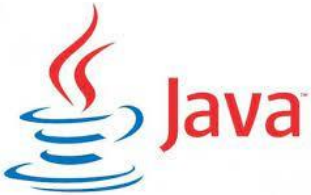
iwtraining



Formação Java
Aula 04

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

Sumário.

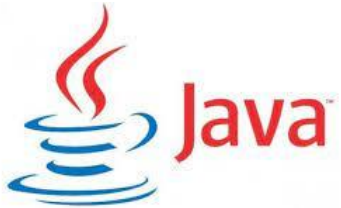
iwtraining

Método de classe - O que é?

Métodos de classe: método estático que é acessado e invocado diretamente pela classe. Não há necessidade de instanciar um objeto.

iwtraining

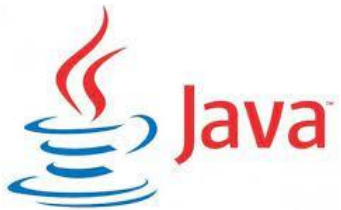
Método de classe - Exemplo




```java
class OperacaoMatematica{
    public static double adicao(double num1, double num2) {
        return num1 + num2;
    }
    public static double subtracao(double num1, double num2) {
        return num1 - num2;
    }
    public static double multiplicacao(double num1, double num2) {
        return num1 * num2;
    }
    public static double divisao(double num1, double num2) {
        return num1 / num2;
    }
}
public class Principal1 {
    public static void main(String[] args) {
        System.out.println("O valor da adição é: " + OperacaoMatematica.adicao(10,200));
        System.out.println("O valor da subtração é: " + OperacaoMatematica.subtracao(100,50));
        System.out.println("O valor da adição é: " + OperacaoMatematica.multiplicacao(10,40));
        System.out.println("O valor da adição é: " + OperacaoMatematica.divisao(150,30));
    }
}
```

iwtraining

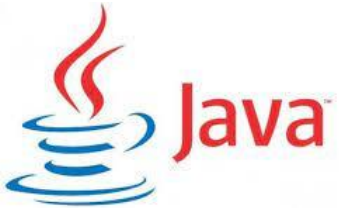
Método de objeto - O que é?



Métodos de objeto: método não estático que é acessado e invocado diretamente pelo o objeto da classe. Há necessidade de instanciar um objeto.

iwtraining

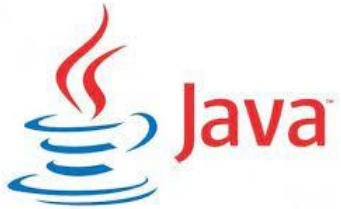
Método de objeto - exemplo.



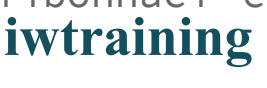

```java
class OperacaoMatematica{
    public  double adicao(double num1, double num2) {
        return num1 + num2;
    }
    public double subtracao(double num1, double num2) {
        return num1 - num2;
    }
    public double multiplicacao(double num1, double num2) {
        return num1 * num2;
    }
    public double divisao(double num1, double num2) {
        return num1 / num2;
    }
}
public class Principal1 {
    public static void main(String[] args) {
        OperacaoMatematica operacaoMatematica = new OperacaoMatematica();

        System.out.println("O valor da adição é: " + operacaoMatematica.adicao(10,200));
        System.out.println("O valor da subtração é: " + operacaoMatematica.subtracao(100,50));
        System.out.println("O valor da adição é: " + operacaoMatematica.multiplicacao(10,40));
        System.out.println("O valor da adição é: " + operacaoMatematica.divisao(150,30));
    }
}
```

iwtraining

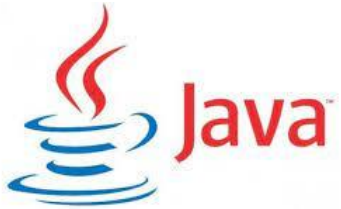
Método de classe e de objeto - exercício.



1. Implemente as 6 operações matemáticas (adição, subtração, multiplicação, divisão, potenciação e radiciação) por meio métodos de objetos e de classe.
   a. Observação 1: A mensagem imprimida no console não deve ser feita no método main.
   b. Observação 2: Faça sobrecargas de métodos, tanto no tipo como na quantidade de parâmetros.
   c. Observação 3: Pode-se utilizar bibliotecas para fazer as operações de potência e radiciação.
   d. Observação 4: Faça pelo menos duas comunicações entre métodos;
   e. Observação 5: Crie outra classe interna, faça um método para calcular o fatorial, fibonnaci e números pares. Use os métodos necessários das quatros operações para calcular o fibonnaci e fatorial.



