

# Variáveis, Tipos, ES, Operadores

Eduardo Piveta

# Introdução

- Linguagens imperativas são abstrações da arquitetura de von Neumann
  - Memória
    - Armazena instruções e dados
  - Processador
    - Operações para modificar o conteúdo da memória
- As variáveis são, em LPs, abstrações para células de memória em um computador

# Declarações de variáveis e tipos de dados

- Variáveis são usadas para nomear espaços reservados de memória, os quais são alocados para o armazenamento de dados dos programas.
- Linguagens de programação do paradigma imperativo (tais como C, C++, Java, Pascal) usam variáveis como abstração de dados.
  - No entanto, **nem todas as LPs usam variáveis** (as linguagens funcionais puras, por exemplo, não as utilizam).

# Paradigmas de Programação

Category	Ratings Aug 2011	Delta Aug 2010
Object-Oriented Languages	55.3%	+1.2%
Procedural Languages	38.2%	-2.9%
Functional Languages	4.9%	+1.6%
Logical Languages	1.6%	+0.1%

Linguagens Imperativas



# Nomes

- Variáveis possuem alguns atributos, incluindo:
  - Nome, tipo e valor/valores
- Os nomes são utilizados em um contexto mais amplo do que simplesmente em variáveis
  - Nomes pra métodos, sub-programas, pacotes, módulos, classes, parâmetros ...
- O termo *identificador* também é usado como sinônimo de um nome em um programa qualquer

# Nomes > Formatos de Nomes

- Exemplos:
  - Fortran95: 31 caracteres
  - C89: sem limitação de tamanho (apenas os 31 primeiros são significativos)
  - C99: 61 primeiros significativos
  - Java, C#, Ada, C++: sem limite
    - Os implementadores podem colocar um limite
- Normalmente o limite é dado para reduzir o tamanho da tabela de símbolos
- Na maioria das linguagens o formato é:
  - letra\_ (letra\_ | dígito)\*

# Nomes > Formatos de Nomes

- Algumas linguagens permitem espaços nos nomes das variáveis
  - Ex.: Soma dos Salarios, Variavel de teste (anteriores ao FORTRAN 90)
- Linguagens baseadas em C diferenciam minúsculas de maiúsculas
  - ROSE != Rose != rose
  - Pode levar a problemas de legibilidade e facilidade de escrita
    - parseInt != parseint != ParseInt
  - Uso de convenções de estilo

# Variáveis > Tipo

- O tipo de uma variável determina a faixa de valores que a variável pode armazenar
- Determina também as operações que estão disponíveis para os valores do tipo
  - Exemplo: inteiros em Java
  - Faixa: de -2147483648 a 2147483648
  - Operações: adição, subtração, multiplicação, divisão e módulo.



