**ESTRUTURA DE REPETIÇÃO**

- Os computadores tem a habilidade de poder repetir uma operação ou uma série de operações diversas vezes.

-Cada repetição é chamada de laço (loop)

- Uma estrutura de repetição permite que uma sequência de comandos seja executada repetidamente, CASO DETERMINADAS CONDIÇÕES SEJAM SATISFEITAS

- Essas condições são representadas por expressões logicas ( boolean – true ou false)

- As estruturas de repetição podem ser classificadas em:

**\*REPETIÇÃO COM TESTE DE INÍCIO (While):** É preciso saber se é true ou false. Se for true inicia com a expressão, vai pra sentença e volta pra expressão novamente, fazendo o mesmo ciclo repetidamente. Se for false, inicia na expressão e encerra diretamente. // NO WHILE PRIMEIRO PASSA PELA EXPRESSAO P PODER EXECUTAR OS COMANDOS – A DEPENDER SE FOR TRUE OU FALSE

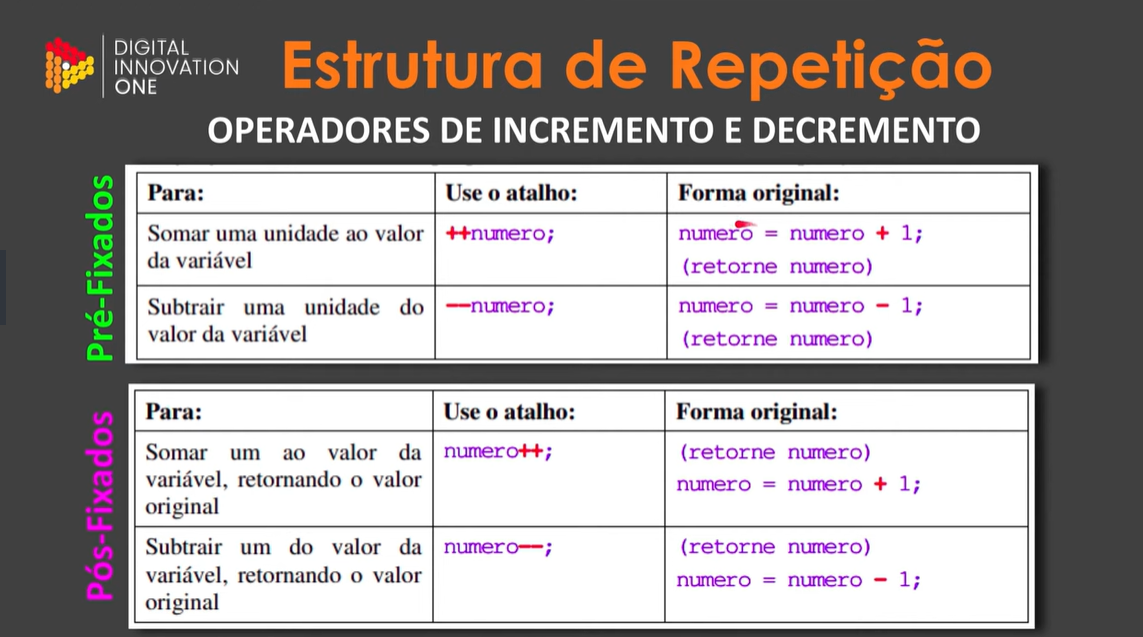
**\*REPETIÇÃO COM TESTE NO FINAL (Do-while):** Inicia na sentença, que vai ser executada independente de true ou false. Depois vai p expressao – se for true ele volta pra sentença p ser executado mais uma vez e assim repetidamente enquanto for true. Se for false depois da expressão encerra-se o laço de repetição imediatamente. // NO DO-WHILE PRIMEIRO EXECUTA OS COMANDOS INDEPENDENTE DE SER TRUE OU FALSE (PELO MENOS UMA VEZ, JÁ QUE SE INICIA COM A SENTENÇA), E DEPOIS VAI P EXPRESSAO – Q SE FOR FALSE ENCERRA O LAÇO E SE FOR TRUE CONTINUA

**\*REPETIÇÃO CONTADA (For):** O programa inicia e é feito de inicio um teste, se as condições desse teste forem satisfeitas (true, 1) aí executa a sentença e depois abre uma atualização. Depois da atualização, volta pro teste novamente e o ciclo se repete caso seja true – condições satisfeitas no teste. Quando no teste o resultado dá false, o ciclo se encerra imediato.

