

5- Amarrações

- Programas envolvem entidades, como subprogramas, variáveis e comandos
- Entidades têm propriedades ou atributos. Por exemplo, uma variável tem um nome, escopo, tempo de vida, valor, tipo, uma área de memória onde seu valor é guardado
- Amarração = especificação de alguns dos atributos de uma entidade
- Amarração estática = estabelecida antes da execução do programa e não pode ser mudada depois
- Amarração dinâmica = estabelecida durante a execução do programa e pode ser mudada, de acordo com as regras da LP

5- Amarrações...

- Amarrações e ambientes
 - Muitas LPs permitem que um dado identificador / possa ser declarado em várias partes do programa, possivelmente denotando entidades diferentes \Rightarrow a interpretação de uma expressão ou comando contendo uma ocorrência de / dependerá do contexto
 - Uma declaração produz uma associação ou amarração entre o identificador declarado e a entidade que ele irá denotar
 - Um ambiente é um conjunto de amarrações. Cada expressão e comando são interpretados em um ambiente particular

5- Amarrações...

- Denotáveis
 - Denotáveis = entidades que podem ser denotadas por identificadores
 - Exemplo: em Pascal, os denotáveis e os tipos de declarações em que podem ser amarrados são:
 - Valores primitivos e strings – definições de constantes
 - Referências a variáveis – declarações de variáveis
 - Abstrações de procedimentos e funções – definições de procedimentos e funções
 - Tipos – definições de tipos

5- Amarrações...

- Escopo
 - Trecho do programa onde uma declaração é efetiva, ou seja, um identificador é conhecido e pode ser usado

Estrutura de bloco

- Bloco = qualquer sentença de programa que delimita o escopo de quaisquer declarações que ele possa conter
- Estrutura de bloco = relacionamento textual entre blocos

5- Amarrações...

- Tipos de estruturas de bloco:
 - A. Monolítica: o programa inteiro é um único bloco



O escopo de toda declaração é o programa inteiro



Todas as declarações devem estar agrupadas em um único ponto do programa



Todas as entidades declaradas devem ter identificadores distintos

5- Amarrações...

- B. Nivelada (“flat”): o programa é particionado em blocos distintos



Uma variável pode ser declarada dentro de uma sub rotina, sendo, portanto, local à sub rotina

- C. Aninhada (“nest”): cada bloco pode estar aninhado dentro de outro bloco



Um bloco pode ser posicionado onde for conveniente e identificadores podem ser declarados dentro dele

5- Amarrações...

Escopo e visibilidade

- Ocorrência de amarração = ocorrência de um identificador no ponto em que é declarado
- Ocorrência de aplicação = ocorrência de identificador que denota um entidade
- Quando um programa contém mais de um bloco, um mesmo identificador pode ser declarado em blocos diferentes, denotando entidades diferentes
- O que acontece quando um mesmo identificador é declarado em dois blocos aninhados? A declaração mais externa é dita ser invisível ou escondida para a declaração mais interna

5- Amarrações...

Amarração estática e dinâmica

- Amarração estática ou escopo estático: o corpo da função é avaliado no ambiente de definição da função



Pode-se determinar em tempo de compilação a qual ocorrência de amarração corresponde uma dada ocorrência de aplicação de um identificador

- Amarração dinâmica ou escopo dinâmico: o corpo da função é avaliado no ambiente de chamada da função



Deve-se determinar em tempo de execução a qual ocorrência de amarração corresponde uma dada ocorrência de aplicação de um identificador

5- Amarrações...

- A amarração dinâmica não combina com tipagem estática
- Regras de escopo dinâmicas são fáceis de implementar, mas apresentam desvantagens dos pontos de vista da disciplina de programação e eficiência de implementação

- Exemplo 1:

```
const a = 2;  
function aumentado (d: Integer);  
begin  
    aumentado := d * a;  
end;
```

```
procedure ...;  
const a = 3;  
begin  
    ... aumentado(h) ...  
end;
```

```
begin  
    ... aumentado (h) ...  
end
```

5- Amarrações...

- Exemplo 2:
program dinamico;

var r: real;

procedure show; begin write (r : 5 : 3) end;

procedure small;
var r: real
begin
r := 0.125; show
end;

begin
r := 0.25;
show; small; writeln;
end.

5- Amarrações...

