Computação Eletrônica

Variáveis Inteiras

A memória

- A memória pode ser vista como um imenso gaveteiro
- Em cada gaveta, guardamos uma informação. Por exemplo, um pedaço de papel contendo um número. Ou um pedaço de papel contendo um nome.

Variáveis

- As gavetas em programação são chamadas de variáveis
- Antes de utilizarmos uma variável, devemos nomeá-la.
- Em Pascal os nomes de variáveis seguem a seguinte regra:
  - É formado por uma única letra ou por uma letra seguida de letras ou dígitos
  - Não é permitido espaço em branco ou outros caracteres como @, *, ;, /, etc.
  - Exemplos permitidos: A, Nota, Matricula, LucroTotal
  - Exemplos errados: 5B, X-Y, A:B, Terca-Feira, km/h

Variáveis

- Em Pascal, devemos nomear a gaveta através da declaração de uma variável

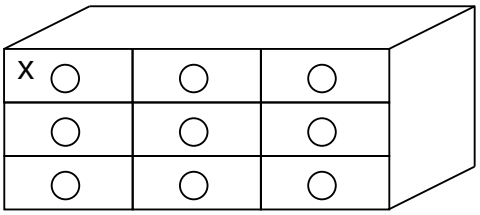
```
Program Variavel;
   var x;
begin
  write('Hello World');
end.
```

- Porém, este programa não compila!
  - “Fatal: Syntax error, : expected but ; found”

Variáveis

- Em Pascal, devemos nomear a gaveta através da declaração de uma variável

```
Program Variavel;
   var x;
begin
  write('Hello World');
end.
```

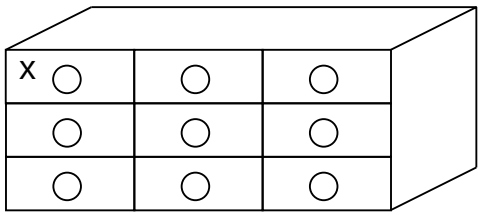


- Porém, este programa não compila!
  - "Fatal: Syntax error, : expected but ; found"

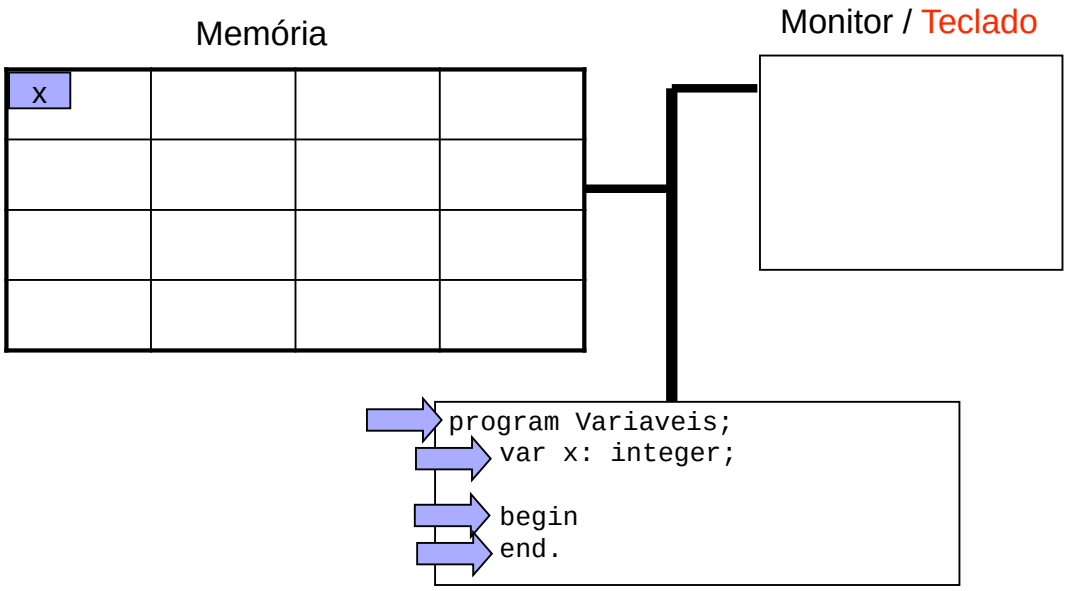
Variáveis

- Computadores trabalham com variáveis de diferentes tipos:
  - Em Pascal: integer, real, boolean, char e String.
  - Vamos trabalhar apenas com inteiros (integer) por enquanto (tipicamente, entre -32769 a +32767)
  - O programa anterior deve ser corrigido para:

```
Program Variavel;
   var x: integer;
begin
  write('Hello World');
end.
```