- O comando BREAK é usado para terminar de forma abrupta uma repetição

- Quando o comando CONTINUE é executado, os comandos restantes da repetição são ignorados e o programa volta a testar novamente ou não. – se tiver a palavra continue no meio da estrutura de repetição ele não continua executando, ele volta pro começo p testar novamente ou não – depende se for true ou false

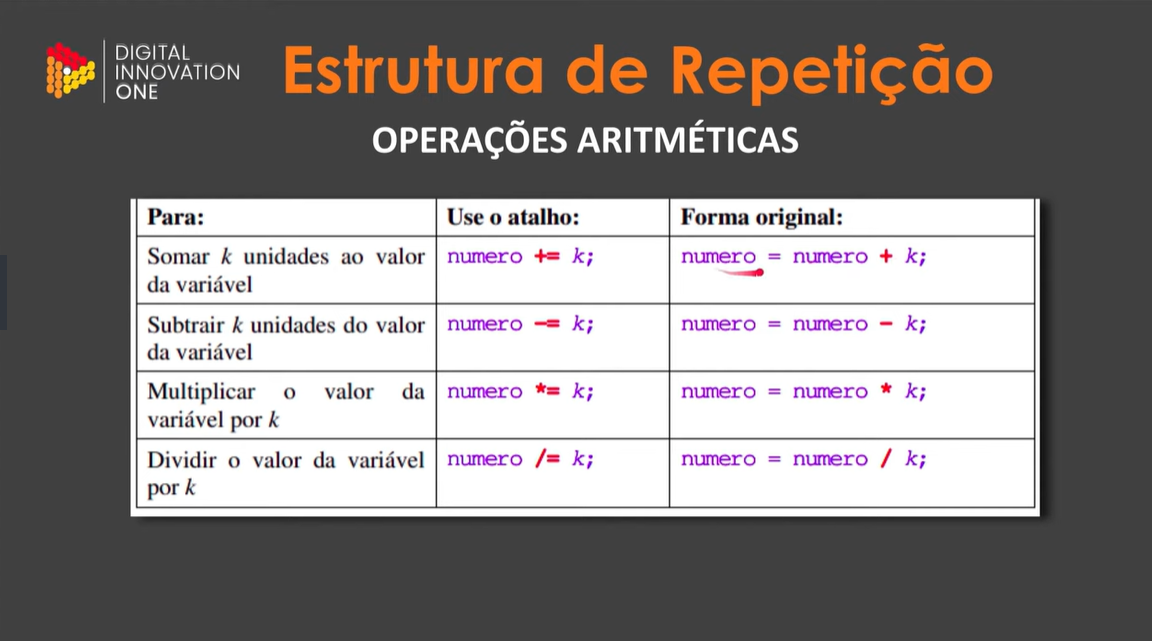
**OPERADORES DE INCREMENTO E DECREMENTO:**



P/ somar/subtrarir uma unidade ao valor da variável, pode-se usar o atalho ++ ou -- antes do nome da variável. Ex: ++numero; ou –numero;

p/ somar ou subtrair uma unidade ao valor da variável, retornando o valor original, deve-se usar o atalho ++ ou – DEPOIS do nome da variável. Ex: numero++; ou numero--;

