

به نام خدا



درس طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

موضوع: کنترل داده (Data Control)

ارائه: صادق سلیمانی

www.Bioinformation.ir

جلسه دهم

داده

عناوین جلسه

- مقدمات
- اسامی
- Association
- Referencing Environments
- Scope
- زبان‌های Block Structure
- اشتراک داده بین برنامه و زیربرنامه‌ها

عناوین ارائه: مقدمات - محیط‌های ارجاع - Scope - زبان‌های بلوکی - اشتراک داده

مرور

● شش جزء اصلی یک کامپیوتر

1. Data

2. Primitive Operations

3. Sequence Control

4. Data Control

عملگرهای یک عملوند چه هستند و از کجا تأمین می‌شوند

5. Storage Management

6. Operating Environment

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

3 از 29

عناوین ارائه: مقدمات - محیط‌های ارجاع - Scope - زبان‌های بلوکی - اشتراک داده

آشنایی با مباحث کنترل داده

● مثالی از مباحث کنترل داده

$$x=y+z+t$$

● آیا اسمی در این عبارت؟

● پارامترهای Formal هستند؟

● اسمی Local هستند؟

● اسمی Global هستند؟

● اسم یک زیربرنامه، بدون پارامتر هستند؟

● Type آن‌ها چیست؟

● عملوند ممکن است به صورت مستقیم مقداری را منتقل کند یا از طریق یک اسم:

● مثال: $y=x+2$

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

4 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اسامی

- ساده: t, x, y
- مرکب: $A[i].x$
- انواع اسامی
 1. اسم متغییر
 2. اسم یک زیربرنامه
 3. نام پارامتر فرمال
 4. اسم یک برچسب
 5. اسم یک Exception Handler
 6. اسم یک Type
- 7. اسم یک Operation
- 8. اسم یک Literal
- 9. اسم ثابتها

طراحی و پیاده‌سازی زبانهای برنامه‌سازی

5 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

Association (تناظر)

- متناظر شدن یک اسم به یک شیء داده یا یک اسم به تعریف یک زیربرنامه
- در اجرای یک برنامه:
 1. در شروع اجرا
 2. در حین اجرا
 3. فراخوانی یک زیربرنامه‌ی دیگر توسط برنامه
 4. Referencing Operationها
 5. با پایان اجرای زیربرنامه

طراحی و پیاده‌سازی زبانهای برنامه‌سازی

6 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

محیطهای ارجاع

1. Local Referencing Environment
2. Non-local Referencing Environment
3. Global Referencing Environment
4. Predefined Referencing Environment

طراحی و پیاده‌سازی زبانهای برنامه‌سازی

7 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

محیطهای ارجاع

```

program main;
var A, B, C: real;
procedure Sub1(A: real);
  var D: real;
  procedure Sub2 (C:real);
    var D: real;
    begin
      - Statements
      C := C+B;
      - Statements
    end;
  begin
    - Statements
    Sub2(B);
    - Statements
  end;
begin
  - Statements
  Sub1(A);
  - Statements
end.

```

• مثال

طراحی و پیاده‌سازی زبانهای برنامه‌سازی

8 از 29

عناوین ارائه: مقدمات-محیط‌های ارجاع-Scope-زبان‌های بلوکی - اشتراک داده

Scope

Visibility •

- یک تناظر (Association)، در حین اجرای برنامه، اگر بخشی از محیط ارجاع آن باشد، Visible و در غیر این صورت Hidden است

Dynamic Scope •

- مجموعه‌ی تمام Subprogram Activation‌هایی است که آن Association در داخل آن‌ها Visible است

Referencing Operations •

Data Object یا تعریف یک زیربرنامه -> محیط ارجاع * اسم Ref op:

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

9 از 29

عناوین ارائه: مقدمات-محیط‌های ارجاع-Scope-زبان‌های بلوکی - اشتراک داده

Scope

```

program main(output);
procedure Sub1(var J: integer);
begin
  ...
end;
procedure Sub2;
var I: integer;
begin
  ...
  Sub1(I);
  ...
end;
begin
  ...
  Sub2
  ...
end.