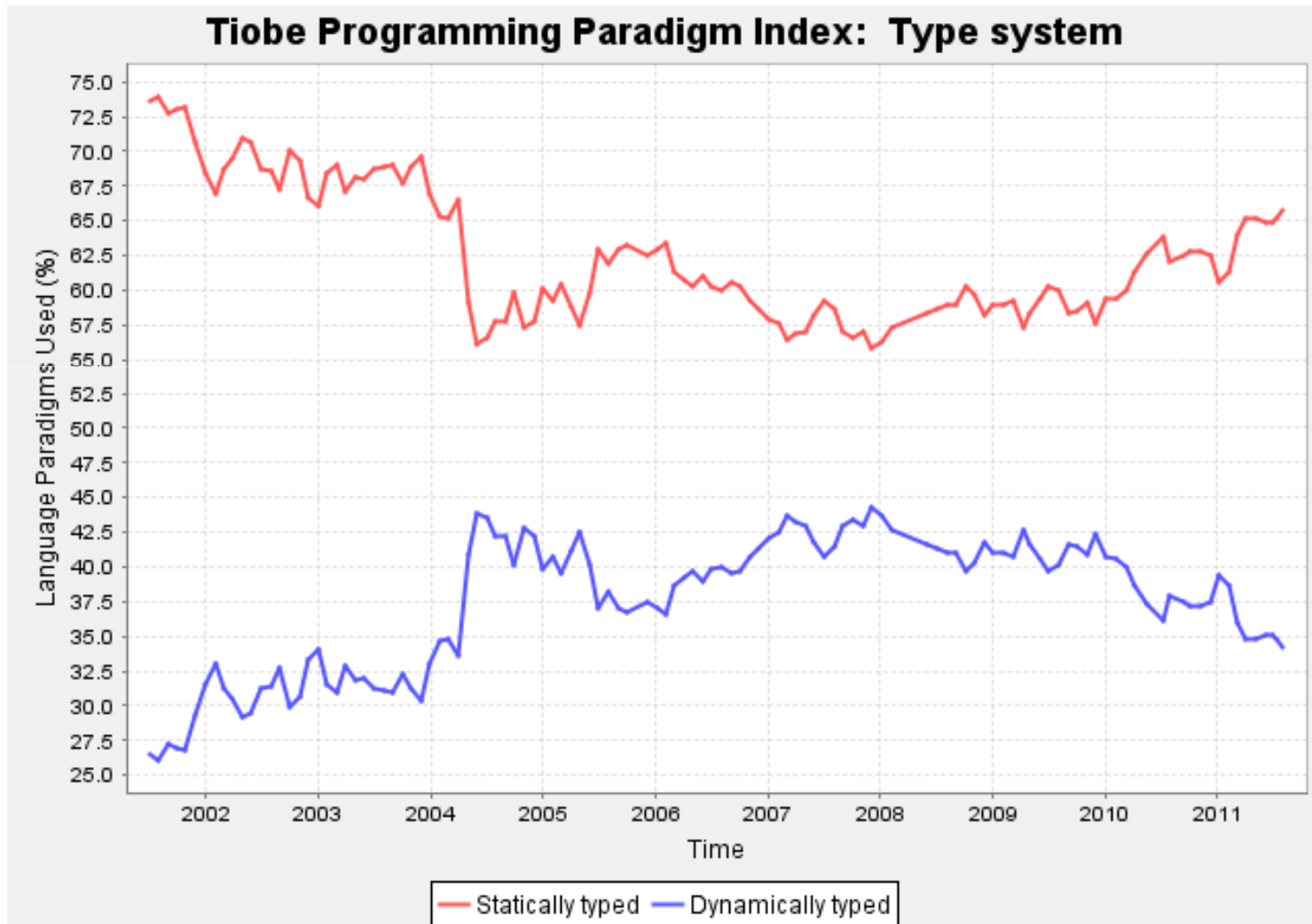
# Variáveis > Valor

- O valor de uma variável é o conteúdo das células de memórias associadas com a variável
  - Tais células de memória são abstratas

# Tipos de Dados

- Descrevem os possíveis valores e operações que uma variável pode conter/executar.
- Boa parte das linguagens é fortemente tipada (os tipos precisam ser explicitamente declarado) – Java, C, C++, etc.
  - No entanto, existem diversas linguagens nas quais o tipo é dinâmico (Smalltalk, Ruby, Python, etc).

# Tipagem Estática vs. Dinâmica



# Tipos de Dados

- Os principais tipos de dados representam:
  - Caracteres
  - Inteiros
  - Valores de ponto flutuante
  - Booleanos
  - Referências
  - Cadeias de Caracteres (Strings)
  - Vetores e Matrizes
  - Tipos Definidos Pelo Usuário

# Exemplo - Tipos de Dados: C (inteiros)

Specifications for standard integer types

Shortest form of specifier	Minimum width (bits)
<code>_Bool<sup>[1]</sup></code>	1
<code>char<sup>[1]</sup></code>	8
<code>signed char</code>	8
<code>unsigned char</code>	8
<code>short</code>	16
<code>unsigned short</code>	16
<code>int</code>	16
<code>unsigned</code>	16
<code>long</code>	32
<code>unsigned long</code>	32
<code>long long<sup>[1]</sup></code>	64
<code>unsigned long long<sup>[1]</sup></code>	64

# Exemplo - Tipos de Dados: Pascal

Data type	Type of values which the variable is capable of storing
integer	Whole numbers
real	Floating point numbers
boolean	The value TRUE or FALSE
char	A single character from an ordered character set