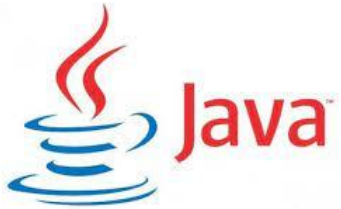
Método de classe e de objeto - exercício.



1. Simule as operações bancárias habitualmente realizadas no quotidiano. Use o estilo de método que mais esteja confortável.
   a. Observação 1: Faça um método para ver o saldo da conta vigente;
   b. Observação 2: Faça um método para realizar transferência entre contas;
   c. Observação 3: Faça um método para realizar depósitos bancários;
   d. Observação 4: Faça um método para realizar um pix. Neste método não se deve perguntar para realizar a operação;
   e. Observação 5: Faça um método para guardar dinheiro na poupança. Aplique um valor de rendimento de 0,02% diário sobre o valor inserido na poupança;
   f. Observação 6: Faça um método para resgatar dinheiro da poupança.

iwtraining

Listas e arrays - o que são?

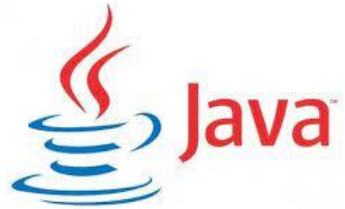


Lista: uma coleção indexada de objetos às vezes chamada de sequência, podendo alterar o seu tamanho depois de criada. Não se recomenda a utilização para trabalhar com índice.

Array: estruturas de dados que consistem em itens de dados do mesmo tipo relacionado, permanecendo com o mesmo tamanho depois de criados. Recomenda-se a utilização para trabalhar com índice.

iwtraining

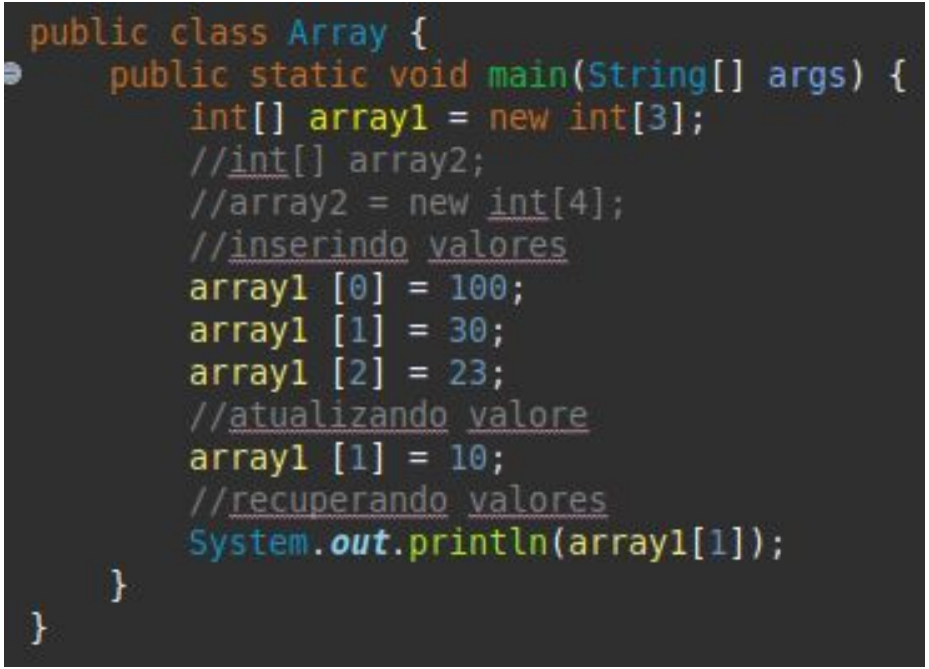
Listas - Exemplo




```java
import java.util.ArrayList;
public class Lista {
    public static void main(String[] args) {
        ArrayList<String> endereco = new ArrayList<>();
        //adicionando valores
        endereco.add("Rua do Bobo");
        endereco.add("Avenida 13 de maio");
        endereco.add("Avenida Pontes Vieira");
        //removendo valores
        endereco.remove(2);
        //recuperando valores
        endereco.get(0);
        //atualizando valores
        endereco.set(1, "Rua da quadra");
        System.out.println(endereco);
    }
}
```