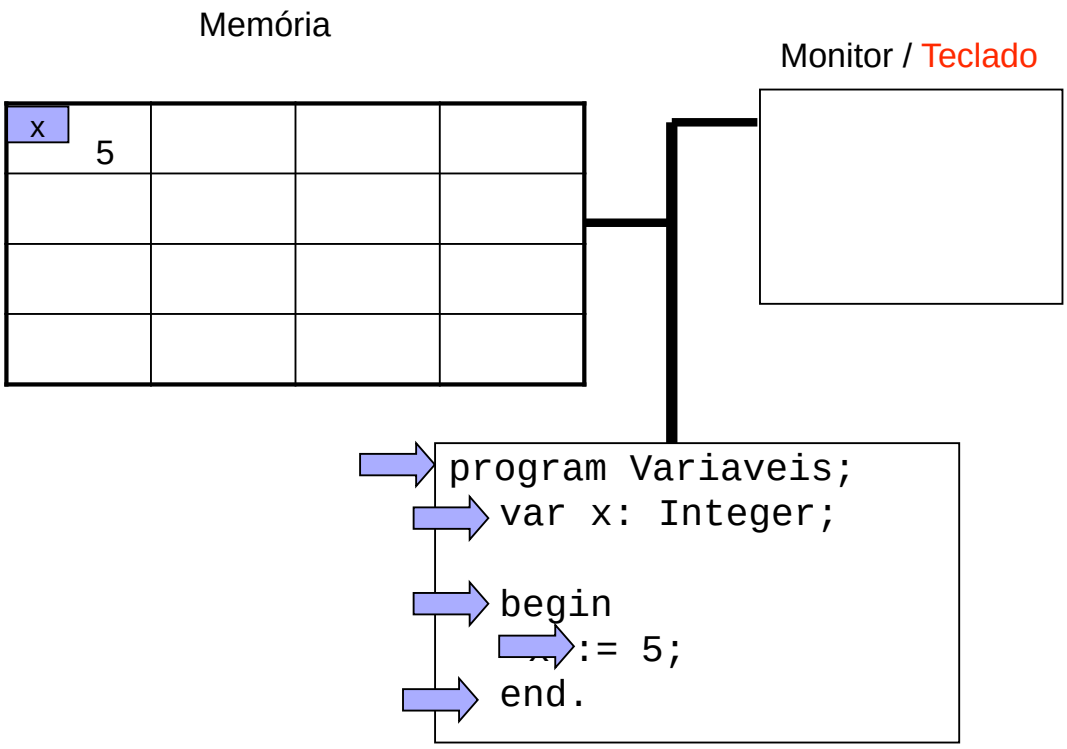
Variáveis



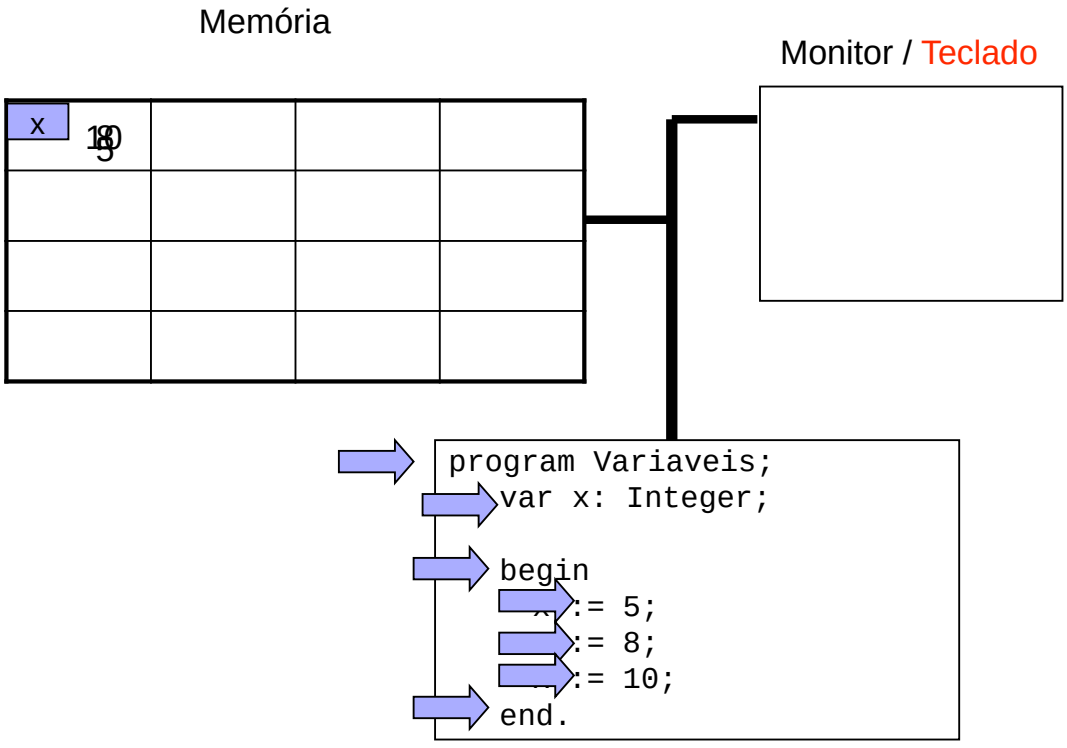
Variáveis

- O programa anterior reserva uma área de memória chamada x, mas não a utiliza. Nenhum número inteiro foi gravado em x.
- Existem 2 modos para gravar um número em uma área de memória
  - Atribuição
  - Função readln