# Tipos de Dados: Java

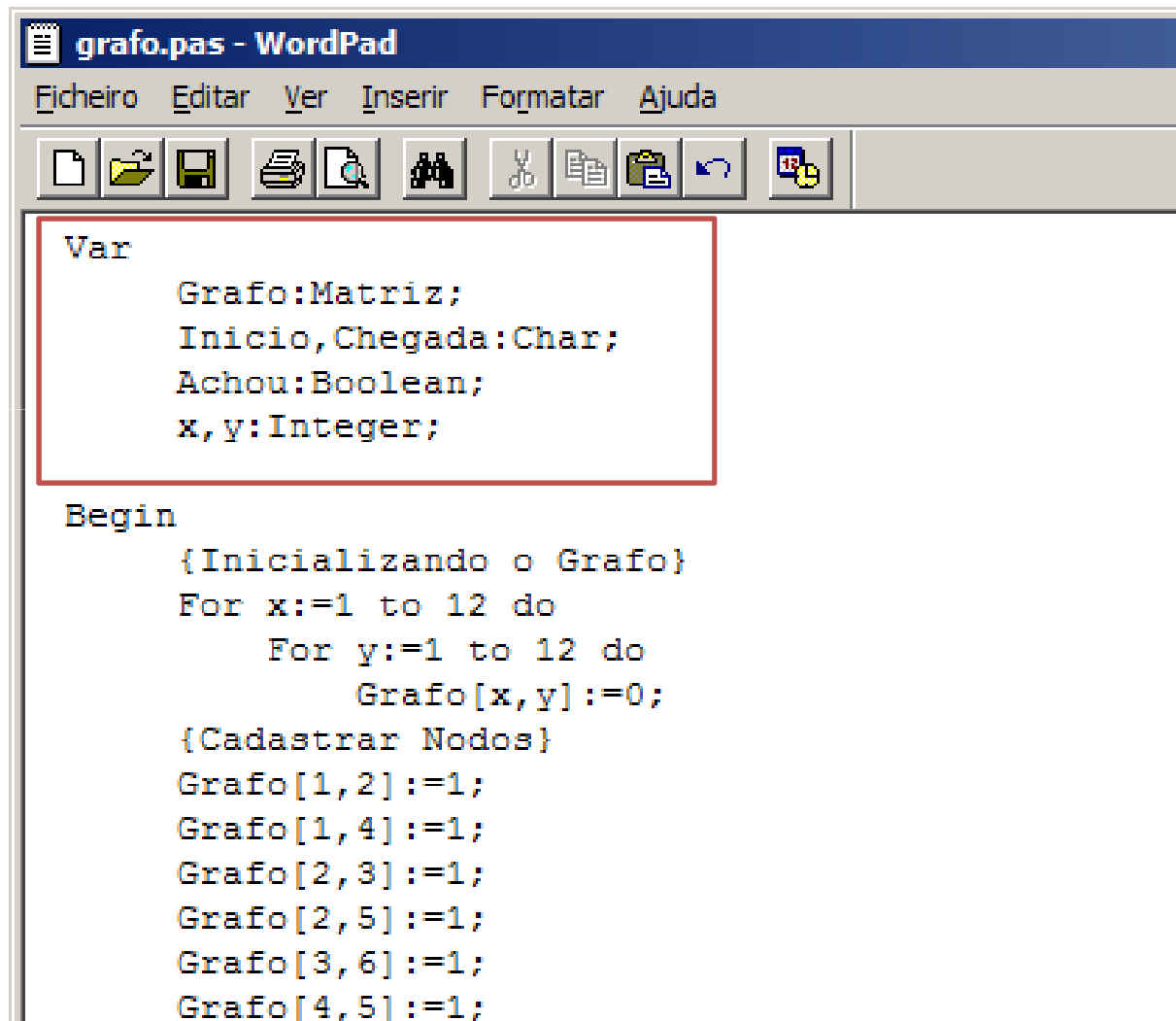
- **byte:** The `byte` data type is an 8-bit signed two's complement integer. The range of the `byte` data type is from `-128` to `127` (inclusive). The `byte` data type can be useful for saving memory in large [arrays](#). The `byte` data type is also useful for working with numbers that are known to be in the range of `-128` to `127`. The `byte` data type is a good choice for the `enum` type. The `byte` data type is also useful for working with numbers that are known to be in the range of `-128` to `127`. The `byte` data type is a good choice for the `enum` type. The `byte` data type is also useful for working with numbers that are known to be in the range of `-128` to `127`. The `byte` data type is a good choice for the `enum` type.
- **short:** The `short` data type is a 16-bit signed two's complement integer. The range of the `short` data type is from `-32,768` to `32,767` (inclusive). The `short` data type is a good choice for the `enum` type. The `short` data type is also useful for working with numbers that are known to be in the range of `-32,768` to `32,767`. The `short` data type is a good choice for the `enum` type.
- **int:** The `int` data type is a 32-bit signed two's complement integer. The range of the `int` data type is from `-2,147,483,648` to `2,147,483,647` (inclusive). For integral values, this data type is generally the best choice. The `int` data type is a good choice for the `enum` type. The `int` data type is also useful for working with numbers that are known to be in the range of `-2,147,483,648` to `2,147,483,647`. The `int` data type is a good choice for the `enum` type.
- **long:** The `long` data type is a 64-bit signed two's complement integer. The range of the `long` data type is from `-9,223,372,036,854,775,807` (inclusive) to `9,223,372,036,854,775,807` (inclusive). Use this data type for integral values that are outside the range of the `int` data type. The `long` data type is a good choice for the `enum` type.
- **float:** The `float` data type is a single-precision 32-bit IEEE 754 floating point number. The range of the `float` data type is from `-3.4028235E+38` to `3.4028235E+38` (inclusive). The `float` data type is a good choice for the `enum` type. The `float` data type is also useful for working with numbers that are known to be in the range of `-3.4028235E+38` to `3.4028235E+38`. The `float` data type is a good choice for the `enum` type.
- **double:** The `double` data type is a double-precision 64-bit IEEE 754 floating point number. The range of the `double` data type is from `-1.7976931348623157E+308` to `1.7976931348623157E+308` (inclusive). The `double` data type is a good choice for the `enum` type. The `double` data type is also useful for working with numbers that are known to be in the range of `-1.7976931348623157E+308` to `1.7976931348623157E+308`. The `double` data type is a good choice for the `enum` type.
- **boolean:** The `boolean` data type has only two possible values: `true` and `false`. The `boolean` data type represents one bit of information, but its "size" is not specified in the Java Language Specification. The `boolean` data type is a good choice for the `enum` type.
- **char:** The `char` data type is a single 16-bit Unicode character. The range of the `char` data type is from `0` to `65,535` (inclusive). The `char` data type is a good choice for the `enum` type.

