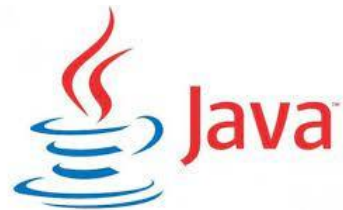


FORMAÇÃO JAVA

AULA 02

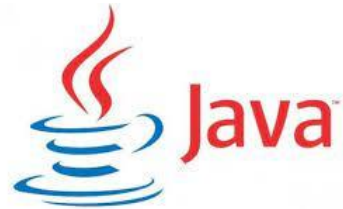
Professor: Leonardo Gomes.
WhatsApp/Telegram: (85) 98421-8509
Email: leonardobrendoti@gmail.com
IwTraining formações.

SUMÁRIO.



1. interação via console;
2. Operadores;
3. Estruturas condicionais;
4. Operador ternário;
5. Métodos.

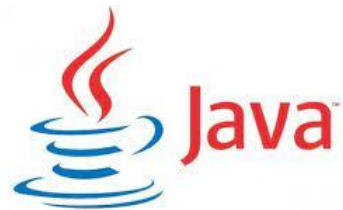
INTERAÇÃO VIA CONSOLE.



A classe **Scanner** é responsável por receber dados via console, sendo amplamente utilizada para operações I/O.

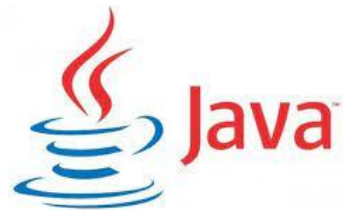
Exemplo: `Scanner entrada = new Scanner(System.in)`

INTERAÇÃO VIA CONSOLE.



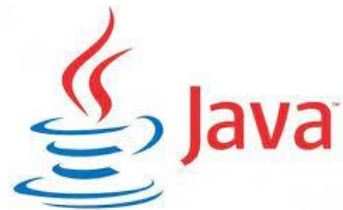
```
Main.java Principal.java X
1 import java.util.Scanner;
2 public class Principal {
3     public static void main(String[] args) {
4         Scanner entrada = new Scanner(System.in);
5         double numero1 = 0;
6         double numero2 = 0;
7         double resultado = 0;
8
9         System.out.println("Digite o número 1");
10        numero1 = entrada.nextDouble();
11        System.out.println("Digite o número 2");
12        numero2 = entrada.nextDouble();
13        resultado = numero1 + numero2;
14        System.out.println("A soma dos números digitados é: " + resultado);
15    }
16 }
```

INTERAÇÃO VIA CONSOLE - EXERCÍCIO



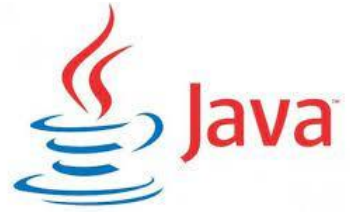
1. Receba 4 valores via console e faça a média aritmética.
2. Receba 4 valores via console e faça a média harmônica.
3. Receba 4 valores via console e faça a média ponderada (pode atribuir pesos aleatórios).
4. Receba o valor de x via console e calcule os polinômios:
 - a. $7x^4 + 3x^3 - 4x^2 + 10x - 3$
 - b. $x^4 + 4x^3 - 2x^2 + 14x - 9$
 - c. $5x^4 + 9x^3 - 6x^2 + 3x - 17$

OPERADORES NA LINGUAGEM JAVA.



1. Operador de atribuição;
2. Operadores aritméticos;
3. Operadores de incremento e decremento;
4. Operadores de igualdade;
5. Operadores relacionais;
6. Operadores lógicos;
7. Precedência dos operadores.

OPERADOR DE ATRIBUIÇÃO



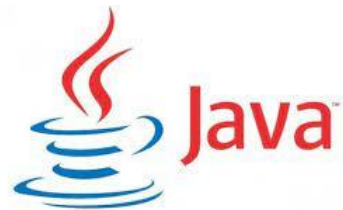
```
Main.java Principal.java ✕
1
2 public class Principal {
3     public static void main(String[] args) {
4         double nota1 = 7.1;
5     }
6 }
7
```

OPERADORES ARITMÉTICOS



+	operador de adição
-	operador subtração
*	operador de multiplicação
/	operador de divisão
%	operador de módulo (ou resto da divisão)

OPERADORES DE INCREMENTO E DECREMENTO.



