

Subprogramas

- Duas formas de abstração são possíveis em linguagens de programação: abstração de processo e abstração de dados.
- Abstração de processos: aparece na forma de subprogramas, que permitem:
 - o reuso de código
 - economia de tempo e de memória.
 - facilitar a leitura do programa (permite ver a estrutura lógica do programa, escondendo detalhes de codificação)

Características dos subprogramas estudados

- Cada subprograma tem um único ponto de entrada
- A unidade chamadora é suspensa durante a execução da unidade chamada – existe um único subprograma em execução a cada momento
- O controle sempre retorna a unidade chamadora quando acaba a execução da unidade chamada
- Outras formas de subprogramas:
 - corotinas
 - unidades concorrentes

- Métodos de POO: semelhantes aos subprogramas
- Principais diferenças:
 - forma de chamada
 - associação com classes e objetos
- Semelhanças:
 - Passagem de parâmetros
 - Variáveis locais

Implementação de subprograma

Ativação de subprograma:

acontece quando o subprograma é chamado.

Registro de Ativação:

formato das informações não-código que devem ser armazenadas durante a ativação.

Instância de registro de Ativação:

coleção particular de dados na forma de RA.

Pilha de Execução:

parte da memória usada como pilha que guarda informações sobre ativações de subprogramas.

Formato de RA X Tamanho de RA

Formato: (layout) determina quais as informações fazem parte do RA

Tamanho: pode mudar quando variáveis locais tem tamanho variável (arrays)

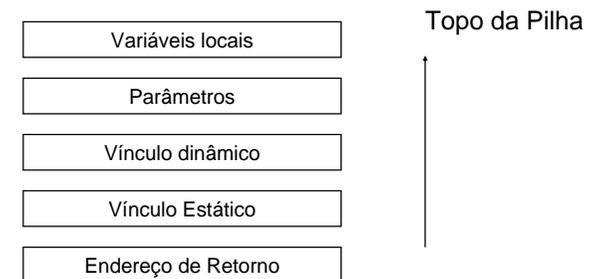
Linguagens de Escopo estático:

Formato estático

Tamanho dinâmico

RA de Linguagens de Programação Estruturada

- Parâmetros passados por dois ou mais métodos
- Variáveis locais alocadas dinamicamente
- implementam recursão (possibilidade de ativações simultâneas múltiplas de um subprograma)
- Escopo estático para variáveis não locais



Endereço de Retorno, Vínculo Estático, Vínculo Dinâmico e Parâmetros - colocados pelo **chamador**
Variáveis Locais: colocadas pelo **chamado**

Parâmetros

- **Endereço de Retorno:**
 - composto de um ponteiro para o segmento de código do chamador e de um deslocamento até a instrução seguinte à chamada .
 - **Vínculo Estático:**
 - (Ponteiro de Escopo Estático) Aponta para a parte inferior do RA do pai estático. Usado para acesso a variáveis não locais.
 - **Vínculo Dinâmico:**
 - Ponteiro para o topo do RA do chamador. Em linguagem de escopo estático é usado na destruição do RA atual, quando termina a execução do procedimento.
- valores ou endereços fornecidos pelo chamador
 - Passagem por valor: valores são copiados para a pilha. Posição é usada como variável
 - Passagem por resultado: Posição usada como variável local. Resultado copiado no parâmetro real.
 - Passagem por valor-resultado: combinação dos dois casos anteriores.
 - Passagem por referência: endereço do parâmetro real é colocado na pilha.

Representação Esquemática de RA

- Pilha de execução cresce de endereços mais baixos para os mais altos
- Todas as variáveis ocupam uma posição na pilha (não é indicado qual o tamanho em bytes de cada dado).
- RA do programa principal está sempre na base e tem apenas as variáveis globais.
- Os valores das variáveis são indicados no espaço reservado a cada variável