# Algoritmos: Tipos de Dados

- Os tipos que usaremos inicialmente são:
  - inteiro
  - real
  - caractere
  - booleano
  - string



# Exemplo: Variáveis em Pascal



```
grafo.pas - WordPad
Ficheiro  Editar  Ver  Inserir  Formatar  Ajuda

Var
    Grafo:Matriz;
    Inicio,Chegada:Char;
    Achou:Boolean;
    x,y:Integer;

Begin
    {Inicializando o Grafo}
    For x:=1 to 12 do
        For y:=1 to 12 do
            Grafo[x,y]:=0;
    {Cadastrar Nodos}
    Grafo[1,2]:=1;
    Grafo[1,4]:=1;
    Grafo[2,3]:=1;
    Grafo[2,5]:=1;
    Grafo[3,6]:=1;
    Grafo[4,5]:=1;
```

# Exemplo: Variáveis em C, C++

```
int _tmain(int argc, _TCHAR* argv[])
{
    srand(time(NULL));

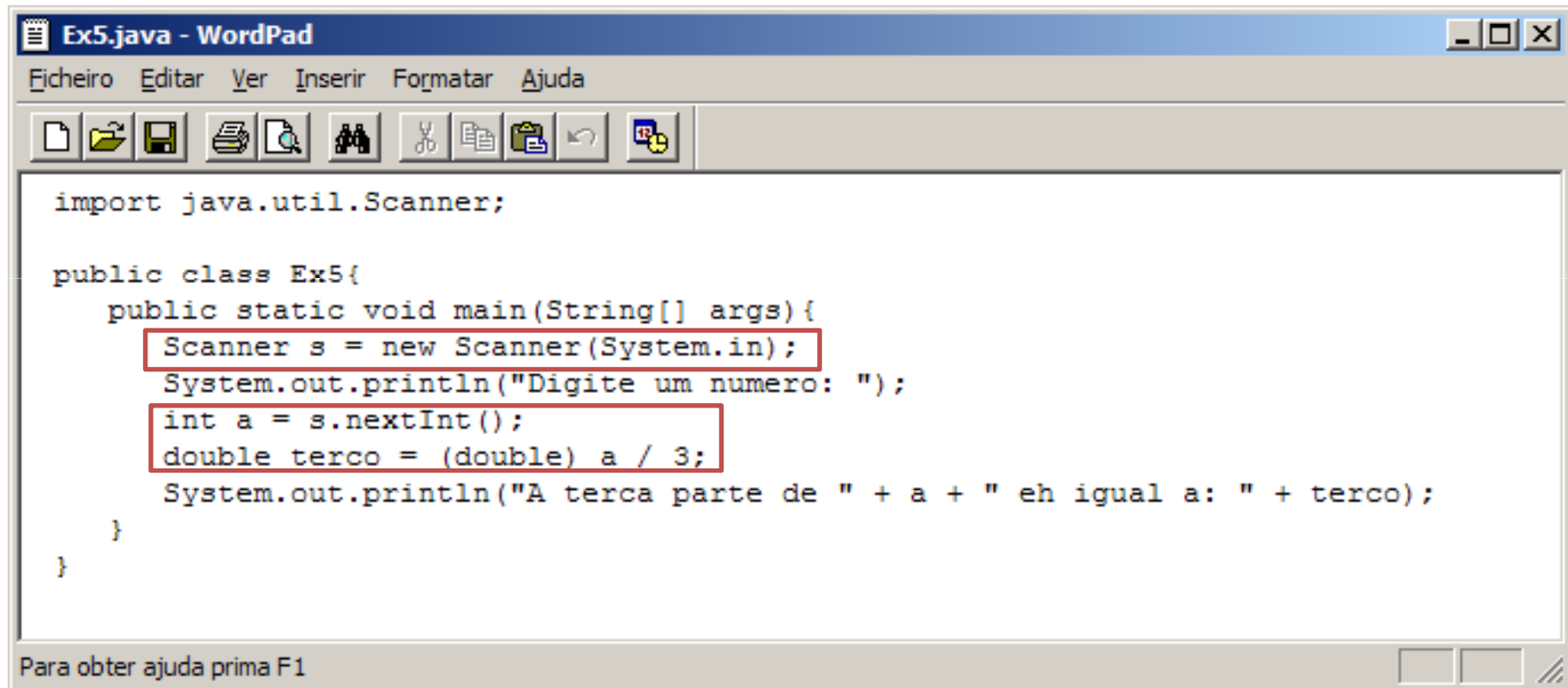
    //define numero de linhas, colunas, e barcos
    int linhas=7, colunas=7, sub=2, destro=3, cruzad=3;
    batalha *novo=cria(linhas, colunas);

    novo->matrizp1=aloca(novo,sub, destro, cruzad);
    novo->matrizp2=aloca(novo,sub, destro, cruzad);
    novo->n_navios1=(sub*2)+(destro*3)+(cruzad*4);
    novo->n_navios2=(sub*2)+(destro*3)+(cruzad*4);

    //inicialização de variaveis de jogo

    int rodada=1;
    int jogador=0;
```

# Exemplo: Variáveis em Java



```
import java.util.Scanner;

public class Ex5{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("Digite um numero: ");
        int a = s.nextInt();
        double terco = (double) a / 3;
        System.out.println("A terca parte de " + a + " eh igual a: " + terco);
    }
}
```

Para obter ajuda prima F1

# Algoritmos: Declaração de Variáveis

- Sintaxe:

<nome> [, <nome>]: <tipo>

- Exemplos:

i: inteiro

r: real

i, j, k: inteiro

c: caractere

b1, b2: booleano

- As LPs permitem também uma inicialização...

