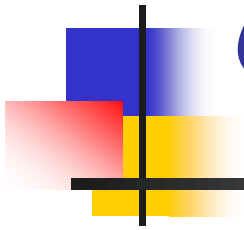
UNIVERSIDADE CATÓLICA DE PERNAMBUCO



Conversão de Tipos e Arrays

Universidade Católica de Pernambuco
Ciência da Computação

Prof. Márcio Bueno
poonoite@marciobueno.com

Fonte: Material da Profª Karina Oliveira

Conversão de Tipos

- Permite converter valores de um tipo para outro
- Pode assumir duas formas:
  - Conversão implícita
  - Conversão explícita
- Conversão implícita
  - Não requer código adicional
  - A conversão é feita de forma transparente pelo compilador
  - Regra geral: um **tipo A** pode ser convertido para um **tipo B** se o intervalo de valores possíveis no tipo A se encaixa perfeitamente dentro do intervalo de valores possíveis no tipo B – alargamento de tipo.

Conversão Implícita

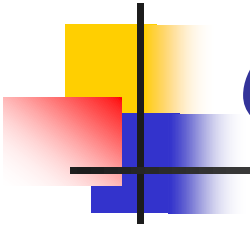
Tipo	Pode ser implicitamente convertido para:
short	int, long, float e double
int	long, float e double
long	float e double
float	double
char	int, long, float e double

Conversão Implícita

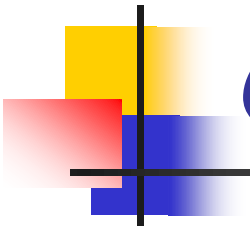
- Exemplo:

```
public class Exemplo {
 public static void main(String[] args) {
      short a = 1;
      int b = 2;
      long c = 3;
      float d = 4.5F;
      double e = 6.3;
      char f = 'a';
 }
}
```

Conversão Explícita



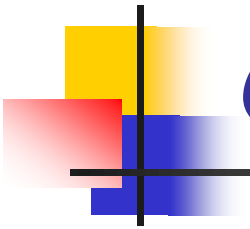
- Requer código adicional
- Deve-se escrever explicitamente para qual tipo deseja-se converter
- É alcançada utilizando um **cast**
- Sintaxe: `(TipoDestino) variavelOrigem`
- Semântica: Esse código converterá o valor da *variavelOrigem* para o *TipoDestino*.
- OBS.: Os tipos que não possuem nenhum relacionamento não terão *casts* definidos.



Conversão Explícita

- Exemplo:

```
public class Exemplo {
 public static void main(String[] args) {
      short a = 1;
      int b = 50000;
      double e = 6.3;
 }
}
```



Conversão Explícita

- Conversão de tipos explícita usando métodos
  - Usados para converter valores do tipo string para os tipos simples (numéricos e caracter) e vice-versa
    - String ⇒ Número
      - A string fornecida deve ser uma representação válida de um número e este número não pode provocar *overflow*.

Conversão Explícita

- Conversão de tipos explícita usando métodos

Comando	Resultado
Short.parseShort(val)	val é convertido para short
Integer.parseInt(val)	val é convertido para int
Long.parseLong(val)	val é convertido para long
Float.parseFloat(val)	val é convertido para float
Double.parseDouble(val)	val é convertido para double

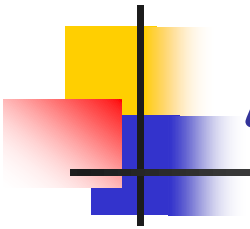
Conversão Explícita

- Conversão de tipos explícita usando métodos

Comando	Resultado
String.valueOf(val)	val é convertido para String

- OBS: val pode ser qualquer um dos tipos simples, por exemplo: short, int, long, float, double, etc.
- Uma outra forma de conversão dos tipos simples para String é concatenar o valor do tipo simples com uma String vazia, exemplo:

```
int a = 10;
String s = a + "";
```

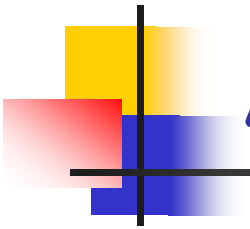


Array

- Definição
  - Tipo de dado usado para representar uma coleção de variáveis de um mesmo tipo.
  - Uma dimensão $\Rightarrow$ vetor
  - Duas dimensões $\Rightarrow$ matriz
  - Uma vez criado, um array não pode ter seu tamanho alterado.