**OPERAÇÕES ARITIMÉTICAS**



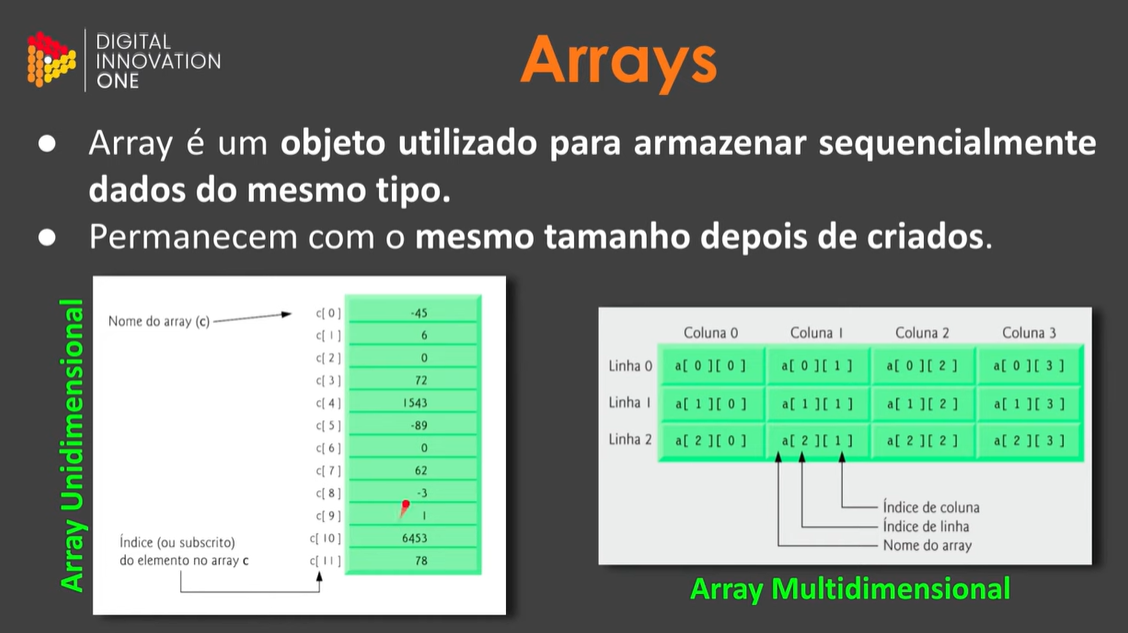
K= determinada quantidade, poderia ser qq valor

**ARRAYS**

- Um objeto usado p armazenar sequencialmente dados do mesmo tipo. ( é como se tivéssemos uma variável e dentro dessa variável colocássemos vários dados. Por ex uma turma de sala de aula – TurmaX tem 47 alunos. P cada aluno eu teria que registrar um dado, ex: aluno1 = maria; aluno2= joão, etc. Com o arrey eu consigo armazenar todos os alunos – na sequencia q forem registrados – na variável TurmaX).

-Os dados do arrey tem que ser do mesmo tipo (idade, nome, altura, etc)

-Uma vez criado o arrey, ele permanece p sempre do mesmo tamanho. Pois qdo vamos criar o arrey ele pede quantos elementos terão dentro e esse numero não poderá ser modificado. Eu posso criar um arrey e colocar 10 posiçoes mas usar 5. Mas não posso criar um arrey de 10 posicoes e querer colocar 11 dentro.



- As posições iniciam com o numero ZERO

- nome do arrey (c) / índice do elemento no arrey: c[3]

- NO MULTIDIMENSIONAL: índice do elemento no arrey: a[1][3] 🡪 o elemento está na linha 1, coluna 3 do arrey a

- A CLASSE DE DADOS “ Scanner scan = new Scanner(System.in); “ é a melhor classe pra usar qdo pede entrada de dados pelo teclado!!!

- DICA: QDO PRECISAR USAR UM CONTADOR, USA-SE O LAÇO “FOR” – e nele se usa o “i” que é igual o “count” do while. TEM SEMPRE Q CRIAR A VARIAVEL DO CONTADOR

- É MAIS COMUM USAR O WHILE E O FOR, O DO-WHILE NÃO É TAO USADO

**EXERCÍCIOS**

LOOPS

1. NomeIdade

Faça um programa que leia conjuntos de dois valores, o primeiro representando o nome do aluno e o segundo representando a sua idade. Pare o programa inserindo o valor 0 no campo nome. Ele quer que fique pedindo esse conjunto de valores e que pare quando colocar 0 no campo “nome”.

public static void main(String[] arg) {

Scanner scan = new Scanner(System.in);

String nome;

int idade;

System.out.println(“Nome: ”);

System.out.println(“Idade: ”);

{

while(true) {

            System.out.println("Nome: ");

            nome = scan.next();

            if (nome.equals("0")) break;

            System.out.println("Idade: ");

            idade = scan.nextInt();

1. Notas

Faça um programa que peça uma nota entre zero e dez.

