

# Introdução a linguagem de programação Java

Vetores e repetidores

Prof. Rhafael Freitas da Costa, M. Sc.

## Desenvolvimento de Software

# Aspectos básicos



## Comentários:

- ❑ Anotações do programador. Não são compiladas;
- ❑ Útil para remover linhas de código, sem deletar ("comentar").

```
21 // Isso é um comentário de uma linha
22
23 /**
24  * Isso é um comentário para
25  * multi-linhas
26  */
```



## Variáveis:

- ☐ São uma forma de guardar dados;
- ☐ Variáveis possuem tipos (por ex: int - número inteiro);
- ☐ Devem ser declaradas e inicializadas;
- ☐ Sinal de igual atribui o valor do lado direito à variável do lado esquerdo;
- ☐ É possível mudar o valor depois de declarada;

tipo nomeDaVariavel;

nomeDaVariavel = valor;

nomeDaVariavel = outroValor;

int numeroInteiro = 10;



## Variáveis:

- ❑ Números inteiros: int (32 bits), byte (8 bits), short (16 bits), long (64 bits);
- ❑ Números decimais: float (32 bits), double (64 bits);
- ❑ boolean: true ou false (1 bit);
- ❑ char: um único caractere entre aspas simples (16 bits), ex: 'A';
- ❑ String: sequência de chars entre aspas duplas (tamanho variável), ex: "Olá mundo".



## Constantes:

- ❑ Variáveis que nunca mudam de valor;
- ❑ Declaração e inicialização em uma única linha;
- ❑ Por convenção, nome em letras maiúsculas.

```
final tipo CONSTANTE = valor;
```

```
final double PI = 3.1415;
```



# Aspectos básicos da linguagem Java

## ☐ Condicionais

- ☐ if / else / else if / switch + break

## ☐ Repetição

- ☐ for / while / do ... While

- ☐ break / continue



## Funções da classe Math:

- ☐ **Constantes:** `Math.PI` (3,1415...) / `Math.E` (2,7182...);
- ☐ **Arredondamento:** `Math.round()`;
- ☐ **Log:** `Math.log()` e `Math.log10()`;
- ☐ **Potência:** `Math.pow()`;
- ☐ **Raiz quadrada:** `Math.sqrt()`;
- ☐ **Trigonometria:** `Math.sin()`, `Math.cos()`, `Math.tan()`;
- ☐ **Números aleatórios:** `Math.random()`;





## Funções da Classe Character:

- ☐ toUpperCase();
- ☐ toLowerCase();
- ☐ isDigit();
- ☐ isLetter();
- ☐ isLetterOrDigit().



## Funções da Classe String:

- ❑ **Posição da String:** `charAt()` (começa em 0);
- ❑ **Tamanho da String:** `length()`;
- ❑ **Conversão de String para Maiúsculas e Minúsculas:** `toUpperCase()` / `toLowerCase()`;
- ❑ **Concatenação de Strings:**
  - ❑ `System.out.println("Olá " + "mundo").`



# Aspectos básicos da linguagem Java

- ❑ Data e tempo

- ❑ Classe Date e Calendar

```
import java.util.Date;  
  
import java.util.Calendar;  
  
Date date = new Date();  
  
date.toString();
```

- ❑ Para manipular a data, utilizar a classe Calendar

```
Calendar cal = Calendar.getInstance();  
  
cal.setTime(d);
```



## Aspectos básicos da linguagem Java

- ❑ **Ano:** `cal.get(Calendar.YEAR);`
- ❑ **Mês:** `cal.get(Calendar.MONTH);` **CUIDADO:** Mês inicia em 0;
- ❑ **Dia:** `cal.get(Calendar.DAY_OF_MONTH);`
  
- ❑ **Somar um intervalo de tempo:** `cal.add(Calendar.HOUR, 2);`
- ❑ **Converter Calendar de volta para Date:** `cal.getTime();`



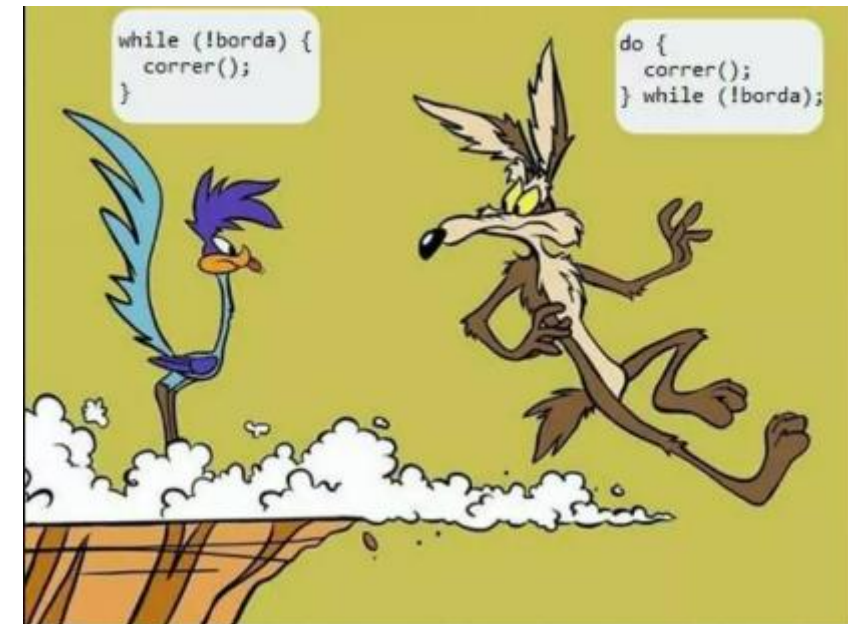
# Estruturas de Repetições



# Controle de fluxo de execução

- ❑ Um loop while se repetirá infinitamente enquanto a condição dentro dos parênteses () for verdadeira. Algo deve mudar a condição, ou o loop while nunca encerrará. O loop do...while funciona da mesma forma que o loop while, com a exceção da condição ser testada no final.

```
7
8      boolean result = true;
9
10     while (result) {
11         // Bloco de Comando
12     }
13
14     do {
15         // Bloco de Comando
16     } while (result);
17
```



# Controle de fluxo de execução

- ❑ Estrutura de repetição usada para repetir um bloco de código envolvido por chaves. Um contador de incremento é utilizado para terminar o loop. A inicialização ocorre primeiro e apenas uma vez . A cada repetição do loop, a condição é testada e se for verdadeira, o bloco de comando e o incremento são executados. Quando a condição se torna o loop termina.

```
8      String lista[] = {"1", "2", "3"};
9
10     for (var i = 0; i < 10; i++) {
11         // Bloco de Comando
12     }
13
14     for (String item : lista) {
15         // Bloco de Comando
16     };
17
```



# Arrays



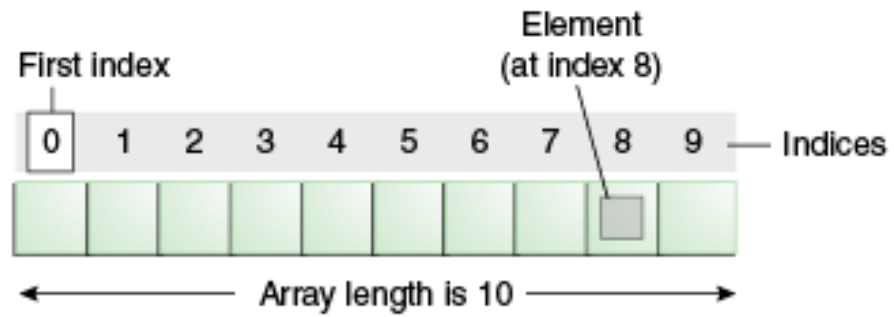


# Arrays

- ☐ Arrays, são estruturas de dados que permitem armazenar uma coleção de elementos do mesmo tipo. Esses elementos são organizados sequencialmente na memória e acessados por um índice;
- ☐ Acesso direto;
- ☐ Conjunto finito;
- ☐ Elementos identificados por índices;
- ☐ Pode ter mais de uma dimensão.



# Arrays unidimensionais



An array of 10 elements.



# Criação de array

- ❑ Criação:

- ❑ `int [] vetor = new int []`

- ❑ Atribuição:

- ❑ `vetor[0] = 34;`

- ❑ `vetor[1] = 27;`

- ❑ `vetor[2] = 3;`

- ❑ Leitura:

- ❑ `int valor1 = vetor[1];`

- ❑ `int valor2 = vetor[2];`



- ❑ Inicialização em tempo de compilação:

```
String[ ] naipes = {"copas","ouros","paus","espadas"};
```

```
double[ ] temperaturas = {45.0,32.0,21.7,28.2,27.4};
```



- ❑ Ciclo de laço implícito (“for-each”):

```
String[ ] naipes = {"copas","ouros","paus","espadas"};  
for (String naipe : naipes)  
    system.out.println(naipe);
```



# Exercícios



## Exercício 3

- ☐ Transformar o programa do exercício 1 para um sistema que permita ler a entrada de dados pelo usuário em um vetor durante execução;
- ☐ Permitir que o usuário informe, primeiro, os dados de 5 (cinco) alunos e depois de capturados os dados, imprimir o relatório final com todos os dados:
- ☐ Matrícula: xxxxx
- ☐ Nome: xxxxx xxxxx
- ☐ Aprovado: (x ) Sim ( ) Não
- ☐ Nota final: xxxxx



## Exercício 4

- ☐ Criar um programa para processar as seguintes informações de um aluno: matrícula, nome, nota 1, nota 2, nota 3, nota 4 e nota 5;
- ☐ O sistema deverá ler as notas do aluno em um vetor e a entrada de dados pelo usuário só poderá ser nos valores de 0 a 10;
- ☐ No final da execução imprimir o seguinte relatório:
- ☐ Matrícula: xxxxx
- ☐ Nome: xxxxx xxxxx
- ☐ Nota (x ) : xxxxx
- ☐ Nota (y ) : yyyyyy
- ☐ (...)





**OBRIGADO!**

## **Desenvolvimento de Software**

Professor Rhafael Freitas  
da Costa, M. Sc.