# Atribuição

- Uma vez que as variáveis tenham sido declaradas, deve existir algum mecanismo para atribuir valores a tais variáveis.
  - Tal mecanismo é chamado de atribuição.
- Na atribuição, a célula de memória representada pela variável recebe o valor expresso na atribuição...
  - ...que pode ser uma expressão, uma variável, uma constante, um literal, etc.

# Atribuição em LPs

Representações textuais comuns do operador de atribuição incluem um sinal de igual ("=") e ":=". Estas duas formas são típicas das linguagens de programação (tais como C), que classificam o operador de atribuição como um operador infixo.

`variável = expressão` *BASIC, Fortran<sup>[1]</sup>, C, Java, PL/I, Windows PowerShell, Bourne shell, ...*

`variável := expressão` *ALGOL, Pascal<sup>[2]</sup>, Ada<sup>[3]</sup>, Dylan<sup>[4]</sup>, Eiffel<sup>[5][6]</sup> ...*

`variável << expressão` *Magik*

Outras possibilidades incluem uma seta para a esquerda ou uma palavra-chave, embora existam outras, variantes mais raras:

`variável <- expressão` *Objective Caml, S, R, ...*

`variável ← expressão` *APL<sup>[7][8]</sup>*

`LET variável = expressão` *BASIC*

`set variável to expressão` *AppleScript*

`set variável = expressão` *C shell*

`Set-Variable variável (expressão)` *Windows PowerShell*

`variável : expressão` *Macsyma, Maxima*

`val variável = expressão` *ML<sup>[9]</sup>*

Fonte: [http://pt.wikipedia.org/wiki/Atribui%C3%A7%C3%A3o\\_%28computa%C3%A7%C3%A3o%29](http://pt.wikipedia.org/wiki/Atribui%C3%A7%C3%A3o_%28computa%C3%A7%C3%A3o%29)

# Atribuição em LPs

Algumas linguagens funcionais como Lisp, Common Lisp<sup>[11][12]</sup> e Scheme<sup>[13][14]</sup> tem funções como uma forma genérica de atribuir valores para estruturas de dados, para espelhar os *left-values* de outros linguagens:

`(setq variável expressão )`    *Lisp, Common Lisp*

`(set! variável expressão )`    *Scheme*

`(setf variável expressão )`    *Common Lisp*

Fonte: [http://pt.wikipedia.org/wiki/Atribui%C3%A7%C3%A3o\\_%28computa%C3%A7%C3%A3o%29](http://pt.wikipedia.org/wiki/Atribui%C3%A7%C3%A3o_%28computa%C3%A7%C3%A3o%29)

# Algoritmos: Atribuição

- Sintaxe:

`<var> := <expressão>`

- Exemplos:

`a := 1`

`x := 20 + (b * c)/2`

`y := calcularMedia(a, b, c);`

`b := verdadeiro`

`f := 21,5`

`c := "a"`

`s := "Maria da Silva"`



# Leitura

- Para que possamos fazer aplicações úteis, existe a necessidade de lermos dados.
  - Tais dados são comumente informados pelo usuário através do teclado.
- A forma de leitura depende do cliente usado:
  - console, web, desktop, dispositivos móveis
- As linguagens de programação comumente fornecem mecanismos padronizados para tal.

# Leitura em C

## Example

```
1 /* scanf example */
2 #include <stdio.h>
3
4 int main ()
5 {
6     char str [80];
7     int i;
8
9     printf ("Enter your family name: ");
10    scanf ("%s",str);
11    printf ("Enter your age: ");
12    scanf ("%d",&i);
13    printf ("Mr. %s , %d years old.\n",str,i);
14    printf ("Enter a hexadecimal number: ");
15    scanf ("%x",&i);
16    printf ("You have entered %#x (%d).\n",i,i);
17
18    return 0;
19 }
```

<http://www.cplusplus.com/reference/cstdio/scanf/>

# Leitura em C++

```
1 // i/o example
2
3 #include <iostream>
4 using namespace std;
5
6 int main ()
7 {
8     int i;
9     cout << "Please enter an integer value: ";
10    cin >> i;
11    cout << "The value you entered is " << i;
12    cout << " and its double is " << i*2 << ".\n";
13    return 0;
14 }
```

[http://www.cplusplus.com/doc/tutorial/basic\\_io/](http://www.cplusplus.com/doc/tutorial/basic_io/)

