A decorative wavy line in yellow and white on the left side of the slide.

AULA 06 - ESTRUTURAS DE REPETIÇÃO

**MICHELLE NERY NASCIMENTO
PUC MINAS**

ALGORITMOS E TÉCNICAS DE PROGRAMAÇÃO

A decorative wavy line in yellow and white runs vertically along the left side of the page.

AGRADECIMENTO

**À PROFESSORA SORAIA LÚCIA PELA
CESSÃO DO MATERIAL DESTE GUIA DE
AULA**

ESTRUTURAS DE REPETIÇÃO

- Permitem que uma sequência de comandos seja executada repetidamente, até que uma condição de interrupção seja satisfeita.
- **Exemplo:** Como imprimir os 1000 primeiros números a partir de 1?

ESTRUTURAS DE REPETIÇÃO

- Permitem que uma sequência de comandos seja executada repetidamente, até que uma condição de interrupção seja satisfeita.
- **Exemplo:** Como imprimir os 1000 primeiros números a partir de 1?

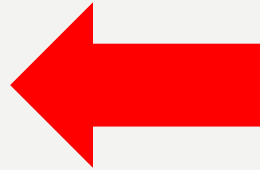
Escrever o comando `System.out.println()` 1000 vezes não é prático!!!

ESTRUTURAS DE REPETIÇÃO

- Permitem que uma sequência de comandos seja executada repetidamente, até que uma condição de interrupção seja satisfeita.

SUMÁRIO

- Introdução
- Comando while
- Comando do-while
- Comando for



COMANDO WHILE - SINTAXE

```
while (expressão) {  
  
    lista de comandos;  
  
}
```

COMANDO WHILE

- Primeiro, avalia-se a expressão;
- Se verdadeira, executa-se a lista de comandos;
- Ao término da lista de comandos, reavalia-se a expressão;

```
while (expressão) {  
    lista de comandos;  
}
```

- **O processo é repetido até que a expressão seja falsa.**

COMANDO WHILE

- O corpo de um comando ***while*** pode ter **zero, um ou n** comandos;
- O { e o } são obrigatórios apenas se tivermos mais de um comando.
- O corpo de um comando ***while*** pode ser executado **zero ou mais vezes**.

COMANDO WHILE - EXEMPLO

```
int i = 1;  
  
while (i <= 1000) {  
    System.out.println(i);  
    i ++;  
}
```

EXEMPLO 1

- Faça um programa que exiba quantas pessoas possuem mais de 18 anos. O algoritmo deverá ler a idade de 10 pessoas.

```
import java.util.*;
public class Switch Exemplo{
    Scanner sc = new Scanner (System.in);
    public static void main(string[ ] args){
        int qtde = 0;
        int idade = 0;
        int i = 0;
        while (i <= 9) {
            i = i + 1;
            System.out.println("Informe a idade da pessoa:", i);
            idade = sc.nextInt();
            if (idade >= 18) {
                qtde = qtde + 1;
            }
        }
        System.out.println("Pessoas com mais de 18 anos", qtde);
    }
}
```

EXEMPLO 2

- Escreva um programa que faz a leitura de um valor, calcule este número multiplicado por 3 e o imprima. O programa continuará a pedir novos valores com base na decisão do usuário para continuar que deverá ser "sim".

```

import java.util.*;
public class Exemplo{
    static void Main(string[] args) {
        Scanner sc = new Scanner (System.in);
        double cubo, n;
        String resposta = "sim";
        while (resposta.equals( "sim")) {    //String em Java é objeto,
                                                então o == costuma não funcionar
            System.out.println("Informe um número qualquer:");
            n =sc.nextDouble();
            sc.nextLine(); //para descartar o "\n"
            cubo= n * 3;
            System.out.println("O resultado do número n multiplicado
                                por 3 é:“ + cubo);
            System.out.println("Digite—sim—para continuar ou
                                pressione qualquer tecla para sair");
            resposta = sc.nextLine();
        }
    }
}

```

EXEMPLO 3

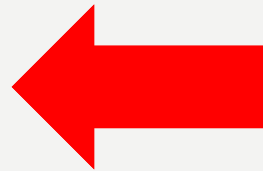
- Escreva um programa que calcule a seguinte série:

$$S = 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50$$

```
static void main(string[] args)
{
    int numerador = 1, denominador = 1;
    double s = 0, parc;
    while (denominador <= 50)
    {
        parc = (double)numerador/denominador;
        s += parc;
        numerador += 2;
        denominador++;
    }
    System.out.println("A soma da série é: " + s);
}
```


SUMÁRIO

- Introdução
- Comando while
- **Comando do-while**
- Comando for



COMANDO DO-WHILE

- Similar ao comando ***while***:
 - a diferença é o momento em que a expressão é avaliada;
- No comando ***do-while***, a lista de comandos é executada e, depois, a expressão é avaliada;
- O corpo de um comando ***do-while*** é executado **pelo menos uma vez.**

COMANDO DO-WHILE

```
do {
```

```
    lista de comandos;
```

```
} while (expressão) ;
```

COMANDO DO-WHILE - EXEMPLO

```
int i = 1;  
  
do {  
    System.out.println(i);  
    i ++;  
} while (i <= 1000) ;
```

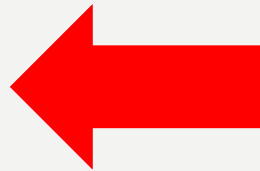
EXEMPLO 1

- Faça um programa que exiba quantas pessoas possuem mais de 18 anos. O algoritmo deverá ler a idade de 10 pessoas.

```
public static void main(string[ ] args){
    Scanner sc = new Scanner();
    int qtde = 0;
    int idade = 0;
    int i = 0;
    do {
        i = i + 1;
        System.out.println("Informe a idade da pessoa:", i);
        idade = sc.nextInt();
        if (idade >= 18) {
            qtde = qtde + 1;
        }
    } while (i < 10);
    System.out.println("Pessoas com mais de 18 anos", qtde);
}
}
```

SUMÁRIO

- Introdução
- Comando while
- Comando do-while
- Comando for



COMANDO FOR

```
for (início; expressão; incremento)
{
    lista de comandos;
}
```

- Primeiro, executa-se o código contido em início;
- Segundo, avalia-se a expressão;
- Se a expressão for verdadeira, executa-se a lista de comandos. Senão, encerra o comando **for**.
- Ao término da lista de comandos, executa-se os comandos de incremento;
- Reavalia-se a condição e repete-se o processo enquanto a mesma for verdadeira.

COMANDO FOR

```
for (início; expressão; incremento)
{
    lista de comandos;
}
```



```
início;
while (expressão)
{
    lista de comandos;
    incremento;
}
```

COMANDO FOR - EXEMPLO

```
for (int i = 1; i <= 1000; i++)  
{  
    System.out.println(i);  
}
```

```
for (int i = 1; i <= 1000; i++)  
    {  
        System.out.println(i);  
    }
```

```
int i = 1;  
  
do {  
    System.out.println(i);  
    i++;  
} while (i <= 1000) ;
```

```
int i = 1;  
  
while (i <= 1000) {  
    System.out.println(i);  
    i++;  
}
```

- Algumas variações do **for**:

- O laço **for** pode seguir em sentido positivo ou negativo e mudar a variável de controle de laço de acordo com qualquer valor. O exemplo abaixo que exibe números de 0 a -95 em decrementos de 5.

```
public class ForExemplo{  
    public static void main(string[ ] args){  
        int x;  
        for (x = 0; x > -100; x -= 5) {  
            System.out.println("valor de x:" + x);  
        }  
    }  
}
```

- Uso de diversas variáveis de controle, que são separadas por vírgula. Na prática, mais de duas ou três tornam o laço **for** difícil de controlar.

```
public class ForExemplo2{  
    public static void main(string[ ] args){  
        int i, j;  
        for (i = 0, j = 0; i < j ; i++, j--) {  
            System.out.println("valor de i:" + i + "valor de j:" + j);  
        }  
    }  
}
```

- Que tipo de laço utilizar?
 - Use um laço for para executar um número conhecido de iterações;
 - Use o do-while quando precisar de um laço que execute sempre pelo menos uma iteração;
 - Use o laço while quando é repetido até alguma condição ser falsa.

EXEMPLO 1

- Faça um programa que exiba quantas pessoas possuem mais de 18 anos. O algoritmo deverá ler a idade de 10 pessoas.


```
import java.util.*;
public class Switch Exemplo{
    Scanner sc = new Scanner (System.in);
    public static void main(string[ ] args){
        int qtde = 0;
        int idade = 0;
        for (int i = 1; i <= 10; i++) {
            System.out.println("Informe a idade da pessoa:", i);
            idade = sc.nextInt();
            if (idade >= 18) {
                qtde = qtde + 1;
            }
        }
        System.out.println("Pessoas com mais de 18 anos", qtde);
    }
}
```

Obs.: Observe que a variável de controle foi declarada dentro do **for** e, portanto, é conhecida somente dentro do laço **for** .

EXEMPLO 2

- Escreva um programa que calcule a seguinte série:

$$S = 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50$$

```
static void main(string[] args)
{
    int num = 1, den = 1;
    double s = 0, parc;
    for (den = 1; den <= 50; den ++)
    {
        parc = (double)numerador/denominador;
        s += parc;
        numerador += 2;
    }
    System.out.println("A soma da série é: " + s);
}
```

EXEMPLO 3

- Faça um programa que lê um número inteiro (N) fornecido pelo usuário e calcule/imprima a soma dos números entre 1 e N .

```
static void main(string[] args)
{
    int n, soma = 0;
    System.out.println("Informe um número: ");
    n = sc.nextInt();
    for (int cont = 1; cont <= n; cont++)
    {
        soma = soma + cont;
    }
    System.out.println("A soma é: " + soma);
}
```

EXERCÍCIOS

1) Escreva um programa que calcule a seguinte série:

$$S = 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50$$

2) Faça um programa que leia 10 valores quaisquer, um de cada vez, e conta quantos destes valores são negativos.

3) Ler o número de alunos existentes em uma turma, ler as 3 notas destes alunos e calcular a média aritmética destas notas.

4) Faça um algoritmo para ler uma quantidade indeterminada de valores inteiros. Para cada valor fornecido escrever uma mensagem que indica se cada valor fornecido é PAR ou ÍMPAR. O algoritmo será encerrado imediatamente após a leitura de um valor NULO (zero) ou NEGATIVO.

EXERCÍCIOS

- 5) Faça um programa que calcule e imprima a tabuada de um número qualquer.
- 6) Faça um programa que leia 10 valores inteiros quaisquer, um de cada vez, e conta quantos destes valores são negativos.
- 7) Escreva um programa que calcule o valor do produto:

$$\left(1 - \frac{1}{2 \times 3}\right) \times \left(1 - \frac{1}{3 \times 4}\right) \times \left(1 - \frac{1}{4 \times 5}\right) \times \dots \times \left(1 - \frac{1}{99 \times 100}\right)$$

AVISO LEGAL

- O material presente nesta apresentação foi produzido a partir de informações próprias e coletadas de documentos obtidos publicamente a partir da Internet. Este material contém ilustrações adquiridas de bancos de imagens de origem privada ou pública, não possuindo a intenção de violar qualquer direito pertencente à terceiros e sendo voltado para fins acadêmicos ou meramente ilustrativos. Portanto, os textos, fotografias, imagens, logomarcas e sons presentes nesta apresentação se encontram protegidos por direitos autorais ou outros direitos de propriedade intelectual.
- Ao usar este material, o usuário deverá respeitar todos os direitos de propriedade intelectual e industrial, os decorrentes da proteção de marcas registradas da mesma, bem como todos os direitos referentes a terceiros que por ventura estejam, ou estiveram, de alguma forma disponíveis nos slides. O simples acesso a este conteúdo não confere ao usuário qualquer direito de uso dos nomes, títulos, palavras, frases, marcas, dentre outras, que nele estejam, ou estiveram, disponíveis.
- É vedada sua utilização para finalidades comerciais, publicitárias ou qualquer outra que contrarie a realidade para o qual foi concebido. Sendo que é proibida sua reprodução, distribuição, transmissão, exibição, publicação ou divulgação, total ou parcial, dos textos, figuras, gráficos e demais conteúdos descritos anteriormente, que compõem o presente material, sem prévia e expressa autorização de seu titular, sendo permitida somente a impressão de cópias para uso acadêmico e arquivo pessoal, sem que sejam separadas as partes, permitindo dar o fiel e real entendimento de seu conteúdo e objetivo. Em hipótese alguma o usuário adquirirá quaisquer direitos sobre os mesmos.
- O usuário assume toda e qualquer responsabilidade, de caráter civil e/ou criminal, pela utilização indevida das informações, textos, gráficos, marcas, enfim, todo e qualquer direito de propriedade intelectual ou industrial deste material.