

Introdução à Java



Prof. Bruno Gomes
bruno.gomes@ifrn.edu.br

Programação Orientada a Objetos

Código Exemplo da Aula



- Utilizem a classe Aula.java da aula passada:

```
public class Aula {  
    public static void main(String[] args) {  
        System.out.println("Codigo Exemplo");  
    }  
}
```

- **Compilar e Executar**



Tipos de Dados Enumerados



- Variável que só armazena um conjunto restrito de valores
- Conhecido como **Tipo Enumerado**
- Este tipo tem um número finito de valores nomeados

```
public class Aula {  
    enum Size {PEQUENO, MEDIO, LARGO, EXTRA_LARGO};  
    public static void main(String[] args) {  
        Size s = Size.MEDIO;  
        System.out.println(s);  
    }  
}
```



String



- Sequências de caracteres Unicode

```
String s = new String("IFRN");  
String texto = ""; // String vazia  
String outroTexto = "Texto aqui";
```

Quantos caracteres Unicode existem na String: **"Java \u2122"** ?



String



- Texto entre aspas são objetos (instância) da classe String
 - `String s = "Isto é uma String"`
 - mais eficiente que usar `new`
- Usada normalmente onde pede-se String:
 - `int tamanho = "Qual o tamanho?".length();`
 - `String s1 = "um nome qualquer".toUpperCase();`
 - `String s = "CEFET" + "-RN";`
 - São criada 3 Strings: `"CEFET"`, `"-RN"` e `"CEFET-RN"`
- São indexados a partir do **zero**
 - Implementadas como array de char
 - `"CEFET"` usa os índices 0, 1, 2, 3, 4



String



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

```
String texto1 = "Ola ";  
String texto2 = "Mundo!";  
String texto3 = texto1 + texto2;  
  
String texto4 = texto1 + "Mundo!";  
  
System.out.println("Testando o "+texto4);
```



String



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

```
String texto1 = "Ola";  
String texto2 = "Ola";  
  
texto1.equals(texto2);  
texto1.equalsIgnoreCase(texto2);  
"Ola".equalsIgnoreCase(texto2);  
texto1.equals("Ola");
```



String



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

```
String texto = "Texto completo para teste";  
String parteTexto = texto.substring(0, 8);
```

Qual o valor que a variável
parteTexto vai receber?



String



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



Fluxo de Controle



- Java suporta:
 - Instruções Condicionais
 - Loops
 - Loops Determinados
 - Múltiplas Seleções

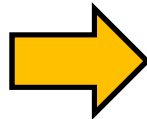


Instruções Condicionais



- If:

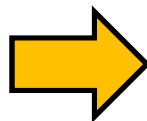
```
if(expressão)  
    comando;
```



```
if(6>5)  
    System.out.println("Verdadeira");
```

- If..Else

```
if(expressão)  
    comando;  
else  
    comando;
```



```
if(6>5)  
    System.out.println("Verdadeira");  
else  
    System.out.println("Falsa");
```



Instruções Condicionais



- Utilizando um bloco de comandos:

```
int meta = 30;  
double salario = 0;  
if(meta>50){  
    System.out.println("Meta atingida");  
    salario = 1000 * 1.5;  
}else{  
    System.out.println("Meta não atingida");  
    salario = 1000 * 1.2;  
}  
System.out.println("Salario: "+salario);
```



Instruções Condicionais



- Exemplo:

```
if(x <=0) if (x==0) variavel = 0; else variavel = -1;
```

O **else** pertence a qual **if**?



Instruções Condicionais



- Exemplo:

```
int x = 0;  
if(x==0){  
    // instrucao  
}else if(x==1){  
    // instrucao  
}else if(x==2){  
    // instrucao  
}
```

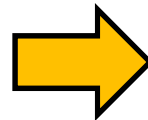


Loops



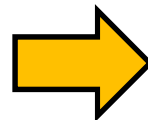
- While:

```
while(expressão)  
    comando;
```



```
int numero = 1;  
int soma = 0;  
while(numero < 5)  
    soma += numero++;
```

```
do  
    comando;  
while (expressão);
```



```
int numero = 1;  
int soma = 0;  
do  
    soma += numero++;  
while(numero < 5);
```



Loops Determinados



- For:

```
for ( inicializações ; expressão ; passo )  
    comando;
```



```
int soma = 0;  
for (int i=0; i<10; i++){  
    soma+=i;  
}
```



Múltiplas Seleções



- Switch:

```
switch ( <expressão> ) {  
    case <valor1>:<comandos 1>  
    [break;]  
    case <valor2>:<comandos 2>  
    [break;]  
    case <valor3>:<comandos 3>  
    [break;]  
    case <valor4>:<comandos 4>  
    [break;]  
    ...  
    default:  
        <comandos>  
}
```

- A variável de teste pode ser:
 - int (tipos inteiros)
 - char
- A instrução break é opcional!
- A cláusula default é opcional!



Múltiplas Seleções



- Exemplo de Switch:

```
int num = 10;

switch (num) {
case 1:
    System.out.println("Numero 1");
    break;
case 5:
    System.out.println("Numero 5");
case 10:
    System.out.println("Numero 10");
    break;
default:
    System.out.println("Conferido");
}
```



Operador Condicional



- O Operador Ternário ?:
 - Forma compacta de expressar uma instrução if – else
- Sintaxe:

(expressão) ? resultadoSeVerdadeiro : resultadoSeFalso;



Operador Condicional



- Exemplos:

```
int a = 10;  
Int b = 20;  
int maior = (a > b) ? a : b;  
System.out.println("Maior = " + maior);
```

```
int a = 10;  
Int b = 20;  
String resultado = (a > b) ? "a maior que b" : "b maior que a";  
System.out.println(resultado);
```



Fluxo de Execução



- Alterando o fluxo normal de execução de um laço:
- Instrução **break**
 - Faz com que o laço completo seja interrompido imediatamente
- Instrução **continue**
 - Faz com que o ciclo atual do laço seja terminado imediatamente. A execução continua com o próximo ciclo do laço



Fluxo de Execução



- Exemplo *break*:

```
int num = 1;  
while(num<=10){  
    if(num==6)  
        break;  
    System.out.println(num);  
    num++;  
}
```



Fluxo de Execução



- Exemplo *continue*:

```
int num = 0;
while(num<=10){
    num++;
    if(num>=6)
        continue;
    System.out.println(num);
}
```



Exercício 1



- Utilizar mesmo exercício da aula passada
- Nele, verificar a situação do aluno (Aprovado ou em Recuperação), sendo a média 7.
- Ao final, exibir no console a situação do aluno



Exercício 2



- Crie um programa em Java que peça para o usuário digitar uma frase, e logo em seguida informa a quantidade de letras digitadas, e quantas letras 'a' tem na frase.



Exercício 3



- Faça um Loop determinado (For) que imprima uma contagem regressiva a partir do 20.