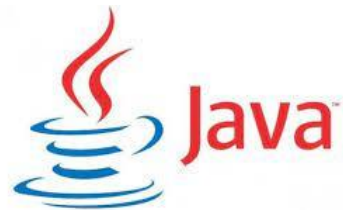
```
Main.java  Principal.java ✕  
1  
2 public class Principal {  
3     public static void main(String[] args) {  
4         double nota1 = 7.1;  
5         nota1++;  
6         //nota1 = nota1 + 1;  
7         nota1--;  
8         //nota1 = nota1 - 1;  
9         System.out.println(nota1);  
10    }  
11 }  
12
```

OPERADORES DE IGUALDADE.



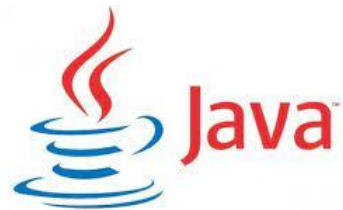
```
Main.java  Principal.java ✕  
1  
2 public class Principal {  
3     public static void main(String[] args) {  
4         double nota1 = 7.1;  
5         System.out.println(nota1 == 7);  
6         System.out.println(nota1 != 7);  
7     }  
8 }  
9
```

OPERADORES RELACIONAIS.



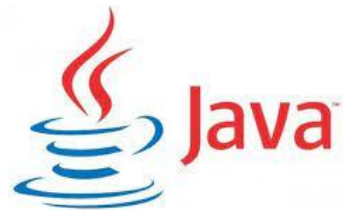
>	Utilizado quando desejamos verificar se uma variável é maior que outra.
>=	Utilizado quando desejamos verificar se uma variável é maior ou igual a outra
<	Utilizado quando desejamos verificar se uma variável é menor que outra.
<=	Utilizado quando desejamos verificar se uma variável é menor ou igual a outra.

OPERADORES LÓGICOS.



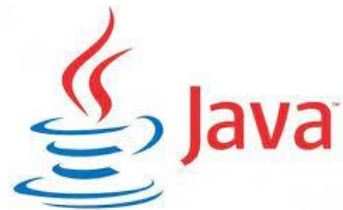
&&	Utilizado quando desejamos que as duas expressões sejam verdadeiras.
	Utilizado quando precisamos que pelo menos uma das expressões seja verdadeira.

PRECEDÊNCIA DOS OPERADORES



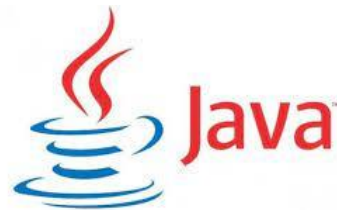
Operador	Operação	Ordem de avaliação(precedência)
* / %	Multiplicação Divisão Resto	Avaliado primeiro. Se houver vários operadores desse tipo serão avaliados da esquerda para a direita
+ -	Adição Subtração	Avaliado em seguida. Se houver vários operadores desse tipo, serão avaliados da esquerda para a direita.
=	Atribuição	Avaliado por último

ESTRUTURA CONDICIONAL - IF E ELSE.



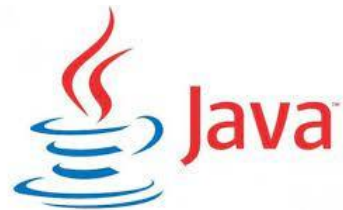
```
if (condição 1...){  
    escopo do if,  
}else if (condição 2...){  
    escopo do else if,  
}  
.  
.  
.  
else{  
    escopo do else.  
}
```

ESTRUTURA CONDICIONAL - IF E ELSE EXEMPLO.



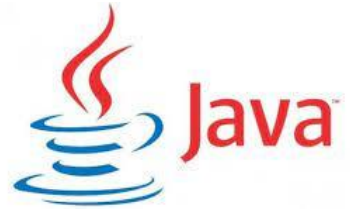
```
Main.java
1
2 public class Main {
3
4     public static void main(String[] args) {
5
6         int nota1 = 7;
7
8         if(nota1 > 7) {
9             System.out.println("Maior que 7");
10        }else if(nota1 < 7) {
11            System.out.println("Menor que 7");
12        }else {
13            System.out.println("Igual a 7");
14        }
15    }
16 }
17
```

ESTRUTURA CONDICIONAL - IF E ELSE ANINHADO.



```
if (condição 1...){
    escopo do if,
    if (condição 1.1 ...){
        escopo do if dentro do if.
    }else{
        escopo do else dentro do if.
    }
}else if (condição 2...){
    escopo do else if,
    if (condição 2.1 ...){
        escopo do if dentro do else if.
    }else if (condição 2.2 ...){
        escopo do else if dentro do else if.
    }else{
        escopo do else dentro do else if.
    }
}else{
    escopo do else.
    if (condição 3.1 ...){
        escopo do if dentro do else.
    }
}
```


ESTRUTURA COND. - IF,ELSE ANIN. - EXEMPLO.