# Leitura em Pascal

## CODE

```
program swappingVariables;

var
    a, b, temp: integer;

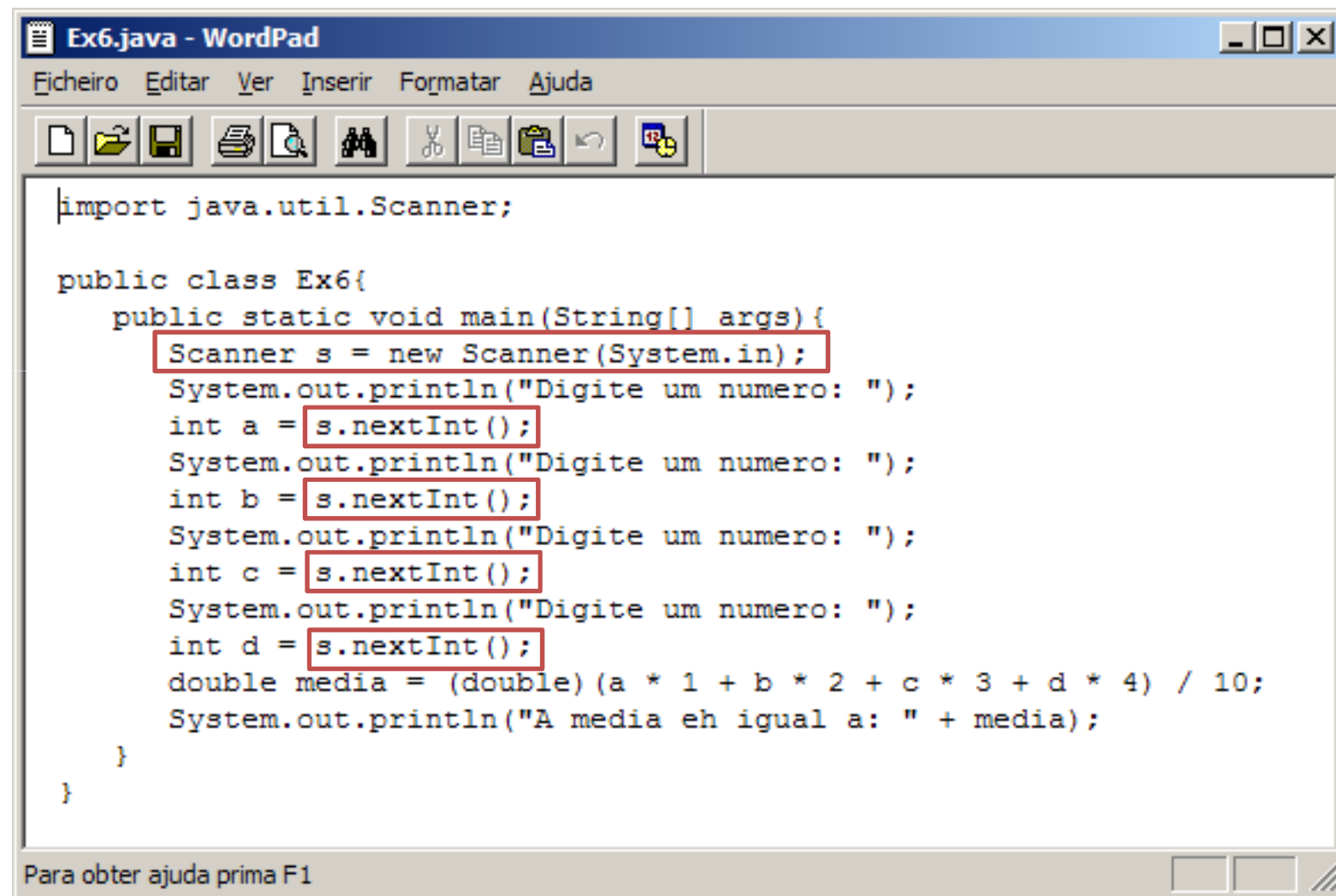
begin
    write('Please enter two integer numbers: ');
    read(a);
    read(b);

    temp:=a;
    a:=b;    // we have "temp" so with don't lose the value contained in "a" in this line
    b:=temp;

end.
```

[http://www.astahost.com/info.php/Pascal-Beginners-Part\\_t15281.html](http://www.astahost.com/info.php/Pascal-Beginners-Part_t15281.html)

# Leitura em Java



```
Ex6.java - WordPad
Ficheiro  Editar  Ver  Inserir  Formatar  Ajuda

import java.util.Scanner;

public class Ex6{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("Digite um numero: ");
        int a = s.nextInt();
        System.out.println("Digite um numero: ");
        int b = s.nextInt();
        System.out.println("Digite um numero: ");
        int c = s.nextInt();
        System.out.println("Digite um numero: ");
        int d = s.nextInt();
        double media = (double) (a * 1 + b * 2 + c * 3 + d * 4) / 10;
        System.out.println("A media eh igual a: " + media);
    }
}
```

Para obter ajuda prima F1

# Algoritmos: Leitura

- Sintaxe:

ler <variável> [, <variável>]

- Exemplos:

ler a

ler b

ler x, y, z

ler y

# Escrita

- Assim como existe a necessidade de ler valores, existe a necessidade de escrever valores na tela...
  - ...sejam constantes do tipo string (textos) quanto valores de variáveis e de expressões.

