[1],p,r`Câu trả lời đúng là: `[1],p,r`

Câu hỏi **5**

Hoàn thành

Điểm 0,00 của 1,00

Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal:

```
var x,y:integer; // line 1

procedure a(x:real) //line 2

var z: integer; // line 3

begin .... end

procedure b(y:integer) // line 4

var z: real; // line 5

begin ... end

begin ... end
```

Hãy viết các tên có hiệu lực trong procedure b? Để đảm bảo so trùng, viết các tên tăng dần theo thứ tự alphabet và cách nhau bằng dấu phẩy, không có khoảng trắng. Chỉ kể các tên có trong đoạn chương trình trên, nếu có nhiều tên trùng nhau (như x,y,z) phải viết kèm dấu // và số line của khai báo, ví dụ x//1 hoặc y//4.

Trả lời:

Câu trả lời đúng là: a,b,x//1,y//4,z//5

Câu hỏi **7**

Hoàn thành

Điểm 3,00 của 3,00

Cho đoạn mã C++ sau, hãy cho biết các đối tượng được cấp phát trong các vùng nhớ nào?

```
int * m;
int * foo(int x) {
    static int y;
    int * z = new int;
    switch x {
        case 1: return &y;
        case 2: return &x;
        case 3: return z;
        case 4: return m;
    }
}
```

Biến m được cấp phát trong vùng nhớ

Biến x được cấp phát trong vùng nhớ

Biến y được cấp phát trong vùng nhớ

Biến z được cấp phát trong vùng nhớ

Đối tượng được tạo ra bởi new int được cấp phát trong vùng nhớ

Câu hỏi 1

Hoàn thành

Điểm 1,00 của 3,00

Cho đoạn chương trình sau viết trên một ngôn ngữ cấu trúc khối tựa Pascal:

```
var x,y:integer; // line 1
procedure a(x:real) //line 2
var z: integer; // line 3
procedure b(y:integer) // line 4
var z: real; // line 5
begin ... end //of b
begin ... end // of a
begin ... end
```

Trong các câu sau, để đảm bảo so trùng, viết các tên tăng dần theo thứ tự alphabet và cách nhau bằng dấu phẩy, không có khoảng trắng. Chỉ kể các tên có trong đoạn chương trình trên, nếu có nhiều tên trùng nhau (như x,y,z) phải viết kèm dấu // và số line của khai báo, ví dụ x//1 hoặc y//4.

Môi trường tham khảo cục bộ của thủ tục b gồm có các tên  y//4,z//5

Môi trường tham khảo không cục bộ của thủ tục b gồm có các tên  a,b,x//2

Môi trường tham khảo toàn cục của thủ tục b gồm có các tên  toàn cục: không thuộc ct con nào hết.

Câu hỏi 3

Hoàn thành

Điểm 3,00 của 3,00

Cho đoạn chương trình sau viết trên ngôn ngữ C++ (các con số đầu dòng là thứ tự hàng):

```
1. int * a;
2. int foo(int *&b) {
3.     int c = 0;
4.     b = new int;
5.     return &c;
6. }
7. int main() {
8.     int *d = foo(a);
9.     a = new int;
10.    return d*;
}
```

Nhắc lại, trên ngôn ngữ C++, tham số được khai báo với dấu & (&b) là tham số truyền tham khảo, biểu thức &c trả về địa chỉ của c. Hãy chọn các phát biểu ĐÚNG trong các phát biểu sau?

Phát biểu ở dòng 5 gây ra lỗi

Phát biểu ở dòng 8 gây ra lỗi

Phát biểu ở dòng 9 gây ra lỗi

**Câu hỏi 4**

Hoàn thành

Điểm 1,00 của 1,00

Cho đoạn mã C++ sau:

```
int s = 0;
```

```
int *p = &s;
```

```
int *q = p;
```

```
int r = *p;
```

Nhắc lại tác vụ &s trên C++ trả về địa chỉ của biến s, \*p trả về giá trị nguyên của đối tượng mà p trỏ đến.

Hãy viết theo thứ tự alphabet các alias được tạo ra trong đoạn code trên?

Đề đảm bảo trùng khớp, các tên phải được viết:

- cách nhau bằng dấu phẩy, không có khoảng trắng,
- theo thứ tự alphabet (ví dụ p,q)
- nếu có dấu \*, & thì các dấu này viết trước ký tự (ví dụ \*r,p), dấu \* viết trước & (ví dụ \*s,&r)

Trả lời:

**Câu hỏi 5**

Hoàn thành

Điểm 0,00 của 1,00

Cho chương trình sau được viết trên một ngôn ngữ cấu trúc khối:

```
var a,b,c:integer; //1
```

```
procedure sub1(a:real) //2
```

```
    var b,d:real;    //3
```

```
    begin ... end // body of sub1
```

```
procedure sub2(a:char) //4
```

```
    procedure sub3(c:real) //5
```

```
        var d:integer;    //6
```

```
        begin ... end // body of sub3
```

```
    begin ... end //body of sub2
```

```
begin ... end // body of main
```

Hãy viết môi trường tham khảo (referencing environment) của hàm sub2?

Đề bảo đảm trùng khớp, hãy viết các danh hiệu theo thứ tự alphabet, sau mỗi danh hiệu (ứng với biến) có thêm //line, ví dụ a//1 hoặc c//5; các danh hiệu được viết cách nhau bằng 1 dấu phẩy ',' và viết liên tục không có khoảng trắng nào.

Trả lời:  [a//4,b//1,c//1,sub1,sub2,sub3](#)



Câu hỏi 6

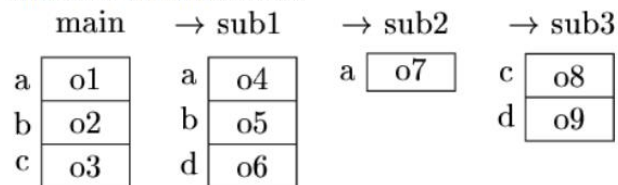
Hoàn thành

Điểm 1,00 của 1,00

Cho chương trình sau được viết trên một ngôn ngữ cấu trúc khối:

```
var a,b,c:integer; //1
procedure sub1(a:real) //2
    var b,d:real;    //3
    begin ... end // body of sub1
procedure sub2(a:char) //4
    procedure sub3(c:real) //5
        var d:integer;    //6
        begin ... end // body of sub3
    begin ... end //body of sub2
begin ... end // body of main
```

Giả sử khi thực thi, main sẽ gọi sub1, sub1 gọi sub2, và sub2 gọi sub3 với các đối tượng được cấp phát trên các bản hoạt động của các chương trình như sau:



Hãy viết danh sách các đối tượng trên các bản hoạt động mà sub3 có thể truy cập được?

Để bảo đảm trùng khớp, hãy viết các đối tượng theo thứ tự tăng dần của số sau chữ o, (ví dụ o3,o4,o9); các đối tượng được viết cách nhau bằng 1 dấu phẩy ',' và viết liên tục không có khoảng trắng nào.

Trả lời: