



GLADIOS

MORDEKAISER



©PHOTO360

Construção de Algoritmos

Iniciamos em alguns minutos...

Aula 03 - 01

Instrução de Repetição

Prof. Luciano Freire

Comando de repetição



🌈 Usado em situações em que é necessário repetir um determinado trecho de um programa várias vezes

🌈 **Duas formas:**

🌈 Escrever o trecho quantas vezes for necessário

OU

🌈 Utilizar o conceito de Looping (REPETIÇÕES)

Considere o seguinte problema



 Ler a nota de 10 alunos

Estrutura de Repetição






```
double n1, n2, n3, n4, n5, n6, n7, n8, n9, n10;  
Scanner entrada = new Scanner(System.in);
```

```
System.out.println("Nota do aluno 1");  
n1 = entrada.nextDouble();  
System.out.println("Nota do aluno 2");  
n2 = entrada.nextDouble();  
System.out.println("Nota do aluno 3");  
.....  
n7 = entrada.nextDouble();  
System.out.println("Nota do aluno 8");  
n8 = entrada.nextDouble();  
System.out.println("Nota do aluno 9");  
n9 = entrada.nextDouble();  
System.out.println("Nota do aluno 10");  
n10 = entrada.nextDouble();
```

*Muito
Trabalhoso*

Leitura da Nota de
10 alunos

Vantagem:

-  O algoritmo passa a ter um tamanho menor
-  Aumentando a amplitude de processamento
-  Sem alterar o tamanho do código

```
double nota;  
int i;  
i = 0;  
while (i < 10)  
{  
    System.out.println("Nota do aluno" + (i+1));  
    nota = entrada.nextDouble();  
    i++;  
}
```

Formas - repetições FINITAS:

 Número conhecido de vezes

 Sabe-se quantas vezes irá repetir

 Número não conhecido de vezes

 Não sabe-se quantas vezes irá repetir

Usuário é quem
define a
quantidade de
repetições

WHILE (enquanto)



- ❖ O comando avalia uma **condição** inicial
- ❖ Se **condição** for **VERDADE** → executa o bloco de instruções
- ❖ Se **condição** for **FALSO** →, não executa o bloco de instruções
- ❖ E segue o fluxo de execução do algoritmo

WHILE (enquanto)



Sintaxe do comando:

Bloco de Instruções

```
while ( condição )  
{  
    instruções_dentro_do_Laço;  
    instruções_dentro_do_Laço;  
    instruções_dentro_do_Laço;  
}  
próxima_instrução_do_algoritmo;
```

Contador – O que ele faz??



🌀 Repetição com **número finito** e **conhecido** de vezes

🌀 É necessário usar uma variável de controle:



$$1 + 1 =$$

$$2 + 1 = 3 + 1 = 4 + 1 = 5...$$

🌀 Mas o que o contador FAZ?

🌀 Ele conta (geralmente) de um em um (acrescenta ou diminui)

🌀 É ele que irá contar quantas vezes o trecho irá se repetir

🌀 Portanto, é necessário iniciar o contador

– antes de entrar no laço

Exemplo 1



- Faça um algoritmo que repita 5 vezes a leitura de dois valores, some-os e mostre o resultado em tela

Exemplo 1



```
class exemplo01{  
    public static void main(String[] args){  
        int a, b, r;  
        int cont;  
        Scanner entrada = new Scanner(System.in);  
        cont = 0;  
        while (cont < 5)  
        {  
            System.out.println("\nEntre com o valor de a e b: ");  
            a = entrada.nextInt();  
            b = entrada.nextInt();  
            r = a + b;  
            System.out.println("Resultado é: " + r);  
            cont = cont + 1;  
        }  
    }  
}
```

inicialização

condição

Incrementa o contador

cont	cont < 5	a	b	r
0	TRUE	3	4	7
1	TRUE	2	8	10
2	TRUE	5	3	8
3	TRUE	2	2	4
4	TRUE	3	2	5
5	FALSE	Sai do loop		

- Repetição com **número finito** e **não conhecido** de vezes

 - O usuário pode determinar quando encerrar o algoritmo!!

Exemplo2

- Faça um algoritmo que execute repetidamente a leitura de dois números, some-os, e mostre o resultado. Porém, a cada resultado encontrado, pergunte ao usuário se ele deseja calcular outra soma

 - Nesse caso, não há como saber quantas vezes a execução se repetirá

 - E, qual a condição que devo usar?

 - Uma comparação com a resposta do usuário!!

WHILE



```
class exemplo01{
    static void main(String[] args)    {
        int a, b, r;
        String resp;

        resp = "s";
        while (resp.equals("s"))
        {
            System.out.println("\nEntre com o valor de a e b: ");
            a = entrada.nextInt();
            b = entrada.nextInt();
            r = a + b;
            System.out.println("Resultado é:" + r);
            System.out.println("Deseja continuar?(s/n): ");
            resp = entrada.next();
        }
    }
}
```

inicialização

condição

Mudança do valor inicial

resp	resp == "s"	a	b	r
s	TRUE	3	4	7
s	TRUE	2	8	10
s	TRUE	5	3	8
n	FALSE	Sai do loop		

Exercícios – Repetição



1 – Faça um algoritmo que imprima a sequência de números abaixo:

1, 2, 3, 4, 5, ..., 98, 99, 100

2 – Faça um algoritmo que calcule a soma dos 100 primeiros números inteiros.

1+2+3+4+5+ ... +100

3 – Faça um algoritmo que apresente os resultados de uma tabuada para um número qualquer informado pelo usuário. Exemplo: tabuada do número 5

- ✚ 4 – Construir um algoritmo que leia um número inteiro maior que zero e imprima a sequência: 1, 2, 4, 8, 16, 32,... enquanto o valor for menor ou igual ao valor lido.

Exemplo:

Valor lido: 82

Sequência impressa: 1, 2, 4, 8, 16, 32, 64

- 5 - Escreva um algoritmo que gere uma tabela com a conversão de graus Celsius para Fahrenheit e vice versa, com valores variando de 1 em 1 grau, de 0 a 50 graus Celsius

$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) / 1,8$$

- 6 – Fazer um algoritmo que leia o valor de uma temperatura. Validar se a temperatura informada é maior que 0 e menor que 70 °.
- Solicitar novamente a leitura da temperatura enquanto estiver for a do intervalo válido!

- 🧩 Construir um algoritmo para imprimir a série de FIBONACCI até o 10º termo. A série de FIBONACCI é formada pela seqüência:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55

- 🧩 **Obs: utilizar estrutura de repetição!**

x
 $x^2 + py$

Muito Obrigado!

Prof. Luciano Freire
luciano.freire@facens.br