```

- گاهی یک شیء داده، در محیط‌های ارجاع مختلف، اسم‌های مختلفی دارد
- این امر مشکل ایجاد

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

10 از 29

عناوین ارائه: مقدمات-محیط‌های ارجاع-Scope-زبان‌های بلوکی - اشتراک داده

Scope

```

program main(output);
var I: integer;
procedure Sub1(var J: integer);
begin
...
end;
procedure Sub2;
begin
...
Sub1(I);
...
end;
begin
...
Sub2
...
end.

```

گاهی یک شیء داده، در یک محیط ارجاع دارای چند اسم است

این امر مشکل ایجاد

چند اسم = Aliases برای یک شیء داده

مشکل در درک برنامه

سعی در حذف یا محدود کردن نام مستعار

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

11 از 29

عناوین ارائه: مقدمات-محیط‌های ارجاع-Scope-زبان‌های بلوکی - اشتراک داده

Scope

تعریف یک Scope Deceleration

محدوده‌ای از کد برنامه است که آن اعلان اثر دارد

Dynamic Scope

برای Associationها تعریف می‌شود

مجموعه زیر برنامه‌هایی که Association در آن‌ها اعتبار دارد

Static Scope

برای اعلان تعریف می‌شود

زیربرنامه‌هایی که اعلان در آن‌ها اعتبار دارد

Dynamic Scope و Static Scope باید با هم سازگاری داشته باشند

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

12 از 29

عناوین ارائه: مقدمات-محیط‌های ارجاع-Scope-زبان‌های بلوکی - اشتراک داده

زبان‌های Block Structure

• تعریف

• برنامه در این زبان‌ها به صورت بلوک‌های داخل یکدیگر است
• ممکن است زیربرنامه‌ها نیز تو در تو باشند

• معرفی این زبان‌های با الگول آغاز شد
• پاسکال، PL/I و Ada از این نوع هستند

• Scope در زبان‌های Block Structure

• محدود به داخل بلاک
• برخورد زبان‌ها با نگهداری مقادیر بین اجراهای مختلف

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

13 از 29

عناوین ارائه: مقدمات-محیط‌های ارجاع-Scope-زبان‌های بلوکی - اشتراک داده

داده‌های مشترک در زیربرنامه‌ها

• نحوه‌ی انتقال اطلاعات بین برنامه‌ها و زیربرنامه‌ها

1. با کمک پارامترها

2. انتقال صریح

1. روش Common Area

2. روش Import و Export کردن متغیرها

3. انتقال ضمنی

1. با قواعد Static Scope

2. با قواعد Dynamic Scope

4. با کمک وراثت

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

14 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

1. Call By Value
2. Call By Value Result
3. Call By Result
4. Call By Reference
5. Call By Name

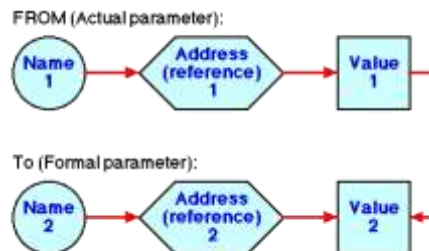
طراحی و پیاده‌سازی زبانهای برنامه‌سازی

15 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

• Call By Value



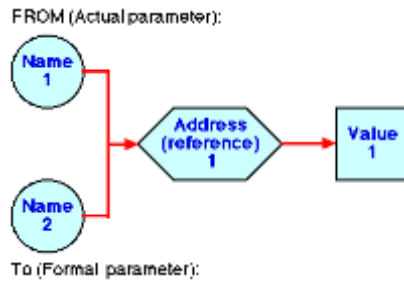
طراحی و پیاده‌سازی زبانهای برنامه‌سازی

16 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

Call By Reference



طراحی و پیاده‌سازی زبانهای برنامه‌سازی

17 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

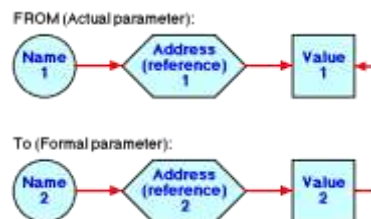
Call By Result

```
with CS_IO; use CS_IO;

procedure OUT_MODE_EXAMPLE is
  VALUE_1, VALUE_2: float;
  MEAN: float;

  procedure MEAN_VALUE (NUM_1, NUM_2: in float; NUM_3: out float) is
  begin
    NUM_3 := (NUM_1+NUM_2)/2.0;
  end MEAN_VALUE;

begin
  MEAN_VALUE(VALUE_1, VALUE_2, MEAN);
  put("The mean is ");
  put(MEAN);new_line;
  new_line;
end OUT_MODE_EXAMPLE;
```