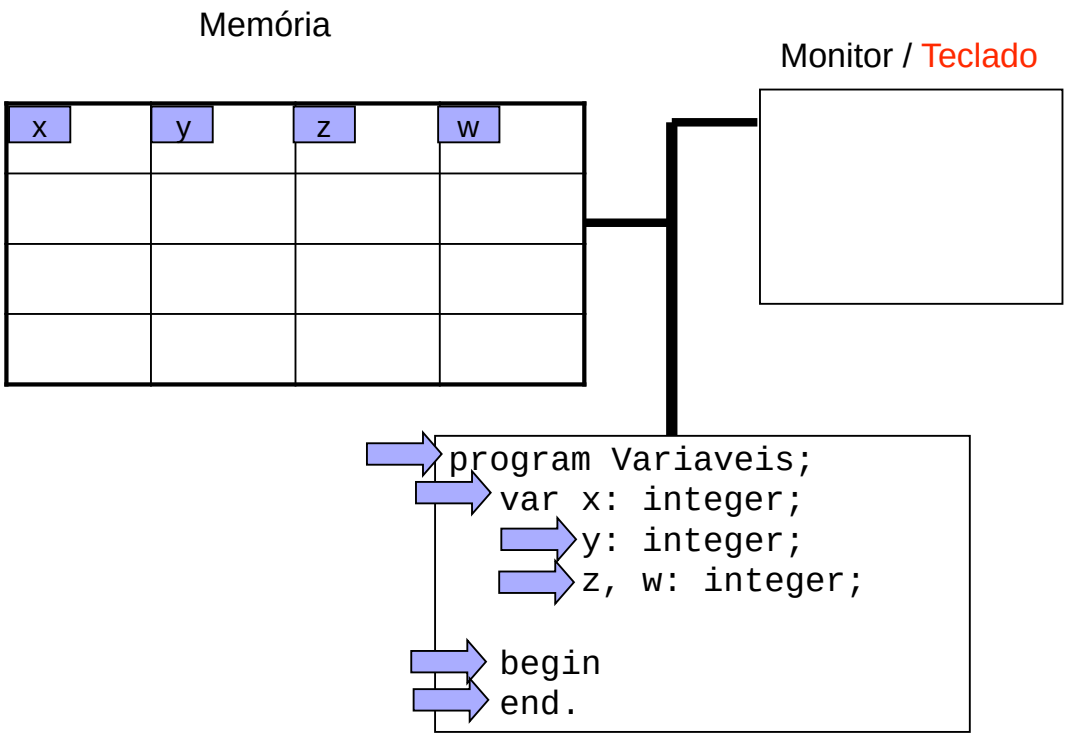
Atribuição



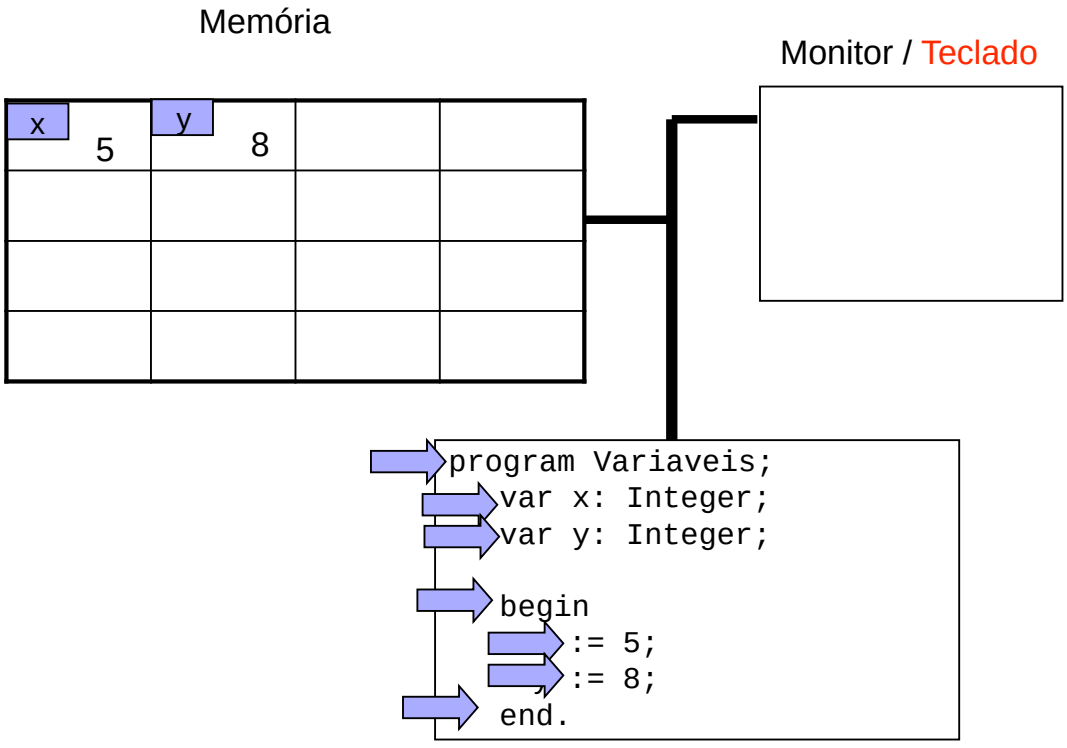
Atribuição



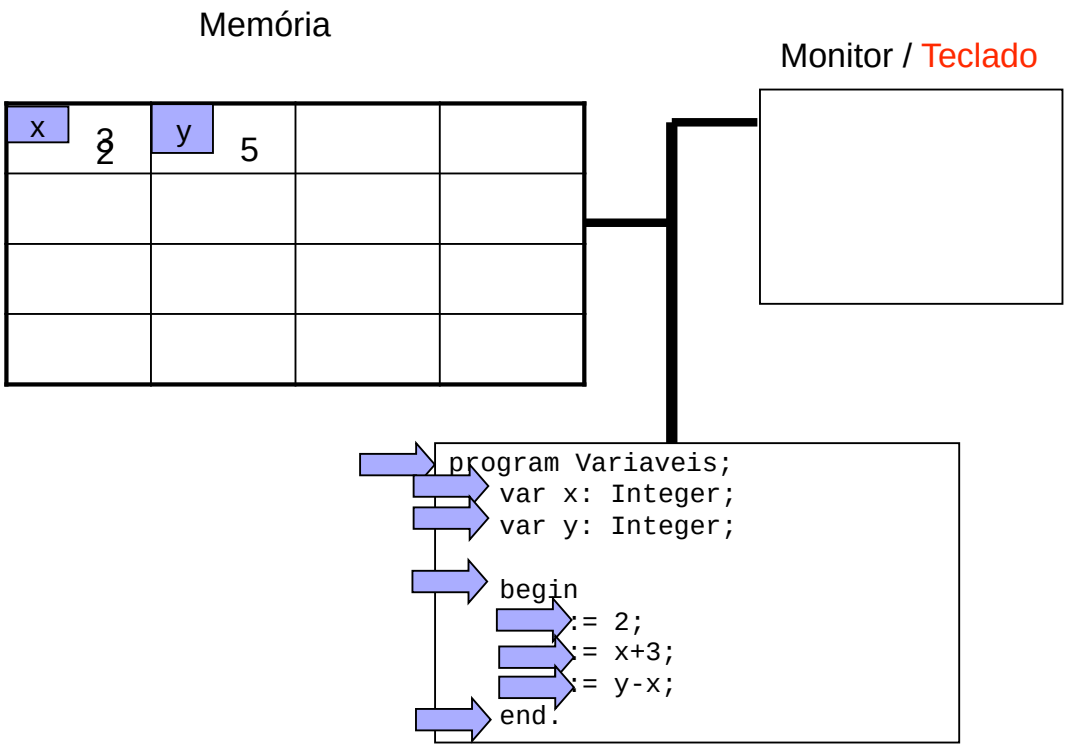
Variáveis



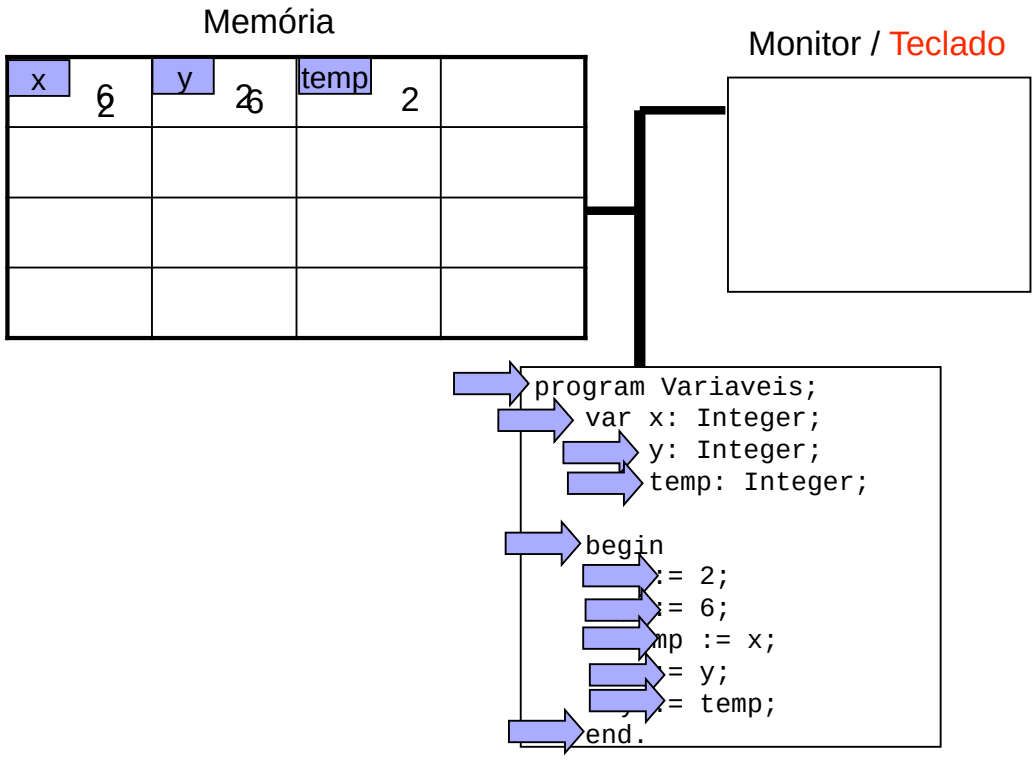
Atribuição



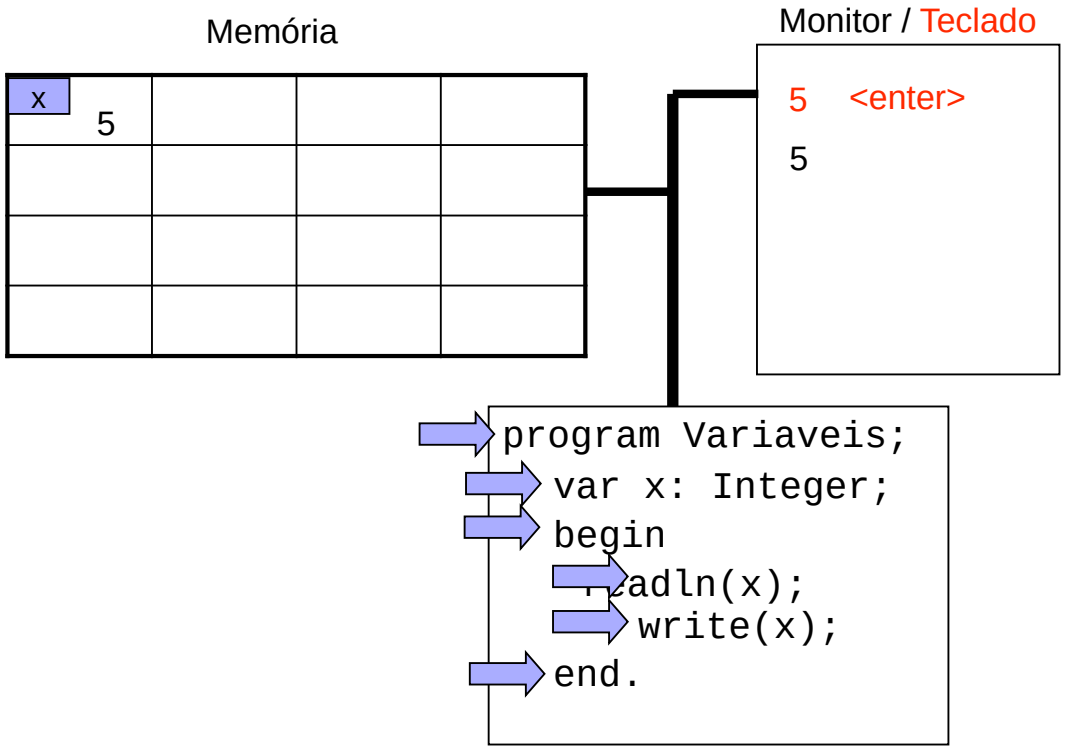
Atribuição



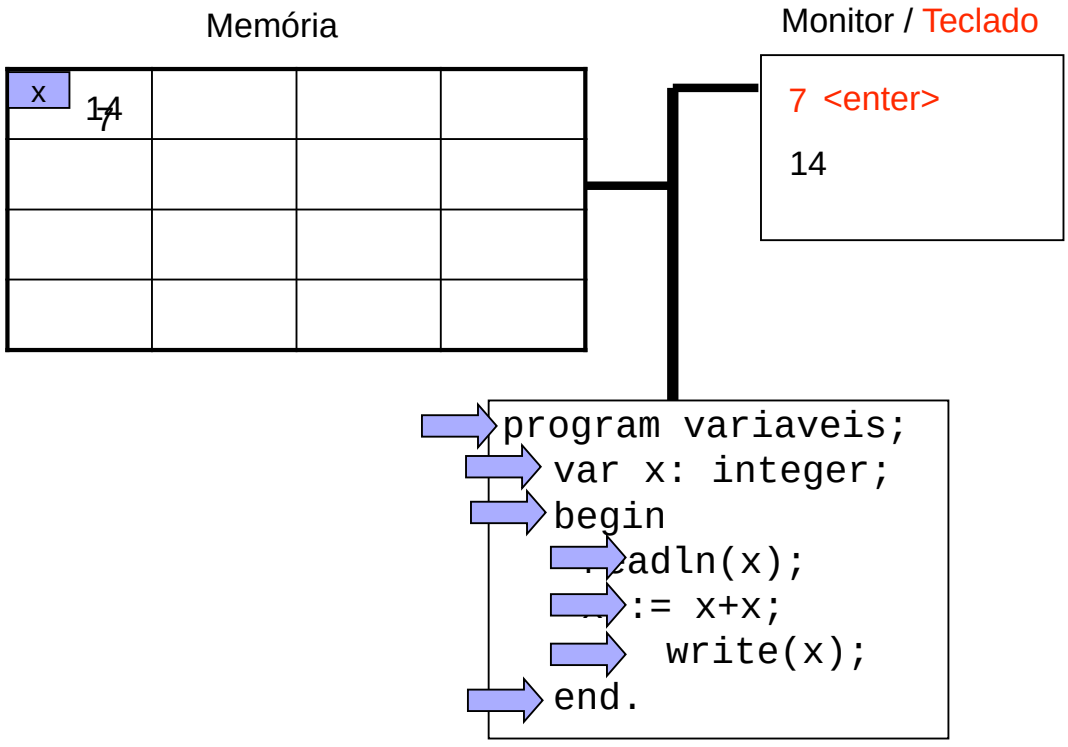
Atribuição



readln



readln

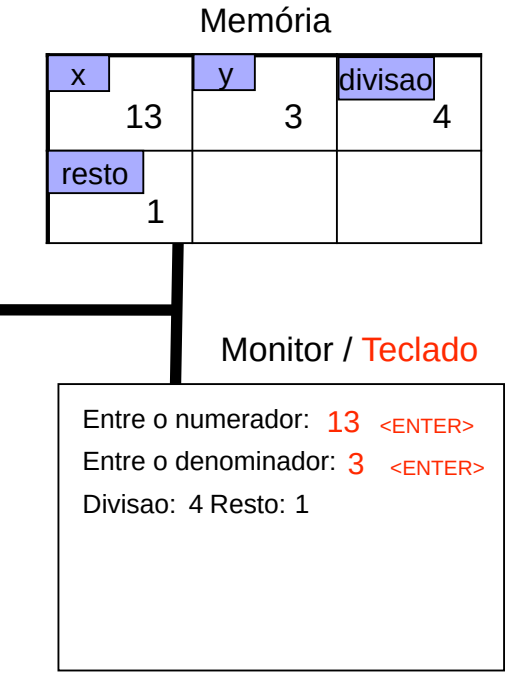


Operadores matemáticos para *inteiros*

