

## PARADIGMAS DE PROGRAMAÇÃO

### EXERCÍCIOS LISTA 03: NOMES, VINCULAÇÕES E ESCOPOS

Prof.<sup>a</sup> Lucília Ribeiro

**01** Quais as questões de projetos para nomes?

**02** O que é o lado esquerdo de uma variável? O que é o lado direito?

**03** Defina:

- a) vinculação e tempo de vinculação.
- b) vinculação estática e vinculação dinâmica

**04** Defina:

- a) variáveis estáticas, dinâmicas da pilha, dinâmicas do monte explícitas e dinâmicas do monte implícitas
- b) tempo de vida, escopo, escopo estático, escopo dinâmico

**05** O que é o ambiente de referenciamento de uma sentença?

**06** Quais são as vantagens das constantes nomeadas?

**07** Algumas linguagens de programação não tem tipos. Quais são as vantagens e desvantagens óbvias de não ter tipos em uma linguagem?

**08** Considere o seguinte esqueleto em ADA. Assuma que a execução desse programa é na seguinte ordem de unidades:

Main chama Sub1

Sub1 chama Sub2

Sub2 chama Sub3

a) Assumindo escopo estático, a seguir, qual declaração de X é a correta para uma referência a X?

- i) Sub1
- ii) Sub2
- iii) Sub3

b) Repita a parte "a", mas assumo escopo dinâmico.

```

procedure Main is
  X : Integer;
  procedure Sub3;      -- essa é uma declaração de Sub3
  procedure Sub1 is
    X : Integer;
    procedure Sub2 is
      begin           -- de Sub2
        ...
      end;           -- de Sub2
    begin           -- de Sub1
      ...
    end;           -- de Sub1
  procedure Sub3 is
    begin           -- de Sub3
      ...
    end;           -- de Sub3
  begin           -- de Main
    ...
  end;           -- de Main

```

**09** Considere o programa a seguir. Liste todas as variáveis, com as unidades de programa onde elas estão declaradas, visíveis nos corpos de Sub1, Sub2 e Sub3, assumindo que o escopo estático esteja sendo usado.

```

procedure Main is
  X, Y, Z : Integer;
  procedure Sub1 is
    A, Y, Z : Integer;
    procedure Sub2 is
      A, B, Z : Integer;
      begin           -- de Sub2
        ...
      end;           -- de Sub2
    begin           -- de Sub1
      ...
    end;           -- de Sub1
  procedure Sub3 is
    A, X, W : Integer;
    begin           -- de Sub3
      ...
    end;           -- de Sub3
  begin           -- de Main
    ...
  end;           -- de Main

```

**10** Considere o seguinte esqueleto de programa em C. Dada as seguintes sequências de chamadas e assumindo que o escopo dinâmico é usado, que variáveis são visíveis durante a execução da última função chamada? Inclua com cada variável visível o nome da função na qual ela foi definida.

- main chama fun1; fun1 chama fun2; fun2 chama fun3.
- main chama fun1; fun1 chama fun3.
- main chama fun2; fun2 chama fun3; fun3 chama fun1.
- main chama fun3; fun3 chama fun1.
- main chama fun1; fun1 chama fun3; fun3 chama fun1.
- main chama fun3; fun3 chama fun2; fun2 chama fun1.

```

void fun1(void);           //protótipo
void fun2(void);           //protótipo
void fun3(void);           //protótipo
void main(){
    int a, b, c;
    ...
}
void fun1(){
    int b, c, d;
    ...
}
void fun2(){
    int c, d, e;
    ...
}
void fun3(){
    int d, e, f;
    ...
}

```

**11** Considere o seguinte programa em C. Para cada um dos quatro pontos marcados nessa função, liste cada variável visível, com o número da sentença de definição que a define.

```

void fun(void){
    int a, b, c;           //definição 1
    ...
    while (...){
        int b, c, d;       //definição 2
        ... <----- PONTO 1
        while (...){
            int c, d, e;    //definição 3
            ... <----- PONTO 2
        }
        ... <----- PONTO 3
    }
    ... <----- PONTO 4
}

```