

# Comandos de condicionais e de repetição

# Comandos Condicionais

- Os comandos condicionais existentes em Java são os mesmos aprendidos na linguagem C, até mesmo a sintaxe.
- Existem o if-else e o switch-case.
- Nos próximos slides teremos um exemplo com if-else e outro com o switch-case.

# Exemplo – if ... else

Vale a pena notar que se for digitada uma String não numérica, o programa irá gerar a mensagem de erro "Digite apenas valores numéricos" devido ao tratamento de exceção.

```
package condicionais01;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Condicionais01 {

    public static void main(String[] args) throws IOException {
        try {
            System.out.println("Entre com a nota: ");
            BufferedReader dado = new BufferedReader(new InputStreamReader(System.in));
            String s = dado.readLine();
            double nota = Float.parseFloat(s);

            if (nota <= 100 && nota >= 0) {
                System.out.println("Nota = " + nota + " - valor válido");
            } else {
                System.out.println("Nota = " + nota + " - valor inválido");
            }
        } catch (NumberFormatException erro) {
            System.out.println("Digite apenas valores numericos");
        }
    }
}
```

run:  
Entre com a nota:  
1  
Nota = 1.0 - valor válido

run:  
Entre com a nota:  
-1  
Nota = -1.0 - valor inválido

run:  
Entre com a nota:  
a  
Digite apenas valores numericos  
BUILD SUCCESSFUL (total time: 1 second)

```
package condicionais02;
```

```
public class Condicionais02 {
```

```
    public static void main(String[] args) {
```

```
        if (args.length == 1) {
```

```
            int x;
```

```
            x = Integer.parseInt(args[0]);
```

```
            switch (x) {
```

```
                case 10:
```

```
                    System.out.println("dez");
```

```
                    break;
```

```
                case 50:
```

```
                    System.out.println("cinquenta");
```

```
                    break;
```

```
                case 80:
```

```
                    System.out.println("oitenta");
```

```
                    break;
```

```
                case 90:
```

```
                    System.out.println("noventa");
```

```
                    break;
```

```
                case 100:
```

```
                    System.out.println("cem");
```

```
                    break;
```

```
                default:
```

```
                    System.out.println("Numero desconhecido");
```

```
            }
```

```
        } else {
```

```
            System.out.println("Numero de argumentos invalido");
```

```
        }
```

```
    }
```

```
}
```

# Exemplo switch-case

# Comandos de repetição

- for: A instrução for é um tipo de contador finito, isto é, ela realiza a contagem de um valor inicial conhecido até um valor final também conhecido.

for(inicialização; condição; incremento ou decremento )

Onde:

- Inicialização: é o valor inicial da variável de controle do laço.
- Condição: é a variável que controla a continuação do laço. Essa condição deve conter o valor final que a variável de controle pode assumir dentro do laço.
- Incremento ou decremento: é o passo com que a variável de controle será acrescida ou decrescida.

```

package repeticoes01;
public class Repeticoes01 {

    public static void main(String[] args) {
        for (int i = 10; i > 0; i--) {
            System.out.println(i + " ");
        }
        System.out.println();
        System.out.println("Acabou");
    }
}

```

Results    Output - Repeticoes01 (run) ×

run:

10  
9  
8  
7  
6  
5  
4  
3  
2  
1

Acabou

## Exemplo – Comando for

# Comandos de Repetição

- while: é outro laço condicional, isto é, um conjunto de instruções que são repetidas enquanto o resultado de uma expressão lógica ( uma condição ) é avaliada como verdadeiro.

```
while ( < condição > )  
{  
    < conjunto de instruções >  
}
```

# Comandos de repetição

- do-while: muito parecido com o while, porém o conjunto de instruções é executado antes da avaliação da expressão lógica. Isso faz com que essas instruções possam ser executadas pelo menos uma vez.

do

{

< conjunto de instruções >

} while ( < condição > );



# Exemplo – while e do-while

```
package repeticoes02;
public class Repeticoes02 {
    public static void main(String[] args) {
        int a;
        try {
            a = Integer.parseInt(args[0]);
            while (a >= 0) {
                System.out.println(" " + a);
                a--;
            }
            System.out.println("Fim da primeira contagem");
            System.out.println();
            a = (-1) * Integer.parseInt(args[0]);
            do {
                System.out.println(" " + a);
                a++;
            } while (a <= 0);
            System.out.println("Fim da segunda contagem");
        } catch (ArrayIndexOutOfBoundsException erro) {
            System.out.println("Digite um argumento!");
        } catch (NumberFormatException erro) {
            System.out.println("Nao foi fornecido um número inteiro válido!");
        }
    }
}
```

```
run:
3
2
1
0
Fim da primeira contagem

-3
-2
-1
0
Fim da segunda contagem
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Dúvidas?