- Além de adição (+), subtração (-) e multiplicação (*), Pascal também ofere
  - Divisão: div
    - 15 div 3 = 5
    - 5 div 2 = 2 /* Lembre-se: divisão entre inteiros! */
  - Resto ou módulo da divisão: mod
    - 15 mod 3 = 0
    - 5 div 2 = 1

Um programa maior

```
program divis;
var x,y,divisao,resto: integer;
begin
  write('Entre o numerador: ');
  readln(x);
  write('Entre o denominador: ');
  readln(y);
  divisao := x div y;
  resto := x mod y;
  write('Divisao: ');
  write(divisao);
  write('Resto: ');
  write(resto);
end.
```



Mais sobre write e writeln

```
program divisao2;
   var x,y,divisao: integer;
begin
   write('Entre o numerador: ');
   readln(x);
   write('Entre o denominador: ');
   readln(y);
   divisao := x div y;
   writeln('Divisao: ',divisao);
   write('Resto: ',x mod y);
   readln;
end.
```

Programa que faz a mesma coisa que o anterior.

Mas não usa tantas variáveis.

writeln adiciona uma quebra de linha no final.

write e writeln aceitam tanto texto entre aspas, como 'Divisao: ' quanto variáveis como divisao ou expressões matemáticas como x mod y. Eles tem que vir separados por vírgulas.

Exercício

- Fazer um programa para:
  - Ler, via teclado, um número inteiro. Assuma que o usuário vai digitar um número entre 100 e 999.
  - Imprimir no monitor os dígitos deste número (1 dígito em cada linha).
    - Exemplo. Se o usuário digitar 358, imprimir:

      3

      5

      8

Exercício

- Fazer um programa para:
  - Ler, via teclado, 3 números inteiros x, y e z. Assuma que o usuário vai digitar números entre 100 e 999.
  - Imprimir no monitor a soma dos dígitos destes números.
    - Exemplo. Se o usuário digitar 353, 612 e 999 para x, y e z respectivamente, o programa deve imprimir 11, 9 e 27.