Mostre uma msg caso o valor seja invalido e continue pedindo até q o usuário informe um valor valido.

package Loops;

import java.util.Scanner;

public class Notas {

public static void main(String[] args) {

    Scanner scan = new Scanner(System.in);

    int nota;

    System.out.println("Nota: ");

    nota = scan.nextInt();

    while(nota < 0 || nota > 10) {

        System.out.println("Nota inválida. Digite novamente");

        nota = scan.nextInt();

    }

}

}

1. MaiorEMedia

Faça um programa que leia 5 numeros e informe o maior numero e a media desses números.

Programa que leia 5 numeros – Entao tem que iniciar com a classe SCANNER, depois atribuir “int numero;”, depois “System.out.println(“Número: “); , depois “numero = scan.nextInt();”. (até aqui temos um prog que pede um numero p usuário e encerra qdo ele digita. DEPOIS cria-se a variável “ int count = 0” (contador p contar os 5 numeros q foi pedido no enunciado, mas n entendi pq = 0!, é oq controla o loop).

Laço de repetição – Usar o DO-WHILE. “do { System.out.println(“Número: “);

numero = scan.nextInt();

count = count + 1;

} while(count <5 );

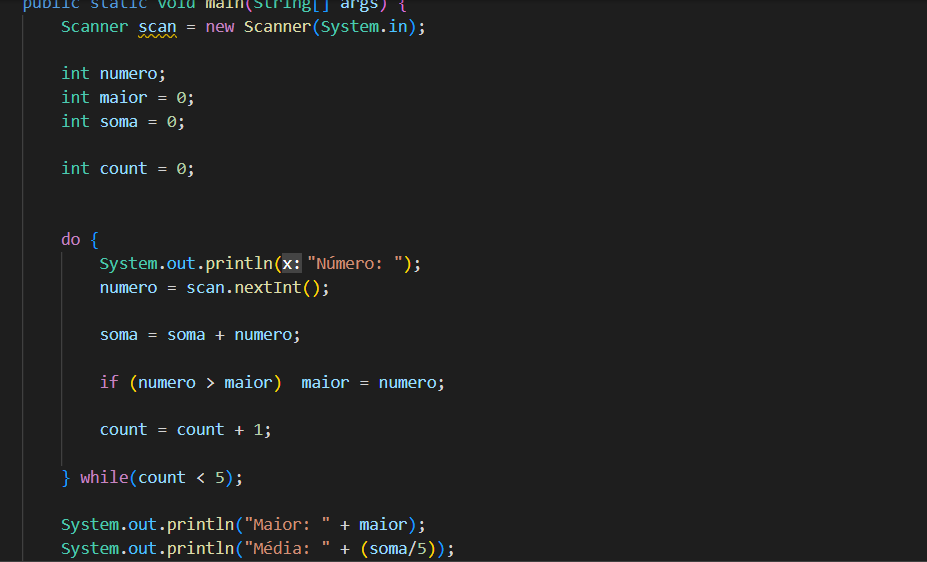
Que informe o maior número – tem que colocar a variável “ int maior = 0” ( n entendi pq)

if (numero > maior) {

maior = numero;

}

soma = soma + numero



1. ParImpar

Faça um programa que peça N números inteiros. Calcule e mostre a qtd de números pares e a qtd de números impares

1. Tabuada – usando o laço FOR

Desenvolva um gerador de tabuada, capaz de gerar tabuada de qq numero inteiro entre 1 a 10. O usuário deve informar de qual numero ele quer a tabuada. A saída deve ser conforme o exemplo abaixo:

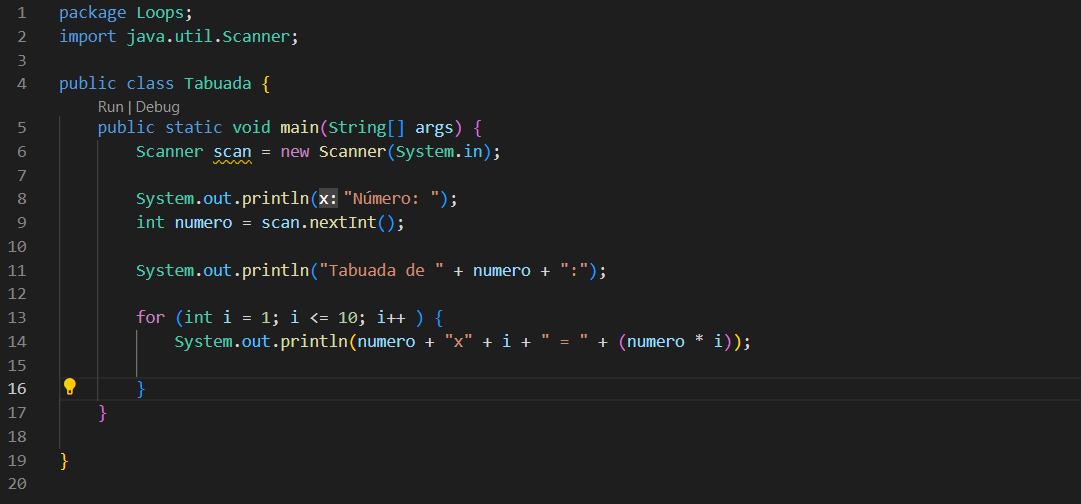
Tabuada de 5:

5x1= 5

5x2 = 10

....

5x10 = 50



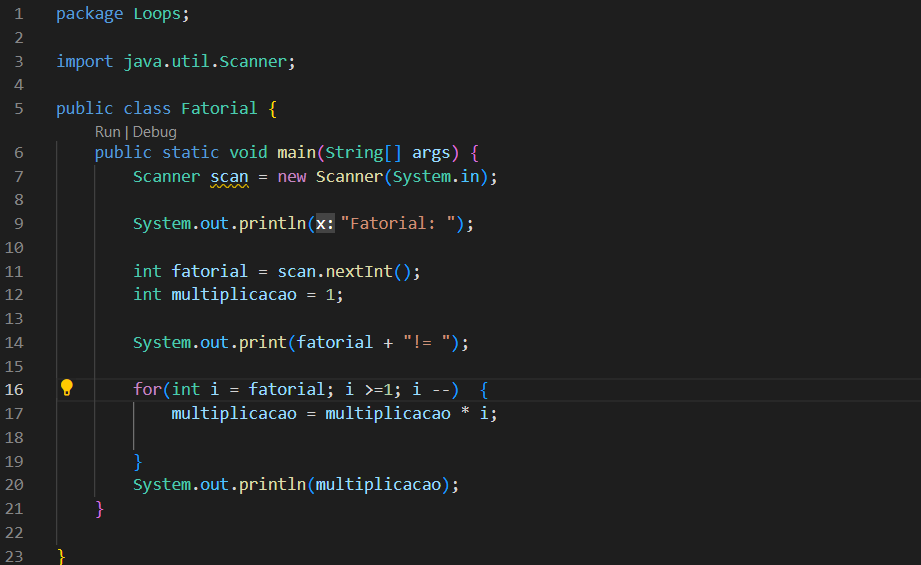
O “i” é igual o “count”, é o contador, oq controla o loop

1. FATORIAL – usando laço FOR

Faça um programa que calcule o fatorial de um numero inteiro fornecido pelo usuário.