# Escrita em C

## Example

```
1 /* scanf example */
2 #include <stdio.h>
3
4 int main ()
5 {
6     char str [80];
7     int i;
8
9     printf ("Enter your family name: ");
10    scanf ("%s",str);
11    printf ("Enter your age: ");
12    scanf ("%d",&i);
13    printf ("Mr. %s , %d years old.\n",str,i);
14    printf ("Enter a hexadecimal number: ");
15    scanf ("%x",&i);
16    printf ("You have entered %#x (%d).\n",i,i);
17
18    return 0;
19 }
```

<http://www.cplusplus.com/reference/clibrary/cstdio/scanf/>



# Escrita em C++

```
1 // i/o example
2
3 #include <iostream>
4 using namespace std;
5
6 int main ()
7 {
8     int i;
9     cout << "Please enter an integer value: ";
10    cin >> i;
11    cout << "The value you entered is " << i;
12    cout << " and its double is " << i*2 << ".\n";
13    return 0;
14 }
```

[http://www.cplusplus.com/doc/tutorial/basic\\_io/](http://www.cplusplus.com/doc/tutorial/basic_io/)

# Escrita em Pascal

## CODE

```
program swappingVariables;

var
    a, b, temp: integer;

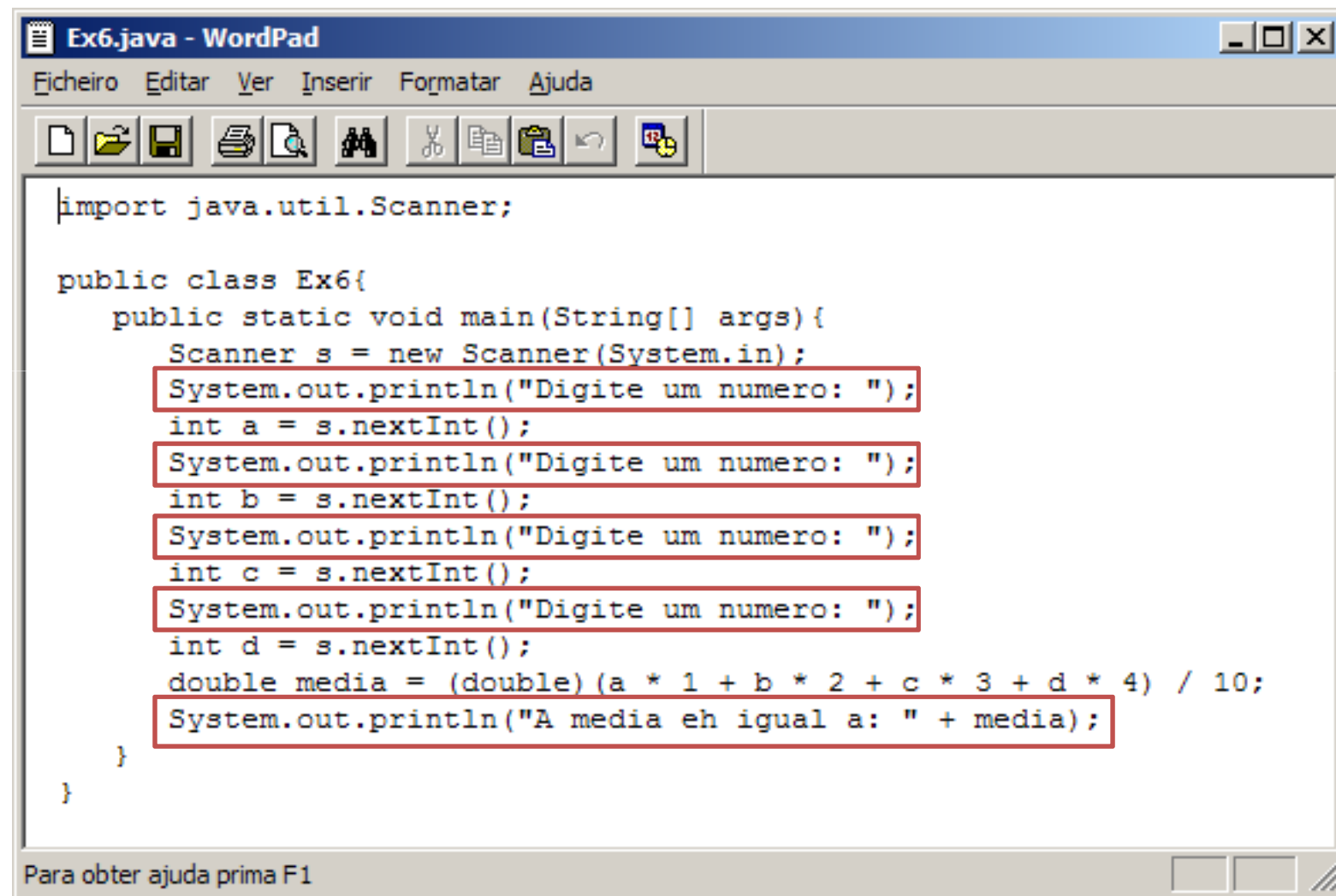
begin
    write('Please enter two integer numbers: ');
    read(a);
    read(b);

    temp:=a;
    a:=b;    // we have "temp" so with don't lose the value contained in "a" in this line
    b:=temp;

end.
```