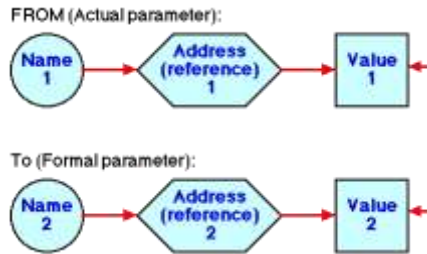
طراحی و پیاده‌سازی زبانهای برنامه‌سازی

18 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

Call By Value Result •



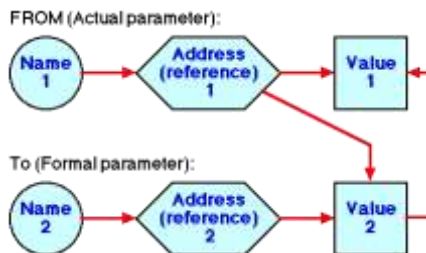
طراحی و پیاده‌سازی زبانهای برنامه‌سازی

19 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

Call By Name •



طراحی و پیاده‌سازی زبانهای برنامه‌سازی

20 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

```

program example;
var A:Array[1...10] of integer;
I:integer;
procedure Exchange(x,y:integer);
var temp:integer;
begin
temp ← x;
x ← y;
y ← temp;
end;
Begin(main)
I ← 4;
A[1] ← 8; A[2] ← 6; A[3] ← 4; A[4] ← 2;
Exchange(I , A[I]);
output(I , A[1] , A[2] , A[3] , A[4]);
End.

```

• مثال

اگر در برنامه روبرو تمام آرگومانها به صورت Call By Name فراخوانی شده باشند. خروجی آن چیست؟

طراحی و پیاده‌سازی زبانهای برنامه‌سازی

21 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

```

var s:array[1...3] of char;
var i j:integer;
procedure P(x:integer ; y:char)
var j:integer;
begin
j := 2;
x := x+1
output(y);
output(i);
end;
S[1] := 'A' ; S[2] := 'B' ; S[3] := 'C';
i := 0 ; j := 1;
P(i , S(1+ j) );
output(i);

```

• مثال

خروجی برنامه روبرو با استفاده از فراخوانی به روش Call By Name چیست؟

طراحی و پیاده‌سازی زبانهای برنامه‌سازی

22 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک پارامترها

مثال •

```

program main;
var k:integer;
procedure XYZ(i, j:integer);
var k:integer;
begin
i = 300; k = 2;
if i = j then j := i * k + j;
end;
begin
k = 100;
XYZ(k, k);
write(k);
end.

```

خروجی برنامه روبرو با استفاده از فراخوانی‌های زیر چیست؟

1. Call By Value
2. Call By Result
3. Call By Reference
4. Call By Name

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

29 از 23

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با کمک Common Area

زبان فرترن •

```

PROGRAM MAIN
INTEGER A
REAL F, R, X, Y
COMMON R, A, F
A = -14
R = 99.9
F = 0.2
CALL SUB(X, Y)
...
END

SUBROUTINE SUB(P, Q)
INTEGER I
REAL A, B, P, Q
COMMON A, T, B
...
END

```

Common Referencing Environment •

Main Program	Common Memory Storage	Subroutine
R	99.9	A
A	-14	I
F	0.2	B

در زبان Ada •
Package •

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

24 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

اشتراک داده با Import و Export متغیرها

```

procedure P(. . .);
  defines X, Y, Z; - X, Y, and Z become available for export
  X, Y, Z: real; - Usual declarations for X, Y, and Z
  U, V: integer; - Other local variables
begin . . . end; - Statements

procedure Q(. . .);
  uses P.X, P.Z; - Imports X and Z from P
  Y: integer; - Other declarations
begin . . . end; - Statements may include references to X and Z.
  
