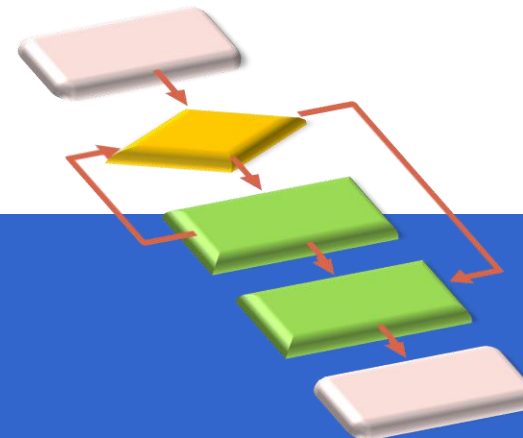




INSTITUTO DE COMPUTAÇÃO



Algoritmos II

ALOCAÇÃO DINÂMICA DE MEMÓRIA

Ponteiros: sintaxe e operadores

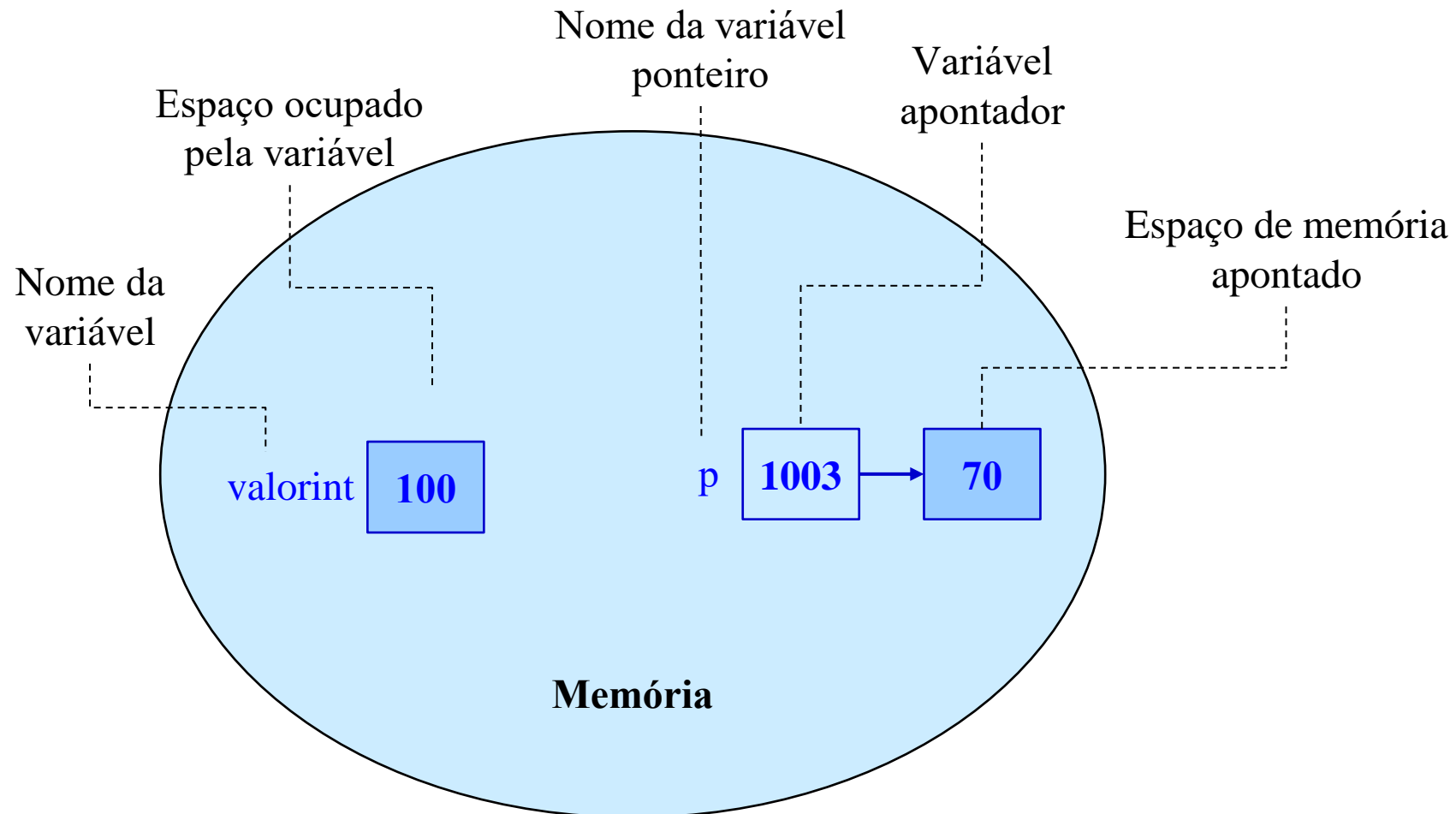
Prof.^a Vanessa de Oliveira Campos

Ponteiros

- **Ponteiros** são variáveis que, em vez de valores de dados, guardam endereços de memória.
 - Quando um programa solicita a alocação de uma variável dinâmica, o endereço da variável dinâmica na memória é armazenado em uma variável do tipo ponteiro.
 - A partir da alocação, é necessário usar o ponteiro correspondente para acessar o valor do dado armazenado na variável dinâmica.