(fatorial é qdo pegamos um numero e decompõe ele multiplicando)

Ex: 5! = 120 (lê-se fatorial de 5 =120. Pq vc pega o cinco e multiplica: 5x4x3x2x1 = 120



Scanner scan... 🡪 é pra abrir o fluxo do programa //

**ARREYS**

- é um vetor

- Podem ser unidimensionais e multidimensionais (várias linhas e várias colunas)

- as posições começam sempre do numero ZERO e não do UM

EX:

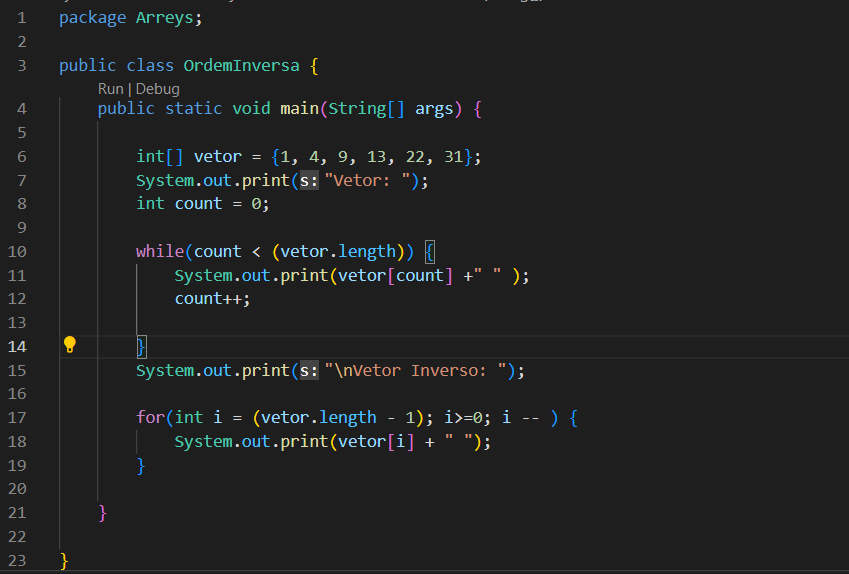
int[] vetor = { 1, 4, 8, 22, 31, 24}

**EXERCÍCIOS**

1. OrdemInversa – usando o laço while

Crie um vetor de 6 numeros inteiros e mostre-os na ordem inversa.

Vetor.lenght-1 (pq n são 6 posicoes e sim 5)



1. CONSOANTES

Faça um programa que leia um vetor de 6 caracteres (ou seja criar um arrey de 6 caracteres – do tipo String) e digaa quantas consoantes foram lidas. Imprima as consoantes.

- se o enunciado pede 6 caracteres, então eu devo iniciar meu vetor com 6 elementos

-qd se cria um arrey ele vai tar preenchido c alguma coisa. Como no caso são 6 elementos mas eu pedi q ele dissesse somente quais consoantes foram lidas, ele imprime: “ null null b null c null” por ex

- começar criando a String[] consoantes = new String[6]; / ou seja existem 6 elementos no arrey

-criar variáveis: int quantidadeConsoantes = 0

int count = 0;

- se ele pede que diga qts consoantes foram lidas, eu tenho que criar um controlador pra que se faça isso.

do{

} while( count < consoantes.length)

- lê-se FAZER ISSO que está entre chaves – DO – ATÉ QDO oq tiver em parênteses.

- em JAVA, uma String é um objeto

- o laço for se lê: para cada elemento dentro desse meu arrey, eu quero q vc imprima esse elemento

1. NumerosAleatorios

Faça um programa que leia 20 numeros inteiros aleatórios (entre 0 e 100) , armazene-os num vetor. Ao final mostre os números e seus sucessores

- Existe uma CLASSE p números aleatórios – RANDOM // usa-se “ Random random = new Randon(); “ p/ abrir o fluxo quando se quer números aleatoriios

- depois criar arrey p guardar os 20 numeros aleatórios:

int[] numerosAleatorios = new int[20];

-depois criar um laço pra q se coloque esses 20 valores dentro de numerosAleatorios + um limite (bound) para os números que será 100:

for(int i = 0; i <  numerosAleatorios.length; i++) {

            int numero = random.nextInt(100);

1. ARREYS MULTIDIMENSIONAIS

Gere e imprima uma matriz M 4X4 com valores aleatórios entre 0-9

- primeiro main

- depois abre o fluxo de números aleatórios:

        Random random = new Random();

- depois criar o arrey: