

Lezione 3 – Esercitazione

prof. Marcello Sette

<mailto://marcello.sette@gmail.com>

<http://sette.dnsalias.org>

Esercizio 1

Si consideri il codice seguente:

```
program esercizio1 (input, output);
  var a,b,c,d: integer;

  procedure p1;
    var d,x: integer;
  begin
    {QUI}
  end;

  procedure p2;
    var b,c: integer;
    procedure p3;
      var b,x: integer;
    begin
      a:=1; p2; x:=a; p1
    end;
  begin
    if a=0 then p3 else p1
  end;

begin
a:=0; p2; p1
end.
```

Supponendo ereditarietà in ambito statico, rappresentare lo stack di attivazione nel momento in cui p1 è chiamato la prima volta, nel punto marcato {QUI}.

Esercizio 2

Si consideri il codice seguente:

```
program esercizio2 (input, output);
  var a,b,c: integer;

  procedure p1 ([MODE] a,b: integer);
  begin
    a:= a*b;
    if (c/b)=a then a:=0 else a:=100
  end;

  procedure p2 ([MODE] a,b: integer);
  begin
    a:= a-b;
    if a=c then p1(b,a) else p1(a,b)
  end;

begin
  a:=1; b:=5; c:=10;
  p2(c,b);
  writeln(a, b, c)
end.
```

Supponendo ereditarietà in ambito statico, valutare l'uscita del programma quando, per entrambe le occorrenze di [MODE], viene usato uno dei seguenti meccanismi di passaggio dei parametri:

1. IN realizzato per riferimento;
2. IN realizzato per copia;
3. OUT realizzato per riferimento;
4. OUT realizzato per copia;
5. IN OUT realizzato per riferimento;
6. IN OUT realizzato per copia.

Esercizio 3

Si consideri il seguente codice Pascal:

```
program esercizio3 (input, output);
  var limit: integer;

  function sommatoria ([MODE] lim: integer): integer;
    var s: integer;
  begin
    s:= 0;
    while lim > 0 do
      begin
        s:= s + limit;
        limit:= limit - 1
      end;
    sommatoria:= s
  end;

begin
  limit:= 6;
  writeln('Sommatoria da 1 a 6: ', sommatoria(limit))
end.
```

Quale meccanismo di passaggio dei parametri provocherebbe un ciclo infinito? E quale invece farebbe funzionare il programma?

Esercizio 4

Si consideri il seguente programma:

```
program esercizio4 (input, output);
  var a,b,c: integer;
      d: boolean;

  procedure p1 (var q:boolean; var r, s: integer);
  begin
    if d then r:=100 else r:= 200;
    s:= s/a
  end;

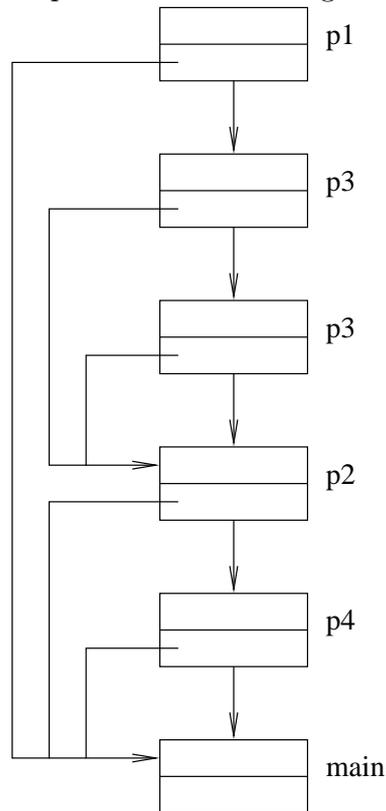
  procedure p2 (var x, y: integer; var z: boolean);
  begin
    x:=15; y:= x+a; z:= (x<a);
    p1(z,y,x);
    z:= (x<a)
  end

begin
  a:= -1; b:= -1; c:= -1; d:= true;
  p2(a,b,d)
end.
```

Supponendo valida l'ereditarietà in ambito statico ed usando il meccanismo di passaggio dei parametri standard del Pascal (**var**: parametri IN OUT realizzati per riferimento; non **var**: parametri IN realizzati per copia), determinare il valore delle variabili locali al main alla terminazione del programma.