Criando Arrays

- Exemplo: Criando Vetores
  - Sintaxe:
    - `<tipo> vetor[ ] = new <tipo>[tamanho];`
    - `<tipo>[ ] vetor = new <tipo>[tamanho];`
    - `<tipo> vetor[ ] = {val0, val1, ..., valN-1};`
    - `<tipo>[ ] vetor = {val0, val1, ..., valN-1};`
    - `<tipo>[ ] vetor = new <tipo>[ ] {val0, ..., valN-1};`
    - `<tipo> vetor [ ]= new <tipo>[ ] {val0, ..., valN-1};`
  - Exemplos:
    - `int v1[ ] = new int[5];`
    - `int[ ]v2 = new int[5];`
    - `int v3[ ] = {1, 2, 3};`
    - `int[ ]v4 = {1, 2, 3};`
    - `int[ ]v5 = new int[ ] {1, 2, 3};`
    - `int v5[ ] = new int[ ] {1, 2, 3};`



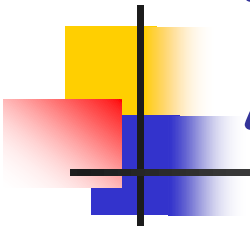
Array

- Exemplo: Criando Matrizes
  - Sintaxe:
    - <tipo> nome[ ][ ] = new <tipo>[linhas][colunas];
    - <tipo>[ ][ ] nome = new <tipo>[linhas][colunas];
    - <tipo>[ ][ ] matriz = {{val_{00}, val_{01}}, {val_{10}, val_{11}}};
    - <tipo>matriz[ ][ ] = {{val_{00}, val_{01}}, {val_{10}, val_{11}}};
  - Exemplos:
    - `int m1[ ][ ] = new int[2][2];`
    - `int[ ][ ] m2 = new int[2][2];`
    - `int[ ][ ] m3 = { {1, 2}, {3, 4} };`
    - `int m3[ ][ ] = { {1, 2}, {3, 4} };`

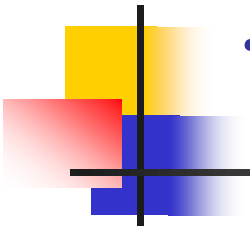
Acessando os Elementos do Array

- Os elementos de um array são indexados a partir da posição 0 (zero).
- Pode-se acessar individualmente os valores de seus elementos
- Sintaxe:
  - Vetor: **nome[<índice>]**
  - Matriz: **nome[<linha>][<coluna>]**
- Exemplos:
  - Vetor: `v1[2]`
  - Matriz: `m1[1][1]`

Acessando os Elementos do Array



- Não é permitido acessar um elemento de um array fora do seu limite
  - Erro em tempo de execução
  - Exemplo: Um vetor de 100 posições tem seus elementos numerados de 0 a 99. O acesso ao elemento de índice 100 causará um erro na execução do programa.

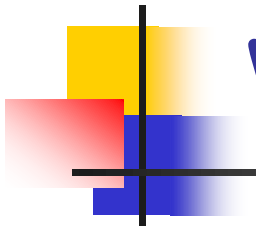


Tamanho do Array

- Obtendo o tamanho de um array
  - Usar a propriedade length
  - Sintaxe: `nomeArray.length`
  - Exemplos:

```
int[ ]vet = new int[10];
int tam = vet.length;

int[ ][ ] matriz = new int[2][3];
int linhas = matriz.length;
int colunas = matriz[0].length;
```

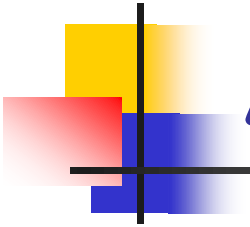


Vetor - Exemplo

```
public class ExemploVetor {
 public static void main(String args[ ]) {
  int[ ] vetor = new int [5];
  for (int i = 0; i < vetor.length; i++) {
   vetor[i]=Integer.parseInt(JOptionPane.showInputDialog("Digite:"));
  }
  for (int i = 0; i < vetor.length; i++) {
   System.out.println(vetor[i]);
  }
 }
}
```

Matriz - Exemplo

```
public class ExemploMatriz {
 public static void main(String args[ ]) {
  int[ ][ ] matriz = new int[2][2];
  for (int i = 0; i < matriz.length; i++) {
   for (int j = 0; j < matriz[0].length; j++) {
    matriz[i][j]=Integer.parseInt(JOPtionPane.showInputDialog("Digite:"));
   }
  }
  for (int i = 0; i < matriz.length; i++) {
   for (int j = 0; j < matriz[0].length; j++)
    System.out.print(matriz[i][j] + " ");
   }
   System.out.println("");
  }
 }
}
```



Array - Exercícios

- Exercício 1: Escreva um programa que receba um vetor com 5 números inteiros. Em seguida, determine e imprima na tela o maior elemento par do vetor (se houver), o menor elemento ímpar do vetor (se houver), o somatório dos elementos do vetor e a média.
- Exercício 2: Escreva um programa que recebe valores para duas matrizes 2x2 de inteiros. Criar uma terceira matriz que é a matriz soma das anteriores e imprimi-la ao final.