- Declarações
 - Uma declaração é uma frase de programa utilizada para elaborar amarrações
 - Declarações podem ser explícitas ou implícitas. Por exemplo, em Fortran, toda variável não declarada cujo nome comece com a letra I denota um inteiro
 - Objetivos de declarações:
 - Escolha da representação de armazenamento – no caso da declaração fornecer tipo de dado
 - Gerenciamento de memória – a partir do tempo de vida de objetos
 - Definição de operações que podem ser realizadas
 - Permitir verificação estática de tipos

5- Amarrações...

Definições

- Definição = uma declaração simples cujo único efeito é produzir amarrações
- Exemplo 1: Em Pascal, uma definição de constante amarra um identificador a um valor determinado em tempo de compilação
- Exemplo 2: Em ML, uma definição de valor amarra um valor a um identificador, possivelmente em tempo de execução
- Exemplo 3: Em Pascal, uma definição de procedimento amarra um identificador a uma abstração de procedimento e uma definição de função amarra um identificador a uma abstração de função

5- Amarrações...

Declarações de tipos

- Definição de tipo = serve somente para amarrar um identificador a um tipo existente. Encontrado em LPs com equivalência de tipos estrutural
- Declaração de novo tipo = cria um novo tipo distinto. Adequado para equivalência de tipos de nome
- Exemplo: Pascal

type pessoa = record

nome: string[30];

idade: integer;

altura: real;

end;

A elaboração dessa definição de tipo cria um novo tipo string anônimo

5- Amarrações...

Declarações de variáveis

- Definição de variável – serve somente para amarrar um identificador a uma variável já existente
- Declaração de variável – cria uma nova variável distinta
- Exemplo: ML

```
val count = ref 0 // cria uma nova variável inteira  
                // com valor inicial 0
```

```
val pop = populacao sub estado  
                // amarra pop a uma variável já  
                // existente
```

5- Amarrações...

Declarações colaterais

- Podem ser escritas na forma *D1 and D2*. O efeito é elaborar as subdeclarações *D1* e *D2* independentemente e combinar as amarrações produzidas. Nenhuma das subdeclarações pode usar um identificador declarado em outra subdeclaração
- Não são muito comuns
- Exemplo: ML
 - val pi = 3.14159**
 - and sin = fn (x: real) ⇒...**
 - and cos = fn (x: real) ⇒...**

5- Amarrações...

Declarações sequenciais

- Podem ser escritas na forma $D1 ; D2$. O efeito é elaborar a subdeclaração $D1$ seguida de $D2$, permitindo que as amarrações produzidas por $D1$ possam ser utilizadas em $D2$
- É o tipo mais comum

Declarações recursivas

- Utiliza amarrações que ela mesma produz
- Nem toda LP a suporta. As LPs mais modernas a suportam, mas, em geral, restringem as declarações recursivas a definições de tipos, procedimentos e funções
- Exemplo: Pascal – uma sequência de definições de tipos, procedimentos e funções é sempre tratada automaticamente como recursiva. Definições de constantes e declarações de variáveis são sempre tratadas como não recursivas

Escopo de declarações

〈 Visualizar no quadro 〉

5- Amarrações...

- Blocos

- Comandos de bloco

- Comando de bloco = um comando contendo uma declaração que produz amarrações que serão usadas somente para executar o comando
 - Apesar de um comando de bloco se comportar como qualquer outro comando, em Pascal não é permitido utilizá-lo com a mesma liberdade com que se usa comandos simples \Rightarrow o programador não pode colocar as declarações onde são necessárias

- Expressões de bloco

- Expressão de bloco = uma expressão contendo uma declaração que produz amarrações que serão usadas somente para avaliar a expressão

5- Amarrações...

Princípio de qualificação

- Princípio de qualificação = é possível incluir um bloco em qualquer classe sintática, desde que as sentenças desta classe especifiquem algum tipo de computação
- Declaração de bloco = um bloco contendo uma declaração local que produz amarrações que serão usadas somente para elaborar a declaração de bloco
- Essa construção dá suporte para o conceito de encapsulamento: um módulo grande pode declarar várias entidades, mas somente algumas delas podem ser exportadas. O uso de uma declaração de bloco permite manter pequena a interface do módulo, limitando o número de amarrações visíveis externamente ao módulo