iwtraining

Array - Exemplo



```java
public class Array {
    public static void main(String[] args) {
        int[] array1 = new int[3];
        //int[] array2;
        //array2 = new int[4];
        //inserindo valores
        array1 [0] = 100;
        array1 [1] = 30;
        array1 [2] = 23;
        //atualizando valore
        array1 [1] = 10;
        //recuperando valores
        System.out.println(array1[1]);
    }
}
```

iwtraining

Listas e arrays - exercícios,

1. Declarar três listas. Realizar as operações de inserção, atualização, remoção, listar por id e listar todos os itens da lista.
    a. Observação 1: A lista 1 deve ser do tipo String.
    b. Observação 2: A lista 2 deve ser do tipo Integer.
    c. Observação 3: A lista 3 deve ser do tipo objeto.
    d. Observação 4: Informar um valor a lista e verificar se este está na lista.
2. Declarar três vetores. Realizar as operações de inserção, atualização e listar os itens do vetores e item por id.
    a. Observação 1: A lista 1 deve ser do tipo String.
    b. Observação 2: A lista 2 deve ser do tipo Integer.
    c. Observação 3: A lista 3 deve ser do tipo objeto.

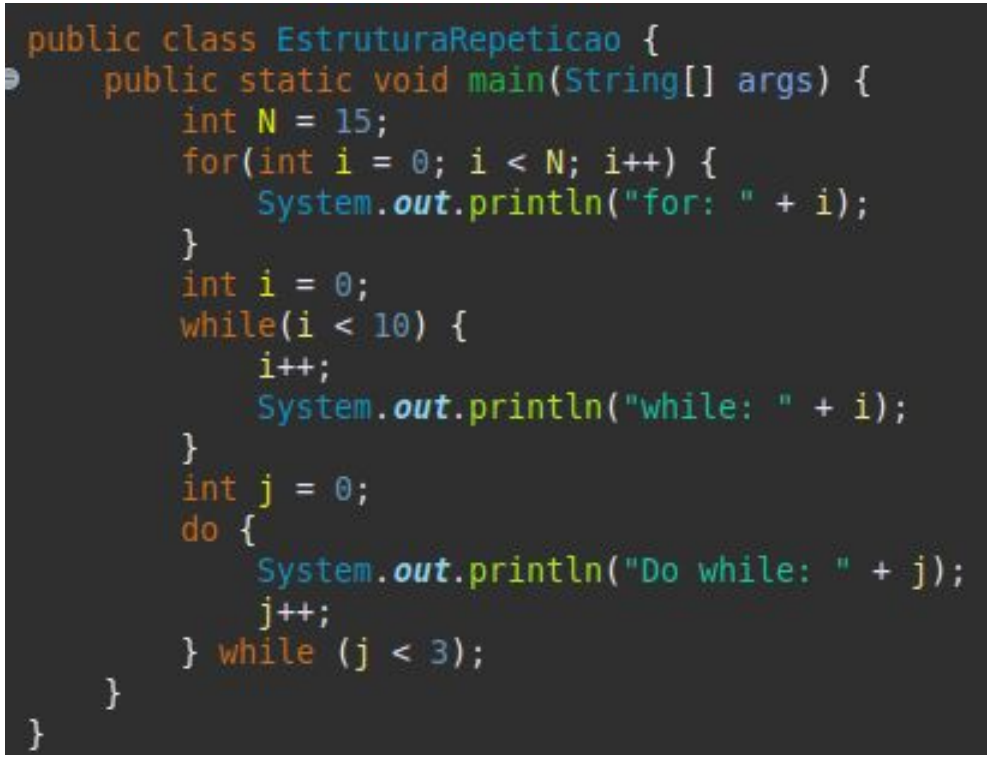
iwtraining

Estruturas de repetição - o que são?

