Nguyên lý ngôn ngữ lập trình

Đề thi cuối kì

Thời gian : 120 **phút**

Sinh viên KHÔNG được phép mở tài liệu

Câu 1 : Cho lớp A và B là lớp con trực tiếp của root Object. Lớp C là lớp con của A, còn D là con của C. Có các khai báo phương thức thực thể foo() trong các lớp Object, A, B, và D (không có khai báo foo trong C). Các đoạn mã Java phía dưới có đoạn nào bị sai không? Nếu có , hãy giải thích tại sao sai.

(1) A x = new C();
(2) C y = new A();
(3) B b = new A();

Cho z được khai báo kiểu A, tức là A z và được gán cho một đối tượng cụ thể. Phương thức foo nào sẽ được thực thi khi gọi câu lệnh z.foo()? Hãy giải thích.

Câu 2 :

a) Hãy viết tất cả kết quả có thể có của x khi gọi hàm main() và kết thúc dòng //1. Các tham số được truyền theo tham khảo. Hãy giải thích các kết quả.

```
int func(int i){
        i+=5;
        return 4;
}
void main(){
        int x = 3;
        x = x + func(x);//1
}
```

b) Cho hàm sau , hãy cho biết kết quả trả về khi gọi f(4)

```
int f(int x){

if(x==0) return 1;
```

```
else if(x==1) throw E;
        else if(x==2) return f(1);
                else try{return f(x-1);} catch E {return x+1;}
};
```

Câu 3: Một kiểu ghi biến thể (variant record type) được cho như sau :

```
type TERM_TYPE = (MID_TERM, FINAL_TERM);
type LABS = (LAB_1, LAB_2, LAB_3, LAB_4, LAB_5, LAB_6, LAB_7, LAB_8);
var STUDENT: record
        ID: integer;
        PASS_ASSIGNMENT : boolean;
        PASS_LABS: set of LABS;
        GRADE: real;
        case TERM_CLASS:TERM_TYPE of
                MID_TERM: (MID_RATE: real);
                FINAL_TERM: (FINAL real;
                        PERMISSION: boolean)
        end
```

Giả sử kích thước các đối tượng kiểu **integer, real, boolean** lần lượt là 4,8 và 2 byte, và đối tượng kiểu **set** được thực hiện theo dạng chuỗi bit. Hãy giải thích và tính kích thước từng thành phần của một đối tượng kiểu STUDENT rồi tính tổng kích thước của nó.

Câu 4: Cho đoạn chương trình viết bằng ngôn ngữ tựa Crazy (có cấu trúc khối) như sau :

```
const x = 2;
function f(a,b:integer): integer;
begin
        a:=b*x; //*
        b:=b+1;
        return b+ a;
end;
procedure main();
begin
const x = 4;
        var i,j: integer;
        var arr:array[5] of integer;
        arr[0] := 5; arr[1] := 4; arr[2] := 3; arr[3] := 2 ; arr[4] : =1;
        i:=4;
        j:=f(I,arr[i]); //**
        writeInt(j);
end;
```

a.Hãy vẽ bảng danh hiệu (symbol table) và chồng tầm vực khi chương trình trên được dịch cho đến hết

+ câu lệnh (*)

+câu lệnh (**)

b.Giá trị của I,j,arr[0], arr[1], arr[2], arr[3], arr[4] là gì sau khi kết thúc lệnh (**), cho biết mọi tham số đều được truyền theo kiểu :

+ tham khảo.Hãy giải thích kết quả bằng cách vẽ các bảng ghi hoạt động trong chồng trung tâm.

+tên.Không cần vẽ chồng trung tâm nhưng vẫn phải giải thích.

Câu 5: Viết hàm trong Ocaml theo mô tả dưới đây.Sinh viên có thể dùng các hàm đã viết ở câu trên để làm câu dưới.Trong các hàm có sẵn trong Ocaml, sinh viên CHỈ ĐƯỢC PHÉP dùng các hàm **List.hd, List.tl , ::, match with, if** và đệ quy.

a.Xóa phần tử thứ K trong một danh sách.Giả sử phần tử thứ K luôn tồn tại trong danh sách(nếu có lỗi, raise bất cứ thứ gì sinh viên muốn).

Ví dụ:

remove_at *[1,2,3,5] 3;;*

-: [1,2,5]

b.Dưới đây là ví dụ về một hàm nhận hàm khác làm thông số(hàm bậc cao):

let *applyTwice f x = f (f x);;*

-:val applyTwice :('a -> 'a) -> 'a -> 'a = <fun>

let *quad = applyTwice (func x->x*2);;*

-:val quad: int-> int = <fun>

quad 2;;

-: int = 8

Hãy viết hàm **takeWhile** nhận 2 thông số. Thông số thứ nhất là một vị từ f (tức là hàm trả về giá trị kiểu bool), thông số thứ 2 là danh sách có tên là lst. Hàm trả về một danh sách các phần tử ei đi từ đầu lst , tiếp tục lấy miễn f(ei) còn đúng.

Ví dụ :

takeWhile (fun x->x>0) [1;2;3;-1;4];;

-:int list= [1;2;3]

Xem thêm ví dụ câu c

c.Dùng hàm currying để viết **takeWhileEven** có thông số duy nhất là một danh sách lst. Nó trả về một danh sách các phần tử ei đi từ đầu lst, tiếp tục lấy miễn là ei vẫn còn là số chẵn.Gợi ý : Ocaml có toán tử mod để chia lấy phần dư và toán tử so sánh bằng là ==(hai dấu bằng).

Ví dụ : sinh viên có thể chép định nghĩa hàm vào giấy làm bài rồi điền vào chỗ trống các lệnh cần thiết, không cần chép phần xuất kết quả từ Ocaml

let rec *takeWhile f lst = <student works here>;;*

-: val takeWhile : ('a->bool)->'a list -> 'a list = <fun>

let *takeWhileEven = takeWhile < student works here>;;*

-:val takeWhileEven : int list -> int list = <fun>

takewhileEven [2;4;6;8;7;5;4;0;2];;

-:int list = [2;4;6;8]