Esercizio 5

Determinare la struttura di annidamento di un programma, il cui stack di attivazione ad un certo istante della sua esecuzione è quello mostrata in figura:



Esercizio 6

Scrivere un programma che determini se, nel linguaggio usato, i parametri IN siano realizzati per riferimento o per copia.

Esercizio 7

Scrivere un programma che determini se, nel linguaggio usato, i parametri OUT siano realizzati per riferimento o per copia.

Esercizio 8

Scrivere un programma che determini se, nel linguaggio usato, i parametri IN OUT siano realizzati per riferimento o per copia.

Esercizio 9

Sapendo che il linguaggio in uso realizza i parametri OUT e IN OUT per copia, scrivere un programma che determini se tali parametri sono copiati da sinistra a destra oppure da destra a sinistra.

Esercizio 10

Scrivere un programma che determini, nel linguaggio in uso, se i parametri di ritorno VRP da una procedura sono parametri OUT oppure IN OUT.

Esercizio 11

Sia dato il programma:

```
program Exam (input, output);
  var a, b, c: integer;

  procedure p1([MODE1] a,c:integer);

    procedure p2([MODE2] a,b:integer);
      begin
        if a < b then
          p2(b, a)
        else
          b:= 1;
          c:= 1
        end;

      begin
        a:= 2; b:= 2;
        p2(b, c);
        writeln(a, b, c)
      end;

    procedure p3([MODE3] a,b:integer);
      begin
        a:= 3; c:= 3;
        p1(c, b);
        writeln(a, b, c)
      end;

    begin
      a:= 4; b:= 4; c:= 4;
      p3(b, c);
      writeln(a, b, c);
      readln
    end.
```

Versione semplice

Si supponga che il linguaggio usi ereditarietà statica e che i possibili meccanismi di passaggio di parametri siano:

- IN realizzato per copia, oppure
- INOUT realizzato per riferimento;

Per ognuna delle possibili scelte di $\text{MODE}_i, i = 1, \dots, 3$ tra quelle specificate in precedenza (8 scelte possibili, per un totale di 8 esercizi diversi):

1. rappresentare lo stack di esecuzione quando esso ha altezza massima;
2. specificare le variabili locali di ogni procedura e quelle ereditate da altre procedure;
3. discutere il comportamento del programma ed i valori intermedi e finali di **a**, **b**, **c**.

Soluzioni degli 8 esercizi P_i

P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
2 2 1	2 1 1	2 1 1	2 2 1	1 2 1	2 1 1	1 2 1	2 2 1
2 3 3	3 4 3	1 3 3	3 1 2	2 1 1	3 1 2	2 1 1	3 4 3
4 2 3	4 1 3	4 1 3	4 2 2	4 2 1	4 1 2	4 2 1	4 2 3

Versione per masochisti

Si supponga che il linguaggio usi ereditarietà statica e che i possibili meccanismi di passaggio di parametri siano:

- IN realizzato per copia, oppure
- INOUT realizzato per riferimento, oppure
- INOUT realizzato per copia.

Per ognuna delle possibili scelte di $\text{MODE}_i, i = 1, \dots, 3$ tra quelle specificate in precedenza (27 scelte possibili, per un totale di 19 esercizi diversi, oltre quelli della sottosezione precedente):

1. rappresentare lo stack di esecuzione quando esso ha altezza massima;
2. specificare le variabili locali di ogni procedura e quelle ereditate da altre procedure;
3. discutere il comportamento del programma ed i valori intermedi e finali di **a**, **b**, **c**.