É uma estrutura que permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador.

iwtraining

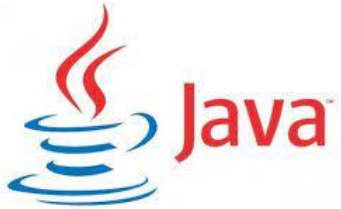
Estrutura. de repet. - for, while e do while



```java
public class EstruturaRepeticao {
    public static void main(String[] args) {
        int N = 15;
        for(int i = 0; i < N; i++) {
            System.out.println("for: " + i);
        }
        int i = 0;
        while(i < 10) {
            i++;
            System.out.println("while: " + i);
        }
        int j = 0;
        do {
            System.out.println("Do while: " + j);
            j++;
        } while (j < 3);
    }
}
```

iwtraining

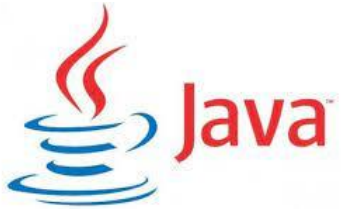
Estrutura de repetição - Exercícios - Pág 1.



1. Faça a soma dos N primeiros números;
2. Desenvolver um algoritmo que efetue a soma de todos os números ímpares que são múltiplos de três e que se encontram no conjunto dos números de 1 até 500.
3. Faça a soma dos N primeiros números ímpares;
4. Faça a soma dos N primeiros números pares;
5. Faça a tabuada de 1 a 9 das quatro operações;
6. Determine os números primos de 1 a 1000;
7. Faça a soma dos números primos entre 100 e 300;
8. Dado três números, determine se este é um número triangular;
9. Dado 10 números, determine o número maior e menor;
10. Imprima os valores das diagonais primárias e secundárias;
11. Dado 100 números, some aqueles que estão entre 50 e

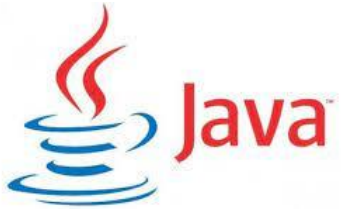
iwtraining

Estrutura de repetição - Exercícios - Pag 2.



1. Desenvolver um algoritmo que leia um número não determinado de valores e calcule e escreva a média aritmética dos valores lidos, a quantidade de valores positivos, a quantidade de valores negativos e o percentual de valores negativos e positivos;
2. Escrever um algoritmo que leia uma quantidade desconhecida de números e conte quantos deles estão nos seguintes intervalos: [0-25], [26-50], [51-75] e [76-100]. A entrada de dados deve terminar quando for lido um número negativo;
3. Faça um algoritmo estruturado que leia uma quantidade não determinada de números positivos. Calcule a quantidade de números pares e ímpares, a média de valores pares e a média geral dos números lidos. O número que encerrará a leitura será zero;

Estrutura de repetição - Exercícios - Pag 3.



1. Escrever um algoritmo que gera e escreve os números ímpares entre 100 e 200;
2. Escreva um algoritmo que leia um valor inicial A e uma razão R e imprima uma seqüência em P.A. contendo 10 valores;
3. Escreva um algoritmo que leia um valor inicial A e uma razão R e imprima uma seqüência em P.G. contendo 10 valores;
4. Escreva um algoritmo que leia um valor inicial A e imprima a seqüência de valores do cálculo de A! e o seu resultado. Ex: 5! = 5 X 4 X 3 X 2 X 1 = 120;
5. Elabore um algoritmo para gerar as sequências de fibonacci.

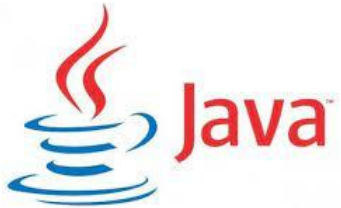
iwtraining

Recursividade

Recursão é um método de programação no qual uma função chama a si mesma. A recursão é utilizada quando queremos resolver um **subproblema** do mesmo tipo menor.

iwtraining

Recursividade - exemplo




```java
class OperacaoRecursiva{
    public static int somarPares(int x) {
        if(x == 0) return 0;
        if(x % 2 == 0) return x + somarPares(x - 1);
        return somarPares(x - 1);
      }
}
public class Recursividade {
    public static void main(String[] args) {
        System.out.println(OperacaoRecursiva.somarPares(5));
    }
}
```

iwtraining

Recursividade - exercício

1. Usando recursividade, faça a soma dos números pares;
2. Usando recursividade, faça a soma dos n números;
3. Usando recursividade, imprima a sequência numérica dos n números;
4. Usando recursividade, determine o fatorial de um número;
5. Usando recursividade, determine o número de fibonacci;

iwtraining