```

- همانند Extern در زبان C
- تفاوت بین Common Area و روش جاری؟
- هر متغیر اینجا دارای یک مالک است، یعنی زیربرنامه‌ایی که آن را اعلان کرده
- طول عمر چنین متغیرهایی تا پایان عمر مالک آن است

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

25 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

انتقال اطلاعات بین برنامه‌ها و زیربرنامه‌ها

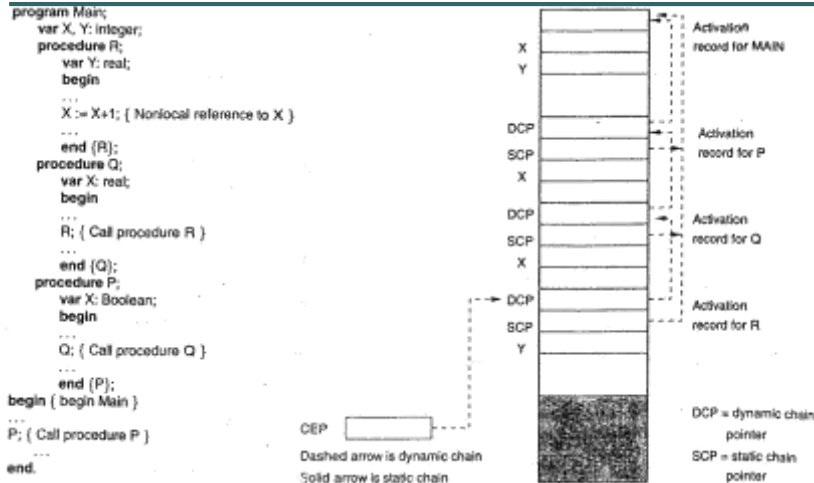
- با قواعد
- Static Scope
- با استفاده از Static Chain Pointer
- Dynamic Scope
- براساس قاعده Most Recently Created Association
- با ردگیری Dynamic Chain of Program Calls
- کنترل نوع پویا نیز نیاز دارد. چرا؟

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

26 از 29

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

انتقال اطلاعات بین برنامه‌ها و زیربرنامه‌ها

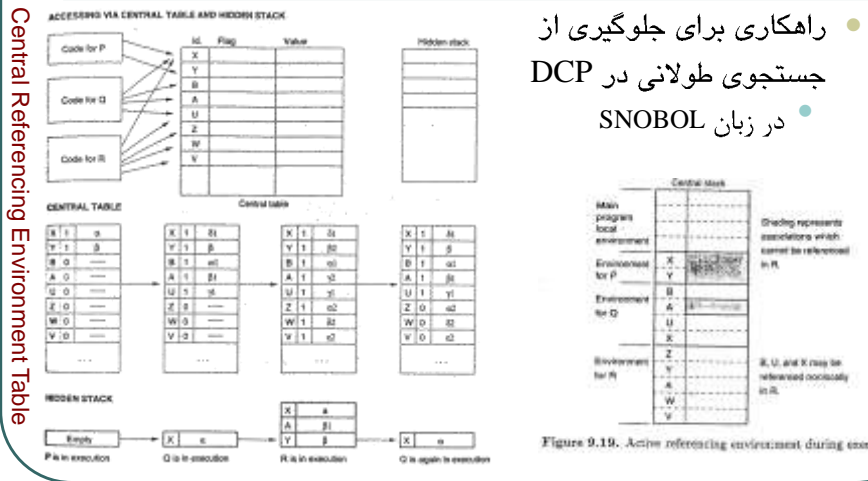


طراحی و پیاده‌سازی زبانهای برنامه‌سازی

29 از 27

عناوین ارائه: مقدمات-محیطهای ارجاع-Scope-زبانهای بلوکی - اشتراک داده

انتقال اطلاعات بین برنامه‌ها و زیربرنامه‌ها



- راهکاری برای جلوگیری از جستجوی طولانی در DCP
- در زبان SNOBOL

Central Referencing Environment Table

طراحی و پیاده‌سازی زبانهای برنامه‌سازی

29 از 28

داده

پرسش؟

طراحی و پیاده‌سازی زبان‌های برنامه‌سازی

29 از 29