Alocação estática e dinâmica de memória



Ponteiros

- Uma variável do tipo **ponteiro** pode apontar (ou seja, guardar o endereço na memória) para uma área de memória associada a qualquer tipo de dado, simples (inteiro, real, etc.) ou complexo (arranjos, estruturas, etc.).
- Variáveis do tipo **ponteiro**: declaradas no início dos programas.



Ponteiros

- Variáveis dinâmicas:
 - **alocadas a qualquer momento;**
 - espaço **liberado a qualquer momento.**



Declaração de Ponteiro

- Na pseudolinguagem, a declaração de um ponteiro tem a seguinte forma:

<nome da variável>: ↑ <tipo da variável a ser alocada>

Exemplos:

var

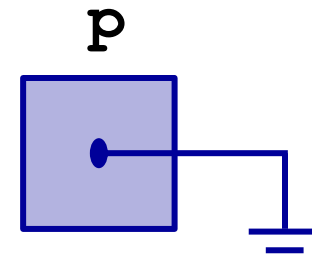
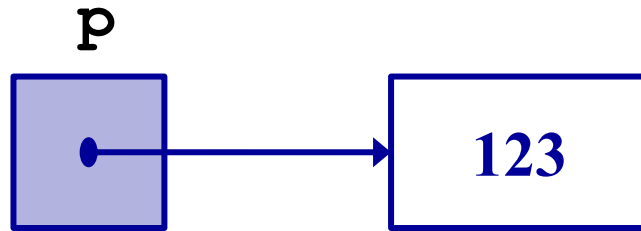
p: ↑ inteiro

pv: ↑ arranjo [1..10] de inteiro



Declaração de Ponteiro

- Representação gráfica de variável do tipo ponteiro



Alocação de memória

- O comando que solicita a alocação de um espaço de memória tem a seguinte forma:

`aloca (<nome de variável do tipo ponteiro>)`

- Exemplo:

`aloca (p)`



Liberação de memória

- Quando o espaço alocado não for mais necessário, ele pode ser liberado por meio do seguinte comando:

`desaloca(<nome de variável do tipo ponteiro>)`

- Por exemplo:

`desaloca(p)`



Acesso a variáveis apontadas por ponteiros

- O acesso a uma variável alocada por um ponteiro é feito por meio do nome declarado para o ponteiro, seguido do símbolo “↑”.

Exemplos:

$p↑ \leftarrow 10$ {atribuir o valor 10 ao espaço de memória apontado por p}

escreva ($p↑$) {informar o valor contido no espaço de memória apontado por p}

escreva ($p↑[3]$) {informar o valor contido no terceiro elemento do arranjo p}



Atribuição de valor a ponteiro

- Endereço **nulo**:

$p \leftarrow \text{nulo}$

Se $p = \text{nulo}$ então

 escreva ('ponteiro nulo')



Atribuição de valor a ponteiro

- Atribuição entre ponteiros:

var

p1, p2: ↑ inteiro

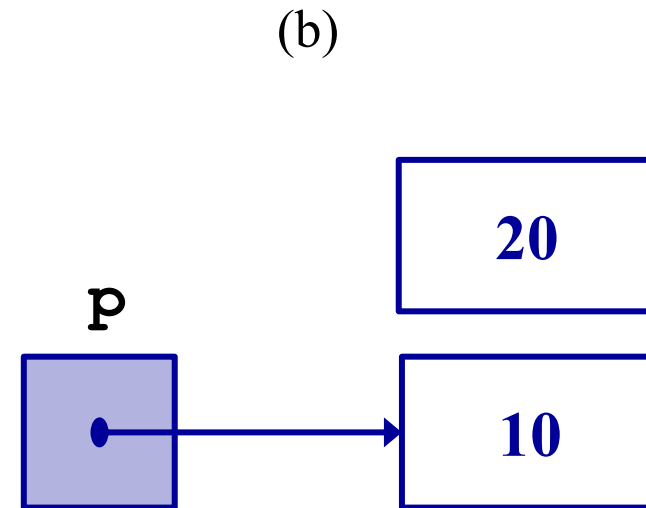
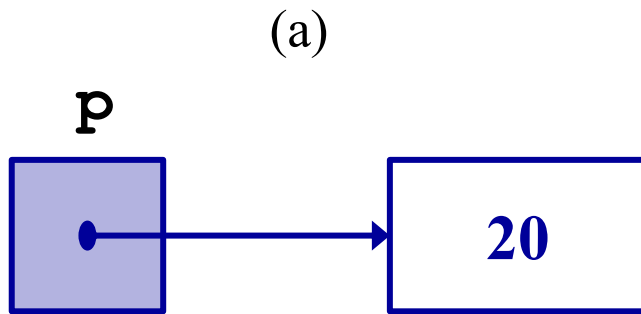
p1 ← nulo

p2 ← p1



Perda de acesso a uma variável

- **Cuidado:** nenhuma variável pode ficar sem acesso!