[http://www.astahost.com/info.php/Pascal-Beginners-Part\\_t15281.html](http://www.astahost.com/info.php/Pascal-Beginners-Part_t15281.html)

# Escrita em Java



```
import java.util.Scanner;

public class Ex6{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("Digite um numero: ");
        int a = s.nextInt();
        System.out.println("Digite um numero: ");
        int b = s.nextInt();
        System.out.println("Digite um numero: ");
        int c = s.nextInt();
        System.out.println("Digite um numero: ");
        int d = s.nextInt();
        double media = (double) (a * 1 + b * 2 + c * 3 + d * 4) / 10;
        System.out.println("A media eh igual a: " + media);
    }
}
```

Para obter ajuda prima F1

# Algoritmos: Escrita

- Sintaxe:

escrever <expressão>

- Exemplos:

escrever "Digite um número:"

escrever x

escrever  $x * y$

escrever "O valor de x é: " + x

# Operadores matemáticos (+, -, \*, /, \*\*, //, %)

- Inicialmente, usaremos alguns operadores matemáticos em nossos algoritmos:
  - Soma: +
  - Subtração: -
  - Multiplicação: \*
  - Divisão: /
  - Potenciação: \*\*
  - Radiciação: //
  - Resto: %

# Outros Operadores

- Serão necessários, ao longo da disciplina, outros operadores:
  - Operadores relacionais ( $=$ ,  $\neq$ ,  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ )
  - Operadores lógicos ( $e$ ,  $ou$ ,  $não$ )
- À medida que eles forem necessários, eles serão discutidos

# Unindo as peças...

- As construções vistas em aula hoje podem ser unidas para a construção de algoritmos...
- ...veremos agora alguns exemplos da utilização dessas construções.

# Exemplo 1

- Escreva na tela Hello World



# Exemplo 1

- Escreva na tela Hello World  
`escrever` “Hello World”

## Exemplo 2

- Declare uma variável inteira x, com valor inicial igual a 10. Imprima o valor de x.

## Exemplo 2

- Declare uma variável inteira  $x$ , com valor inicial igual a 10. Imprima o valor de  $x$ .

$x$ : inteiro

$x := 10$

escrever “O valor de  $x$  é ” +  $x$

## Exemplo 2

- Declare uma variável inteira x, com valor inicial igual a 10. Imprima o valor de x.

x: inteiro

x := 10

escrever "O valor de x é " + x

## Exemplo 2

- Declare uma variável inteira x, com valor inicial igual a 10. **Imprima o valor de x.**

x: inteiro

x := 10

**escrever “O valor de x é ” + x**

## Exemplo 3

- Leia três números e mostre a média entre eles.

# Exemplo 3

- Leia três números e mostre a média entre eles.

a, b, c: **inteiro**

**escrever** "Digite o 1º número: "

**ler** a

**escrever** "Digite o 2º número: "

**ler** b

**escrever** "Digite o 3º número: "

**ler** c

media: **real**

media := (a + b + c) / 3

**escrever** "A média entre " + a + " , " + b + " e " + c + " é " + media

# Exercícios

- Façam os exercícios que estão na página da disciplina.



# Uso de LPs (TIOBE Ranking)

Position Aug 2011	Position Aug 2010	Delta in Position	Programming Language	Ratings Aug 2011	Delta Aug 2010	Status
1	1	=	Java	19.409%	+1.42%	A
2	2	=	C	17.390%	-0.48%	A
3	3	=	C++	8.433%	-1.23%	A
4	4	=	PHP	6.134%	-3.05%	A
5	6	↑	C#	6.042%	+1.06%	A
6	9	↑↑↑	Objective-C	5.494%	+2.34%	A
7	5	↓↓↓	(Visual) Basic	5.013%	-0.40%	A
8	7	↓	Python	3.415%	-0.81%	A
9	8	↓	Perl	2.315%	-1.11%	A
10	11	↑	JavaScript	1.557%	-0.84%	A
11	23	↑↑↑↑↑↑↑↑	Lua	1.362%	+0.83%	A
12	12	=	Ruby	1.329%	-0.65%	A
13	10	↓↓↓	Delphi/Object Pascal	1.076%	-1.35%	A
14	16	↑↑	Lisp	0.905%	+0.28%	A
15	22	↑↑↑↑↑↑↑	Transact-SQL	0.823%	+0.27%	A-

# Variáveis, Tipos, ES, Operadores

Eduardo Piveta