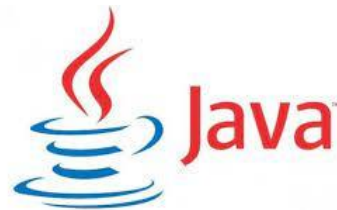
```
Main.java ✕
1
2 public class Main {
3     public static void main(String[] args) {
4         double nota1 = 7.1;
5         if(nota1 > 7) {
6             if (nota1 <= 8) {
7                 System.out.println("Até 8");
8             }else if(nota1 <= 9) {
9                 System.out.println("Até 9");
10            }else {
11                System.out.println("até 10");
12            }
13        }else if(nota1 < 7) {
14            System.out.println("Menor que 7");
15        }else {
16            System.out.println("Igual a 7");
17        }
18    }
19 }
```

ESTRUTURA CONDICIONAL - SWITCH.



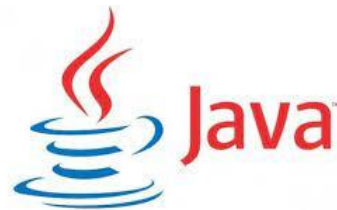
```
switch( opção ) {  
    case opção1:  
        comandos caso a opção 1 tenha sido escolhida  
        break;  
  
    case opção2:  
        comandos caso a opção 2 tenha sido escolhida  
        break;  
  
    case opção3:  
        comandos caso a opção 3 tenha sido escolhida  
        break;  
  
    default:  
        comandos caso nenhuma das opções anteriores tenha sido escolhida  
        break;  
}
```

ESTRUTURA CONDICIONAL - SWITCH - EXEMPLO.



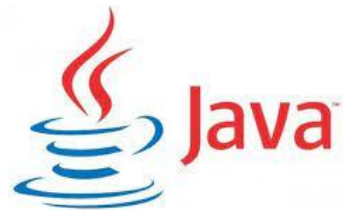
```
Main.java  Principal.java ✖
1
2 public class Principal {
3     public static void main(String[] args) {
4         int nota1 = 1;
5         switch (nota1) {
6             case 1:
7                 System.out.println("Opção 1");
8                 break;
9             case 2:
10                System.out.println("Opção 2");
11                break;
12             case 3:
13                System.out.println("Opção 3");
14                break;
15             default:
16                System.out.println("Outras opções");
17                break;
18         }
19     }
20 }
```

ESTRUT. COND. - SWITCH ANINHADO - EXEMPLO.



```
Main.java  Principal.java  X
1
2 public class Principal {
3     public static void main(String[] args) {
4         int nota1 = 1;
5         int indice = 8;
6         switch (nota1) {
7             case 1:
8                 System.out.println("Opção 1");
9                 if(indice < 7) {
10                     System.out.println("Indice menor que 7");
11                 }else {
12                     System.out.println("Indice maior que 7");
13                 }
14                 break;
15             case 2:
16                 System.out.println("Opção 2");
17                 break;
18             case 3:
19                 System.out.println("Opção 3");
20                 break;
21             default:
22                 System.out.println("Outras opções");
23                 break;
24         }
25     }
26 }
```

OPERADOR TERNÁRIO

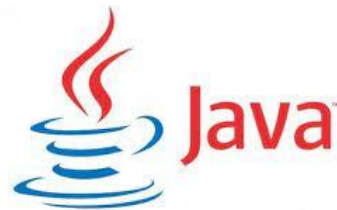


VARIÁVEL com/ ou sem condição? (Resposta, se sim) : (Resposta, se não);

ou

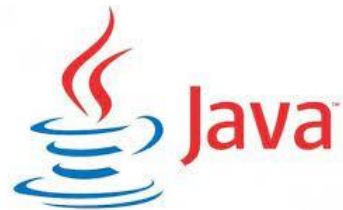
(expressão booleana) ? código 1 : código 2;

OPERADOR TERNÁRIO - EXEMPLO.