Perda de acesso a uma variável

var:

p1, p2: \uparrow inteiro

início

aloca (p1)

p2 \leftarrow nulo

p1 \uparrow \leftarrow 20

p1 \leftarrow p2



Perda de acesso a uma variável

var:

p1, p2: ↑ inteiro

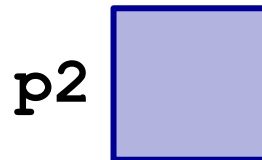
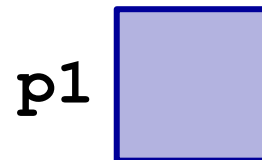
início

aloca (p1)

p2 ← nulo

p1↑ ← 20

p1 ← p2



Perda de acesso a uma variável

var:

p1, p2: ↑ inteiro

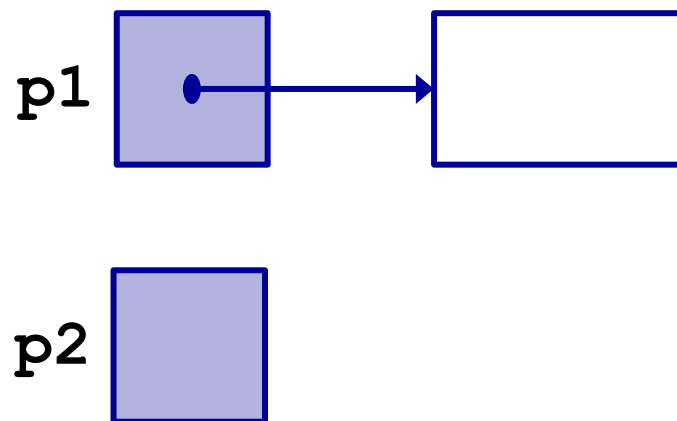
início

aloca (p1)

p2 ← nulo

p1↑ ← 20

p1 ← p2



Perda de acesso a uma variável

var:

p1, p2: ↑ inteiro

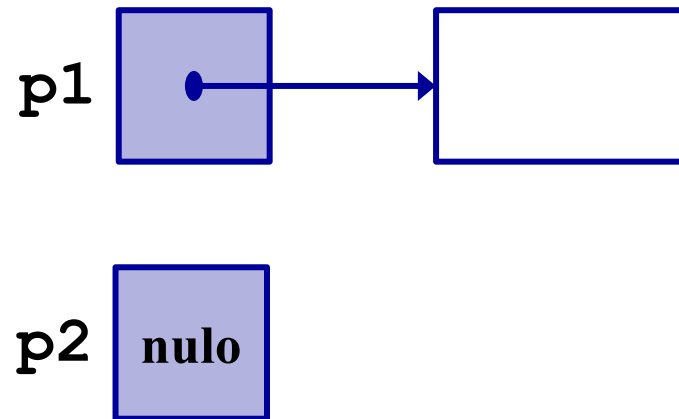
início

aloca (p1)

p2 ← nulo

p1↑ ← 20

p1 ← p2



Perda de acesso a uma variável

var:

p1, p2: ↑ inteiro

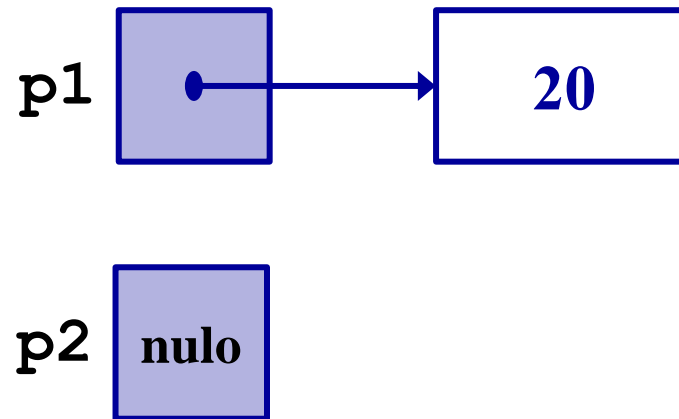
início

aloca (p1)

p2 ← nulo

p1↑ ← 20

p1 ← p2



Perda de acesso a uma variável

var:

p1, p2: ↑ inteiro

início

aloca (p1)

p2 ← nulo

p1↑ ← 20

p1 ← p2

