INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA - INTRODUÇÃO

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Linguagem de Programação
  - Criada pela Sun Microsystems
  - Especificada pelo JCP (Java Community Process)
- Plataforma
  - Ferramentas
  - APIs (Application Programming Interface)
  - Ambiente de execução –JRE (Java Runtime Environment)

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Características:
  - Simples
  - Robusta
  - **Orientada a Objetos**
  - Segura
  - Portável
  - Multi-Plataforma
  - Multithreaded
  - Alto desempenho

INSTITUTO FEDERAL
Norte de Minas Gerais

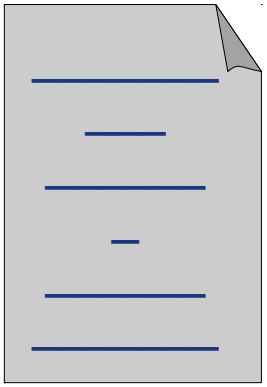
JAVA – INTRODUÇÃO

- Ambiente de execução
  - JRE (Java Runtime Environment)
- Kit de desenvolvimento
  - JDK (Java Development Kit)
    - JRE
    - Compilador (javac)
    - Depurador (jdb)
    - Empacotador (jar)
    - Outros

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Código Fonte:
  - Arquivo de texto simples
  - Extensão .java



OlaMundo.java

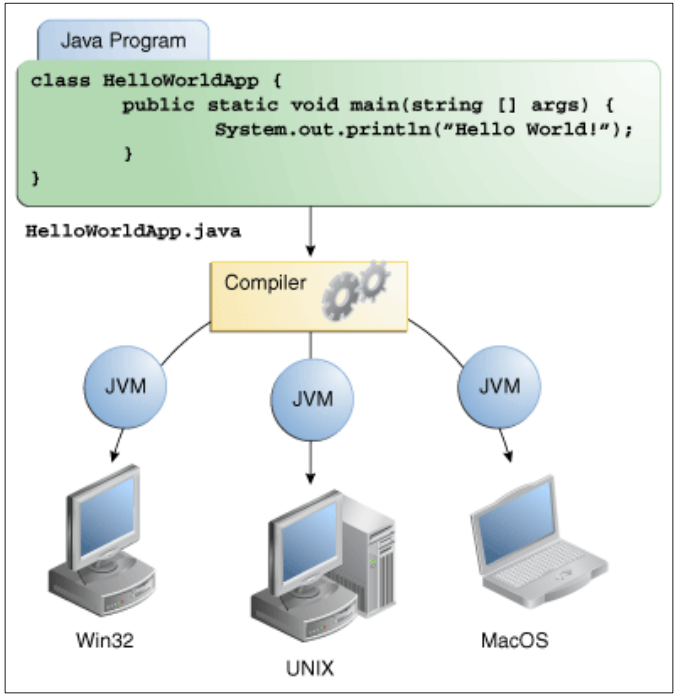
```
public class OlaMundo{
        public static void main( String[ ] args){
                System.out.println(“Olá Mundo”);
        }
}
```

O arquivo .java e a classe possuem o mesmo nome.

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Compilação
  - Converte código fonte em bytecodes.
    - Extensão do Arquivo: .class
  - Compilador Java:
    - Javac



INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Execução
  - Máquina Virtual Java (Java Virtual Machine - JVM)
    - Máquina virtual que interpreta e executa código Java compilado
    - Possibilita que a linguagem seja Multi-Plataforma
    - Segurança

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Programando em Java
  - O código fonte pode ser criado utilizando qualquer editor de textos (gedit, nano, vim, bloco de notas, notepad++, etc.).
  - Existem IDE's que auxiliam no desevolvimento (NetBeans, Eclipse, BlueJ, etc.).

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Primeiros passos:
  - Codificar:
    - Utilizar qualquer editor (texto puro) bloco de notas, wordpad , jedit, vi, kedit, pico, gedit, …
  - Salvar arquivo com a extensão .java
    - Ex.: NomeDaClasse.java

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Primeiros passos:
  - Compilar:
    - javac NomeDaClasse.java
  - Executar:
    - java NomeDaClasse

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – INTRODUÇÃO

- Exemplo:

```
public class OlaMundo{
        public static void main( String[ ] args ) {
                System.out.println(“Olá Mundo”);
        }
}
```

- Crie o arquivo, compile e execute.

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Java é Case Sensitive
  - Diferencia Maiúscula de Minúscula
- Tudo em Java deve estar dentro de uma Classe
- Regras para o nome de uma Classe:
  - Nomes devem iniciar com uma letra **(Maiúscula)**
  - Depois da letra pode ter qualquer combinação de letras e números
  - O nome do arquivo do código fonte deve ter o mesmo nome da classe
  - Não pode utilizar uma palavra reservada do Java

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Convenções em relação ao nome de Classes:
  - Deve ser um substantivo, com a primeira letra em maiúsculo.
  - Se tiver múltiplas palavras, deve ter a letra inicial maiúscula em cada uma das palavras
  - Exemplos:
    - PrimeiroExemplo; ClasseExemplo; CarroDeMao.

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Palavras Reservadas do Java:

byte - short - int - long - char - boolean - double - float - public - private - protected - static - abstract - final - strictfp - transient - synchronized - native - void - class - interface - implements - extends - if - else - do - default - switch - case - break - continue - assert - const - goto - throws - throw - new - catch - try - finally - return - this - package - import - instaceof - while - for - volatile - super

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Tipos de Dados
  - Java é fortemente tipada
  - Tipos:
    - Primitivos
    - Referências para Objetos

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Tipos Primitivos
  - Definidos pela linguagem
  - Não precisam de construtor
  - Podem ser:
    - Inteiro
    - Numéricos de Ponto Flutuante
    - Outros (char e boolean)

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Tipos Primitivos
  - Inteiros:

Palavra	Descrição	Tamanho/Formato	Valores Possíveis
byte	Inteiro de 1 byte	8 bits	- 128 a + 127
short	Inteiro pequeno	16 bits	- 32.768 a + 32.767
Int	Inteiro	32 bits	- 2.147.483.648 a + 2.147.483.647
long	Inteiro longo	64 bits	-9.223.372.036.854.775.808 a + 9.223.372.036.854.775.807

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Tipos Primitivos
  - Numéricos de Ponto Flutuante::

Palavra	Descrição	Tamanho/Formato	Valores Possíveis
float	Ponto flutuante de precisão simples	32 bits	- 1.40239846E-45 a + 3.40282347E + 38 (com nove dígitos significativos de precisão)
double	Ponto flutuante de precisão dupla	64 bits	- 4.94065645841246544E-324 a + 1.79769313486231570E + 308 (com 18 dígitos significativos de precisão)

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Tipos Primitivos
  - Outros:

Palavra	Descrição	Tamanho/Formato	Valores Possíveis
char	Um caractere	16 bits	0 a 65535
boolean	Um valor lógico		true ou false

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Conversão entre tipos primitivos:
  - Implícita
    - Tamanho de uma variável é maior que o tamanho da variável ou o valor que está sendo atribuído

```
int x = 5;
long z = x;
```

  - Explícita
    - Tamanho de uma variável é menor que o tamanho da variável ou o valor que está sendo atribuído (cast)

```
long z = 10;
int x = (int) z;
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Variáveis
  - Servem para armazenam algum valor
  - Cada variável tem um tipo
  - Regras e convenções:
    - Nomes devem iniciar com uma letra (Minúscula)
    - Depois da letra pode ter qualquer combinação de letras e números
    - Se tiver múltiplas palavras, deve ter a letra inicial maiúscula em cada uma das palavras, a partir da segunda
  - Declaração de uma variável:
    - tipo nomeVariavel;

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Variáveis
  - Exemplos:

```
int x;
int numero;
float quantidade;
char primeiraLetra;

int a, b, c;
int num = 0;
float k=10, m=5, n=k+m;
int e=f=g=0;
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Operadores
  - Aritméticos
  - Lógicos

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Aritméticos

Operador	Função
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da Divisão

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Lógicos

Operador	Função
&&	E
||	OU
==	IGUAL
!=	DIFERENTE

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Observações:
  - Quando operandos forem de tipos diferentes em uma operação, ambos são convertidos em um tipo comum antes de a operação ser executada
  - Regras:
    - Se um dos operandos for double, o outro será convertido em double
    - Caso contrário, se um for do tipo float, o outro será convertido em float
    - Caso contrário, se um for do tipo long, o outro será convertido em long
    - Caso contrário, ambos os operandos serão convertidos em um int

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Operadores
  - Precedência de Operadores Aritméticos
    - Da esquerda para a direita:
      - [], (),
      - ++, --
      - *, /, %
      - +, -
      - <, <=, >=, <

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Atribuição:
  - Em JAVA a atribuição é um operador
    - retorna um valor
    - pode ser usada em expressões
  - Exemplos:
    - x = 10;
    - y = 20+x;
    - a = b = c = 0;
    - z = 4+(x=10*y)*(5+y)

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Atribuição:
  - “=“
  - “+=“
  - “-=“
  - “*=”
  - “/=“
  - “%=“

```
a = a + b;
ou
a+=b;
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Operadores:
  - Incremento e Decremento
    - Representação:
      - “++”
      - “--“
- • Exemplo:

```
int a = 10;
int b, c, d, e;
b = a++;
c = ++a;
d = a--;
e = --a;
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- String
  - Sequências de caracteres Unicode
  - Texto entre aspas são objetos (instância) da classe String
    - String s = “Isto é uma String”
    - mais eficiente que usar new

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- String
  - São indexados a partir do zero
    - Implementadas como array de char
    - “CEFET” usa os índices 0, 1, 2, 3, 4

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- String
  - Concatenação:
    - Agrupação de duas Strings
    - Utiliza o sinal de “+”
  -

```
String texto1 = “Olá ”;
String texto2 = “Mundo”;
String texto3 = texto1 + texto2;
String texto4 = texto3 + “!”;
System.out.println(“Teste: “+texto4);
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- String
  - Igualdade em String
    - Para se testar a igualdade em Strings, utilizar *equals* ou *equalsIgnoreCase.*
  -

```
String texto1 = “Olá”;
String texto2 = “olá”;
String texto3 = “Ola”;

texto1.equals(texto2);
texto1.equalsIgnoreCase(texto2);
texto1.equals(texto3);
texto2.equalsIgonereCase(texto3);
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- String
  - Substring:
    - Extração de uma parte da String com o método substring
  -

```
String texto = “Texto Completo”;
String parteDoTexto = texto.substring(0, 8);
System.out.println(parteDoTexto);
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- String
  - Alguns métodos da Classe String:
    - substring(i, j) e substr(i, j)
      - » Obtém uma parte do texto (j caracteres a partir de i)
    - length()
      - » Retorna o tamanho da String
    - charAt(i)
      - » retorna o caractere no índice i
    - indexOf('c')
      - » Retorna o índice do caractere c
    -

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- String
  - Alguns métodos da Classe String:
    - toLowerCase()
      - » Muda o texto para minúsculo
    - toUpperCase()
      - » Muda o texto para Maiúsculo
    - trim()
      - » Tira os espaços em branco a esquerda e direita da String
    - Replace(“Texto1”, “Texto2”)
      - » Substitui o Texto1 por Texto2

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Entrada e Saída
  - A saída de dados é proporcionada pelo atributo *out* da classe *System.*
  - Possui os métodos *print* e *println*.
  - Exemplos:
    - System.out.print(“Teste”);
    - System.out.println(“Teste”);
  - Concatenar com valores de variáveis: “+”
    - System.out.println(“Numero: ”+n);

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Entrada e Saída
  - Outra opção é o método *printf*.
  - Saída formatada (semelhante à C).
  - Exemplos:
    - System.out.printf(“Teste”);
    - System.out.printf(“O valor é %i”, n);
  -

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Entrada e Saída
  - A entrada de dados é proporcionada pela classe *Scanner*.
  - Importar o pacote java.util.Scanner.
    - import java.util.Scanner;
      ou
    - import java.uitil.*;

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Scanner
  - É preciso instanciar um objeto da classe Scanner.
    - Scanner entrada = new Scanner(System.in);
  - Cada tipo de entrada possui um método:
    - Texto
      - entrada.nextLine();
    - Inteiro
      - entrada.nextInt();
    - Double
      - entrada.nextDouble();
    - Float
      - entrada.nextFloat();

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Conceitos Iniciais

- Entrada e Saída
  - Exemplo:

```
import java.util.Scanner;
public class Aula {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Digite o seu nome completo");
        String nome = scan.nextLine();
        System.out.println("Digite a sua idade");
        int idade = scan.nextInt();
        System.out.println("Nome: "+nome+" Idade:"+idade);
    }
}
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Exercícios

- Escreva um programa que leia o nome e três notas de um aluno, calcule e mostre a média.
- Escreva um programa que leia um valor inteiro e apresente os resultados do quadrado e do cubo do valor lido.
- Escreva um programa que leia o valor de um produto e mostre o valor a ser pago com um desconto de 10% á vista ou o valor em três parcelas iguais sem acréscimo.

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Fluxo de Controle:
  - IF

```
If (condição)
    comando;
```

```
If (a>b)
    System.out.print(a+" é maior");
```

  - IF .. ELSE

```
If (condição)
    comando;
else
    comando;
```

```
If (a>b)
    System.out.print(a+" é maior");
else
    System.out.print(b+" é maior");
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Fluxo de Controle:
  - IF (com bloco de comando)

```
If (condição){
    comando1;
    comando2;
}
```

```
If (a>b){
    x = a;
    System.out.print(x+" é maior");
}
```

  - IF .. ELSE

```
If (condição){
    comando1;
    comando2;
}else {
    comando3;
    comando4;
}
```

```
If (a>b){
    x = a;
    System.out.print(x+" é maior");
}else{
    x = b;
    System.out.print(x+" é maior");
}
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Fluxo de Controle:
  - Operador Ternário:

```
condição ? valor : valor ;
```

```
System.out.print( a>b ? a + " é maior" : b + " é maior" );

ou

System.out.print( (a>b ? a : b) + " é maior" );
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Fluxo de Controle:
  - Switch:

```
switch (variável) {
      case <valor1>:comando1; break;
      case <valor2>:comando2; break;
      case <valor3>:comando3; break;
      …
      default: comando;
}
```

  - A variável pode ser int ou char.
  - As cláusulas break e default são opcionais.

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Fluxo de Controle:
  - Switch:

```
switch (opcao) {
      case 1:
            System.out.println("Opção 1");
            break;
      case 2:
            System.out.println("Opção 2");
            break;
      case 3:
            System.out.println("Opção 3");
            break;
      default:
            System.out.println("Opção inválida");
}
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Loop's:
  - do:

```
do{
    comando1;
    comando2;
} while(condição);
```

  - while:

```
while(condição){
    comando1;
    comando2;
}
```

  - for:

```
for(inicialização; condição; passo){
    comando1;
    comando2;
}
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Alteração de Fluxo:
  - break:

```
int i = 10;
do{
    if(i<5)  break;
    System.out.println(i);
    i--;
} while(i>0);
```

  - continue:

```
int i = 10;
do{
    if(i<5)  continue;
    System.out.println(i);
    i--;
} while(i>0);
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Coleções

- Vetores (Arrays):
  - Vetores podem armazenar tipos primitivos ou objetos.
    Ao se criar um vetor suas posições não possuem valor e precisam ser inicializadas antes do uso.
  - Criação de vetores:
    tipo [ ] nome;
  - Criação com definição de tamanho:
    tipo [ ] nome = new tipo [tamanho];
  - Criação de vetor preenchido:
    tipo [ ] nome = {valor1, valor2, valor3, ...}

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Coleções

- Vetores (Arrays):
  - Exemplos:

```
int[] vetor; // Criação de um vetor.

float[] vetor = new float[10]; // Criação de um vetor com tamanho definido;

String[] alunos = {“Alex”, “Ana”, “Douglas”, “Emanuel”}
```

INSTITUTO FEDERAL
Norte de Minas Gerais

JAVA – Estruturas de Controle

- Exercícios