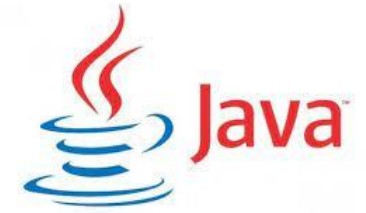
```
Main.java  Principal.java ✕  
1  
2 public class Principal {  
3     public static void main(String[] args) {  
4         int numero = 0;  
5         System.out.println((numero < 7 ) ? "menor que 7" : "maior que 7");  
6     }  
7 }
```

EXERCÍCIO ESTRUTURA CONDICIONAL E OPERADORES



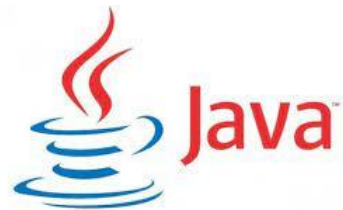
1. Use o operador ternário para identificar se a pessoa é maior ou menor de idade;
2. Ao receber a média aritmética de 5 números, indique se o aluno foi aprovado, reprovado e está de recuperação.
 - a. Aluno aprovado tem sua nota a partir de 7.
 - b. Aluno reprovado tem sua nota menor que 5.
 - c. Aluno de recuperação tem a seguinte situação:
 - i. Precisa fazer uma prova, se a média aritmética de sua nota inicial mais a nota da prova final for a partir de 5, o aluno passa na prova final, senão reprova na prova final;

MÉTODOS



Um **método** em java é um **comportamento**, **função**, **ação** ou **procedimento** da classe a qual o mesmo pertence.

MÉTODOS



O método em java pode ter a seguinte estrutura:

visibilidade + aspecto do método + tipo do método + nome + arg.

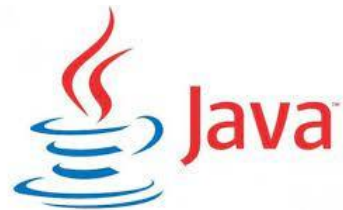
Exemplo 1: `public static void main(String[] args).`

Exemplo 2: `public void cantar(string som).`

Exemplo 3: `public int somar (int num1, int num2).`

Exemplo 4: `public String capturarNome(String mensagem).`

SOBRECARGA DE MÉTODOS



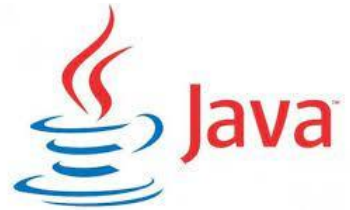
A sobrecarga de método é a invocação do método alterando a quantidade de parâmetros e o tipo do método.

```
public void subtrair(double total)
```

Exemplo 1: `public int subtrair(double total).`

Exemplo 2: `public void subtrair(double total, double num1).`

SOBRECARGA DE MÉTODOS - EXEMPLO



```
Main.java Principal.java ✕
13     }
14 }
15 class ClasseTeste3{
16     public static double subtrair(double num1, double num2) {
17         double resultado = 0;
18         resultado = num1 - num2;
19         return resultado;
20     }
21     public static int subtrair(int num1, int num2) {
22         int resultado = 0;
23         resultado = num1 - num2;
24         return resultado;
25     }
26
27     public static int subtrair(int num1, int num2, int num3) {
28         int resultado = 0;
29         resultado = num1 - num2 - num3;
30         return resultado;
31     }
32
33 }
34
35 public class Principal {
36     public static void main(String[] args) {
37         int numero = 0;
38         System.out.println("Valor da soma: " + ClasseTeste1.somar(2, 3));
39         ClasseTeste2 classeTeste2 = new ClasseTeste2();
40         System.out.println("Valor da subtração: " + classeTeste2.subtrair(2, 3));
41         System.out.println("Valor da subtração 1 da classe 3: " + ClasseTeste3.subtrair(20.3, 3.2));
42         System.out.println("Valor da subtração 2 da classe 3: " + ClasseTeste3.subtrair(10, 3));
43         System.out.println("Valor da subtração 3 da classe 3: " + ClasseTeste3.subtrair(10, 4, 4));
44     }
}
```

EXERCÍCIO SOBRE MÉTODOS.



1. Crie um método para cada operação matemática (adição, subtração, multiplicação), em seguida faça uma sobrecarga de tipo e quantidade de atributos para cada método.
2. Faça três métodos de classes e três de objetos. Cada método de classe deve receber duas informações via console e imprimir a informação na tela. Cada método de objeto deve receber três números via Scanner e detectar se estes são pares.
3. Ao receber três números, identifique se o número 2 está entre o número 1 e número 3. Por exemplo: 1, 4 e 10, o 4 está entre 1 e 10.
4. Ao receber 4 números, informe se este é positivo, negativo, par ou impar.