

.....

Aulas 3-4

Revisão, Tipos de Dados, Operadores e Comandos de Saída, Variáveis/Comandos de Entrada

Prof. Fernando Zuher



Universidade
Anhembi Morumbi

LAUREATE INTERNATIONAL UNIVERSITIES



Revisando...

Revisando...

- O que é um algoritmo?
 - Um algoritmo é uma sequência bem definida de passos (ou ações) que devem ser levados a cabo (executados) para resolver um determinado problema.

Revisando...

- Um algoritmo:
 1. Possui um ponto de entrada (passo inicial) e um ponto de saída (passo final) bem definidos.
 2. É composto de passos individuais.
 3. Cada passo está bem definido, pode ser executado, e o seu resultado é previsível. Isto é, repetidas execuções de um determinado passo produzirão sempre o mesmo resultado.

Revisando...

- Um algoritmo:
 4. Existe uma ordem (sequência) adequada para a execução dos passos. Essa sequência é fundamental para a obtenção de um resultado final correcto. Em geral, depois de executado um determinado passo, a execução prossegue no passo seguinte.
 5. O número de passos a executar é finito.
 6. Quando executado com um conjunto de dados válido um algoritmo termina garantidamente, produzindo o resultado esperado.

Revisando...

- Dentre as formas de representação de algoritmos mais conhecidas podemos citar:
 - Descrição Narrativa;
 - Fluxograma;
 - Pseudocódigo.

Revisando...

- Descrição Narrativa
 - Nesta forma de representação os algoritmos são expressos diretamente em linguagem natural. Exemplo de uma receita de bolo:
 - Misture os ingredientes;
 - Unte a forma com manteiga;
 - Despeje a mistura na forma;
 - Se houver coco ralado
 - então despeje sobre a mistura;
 - ...

Revisando...

- Fluxograma

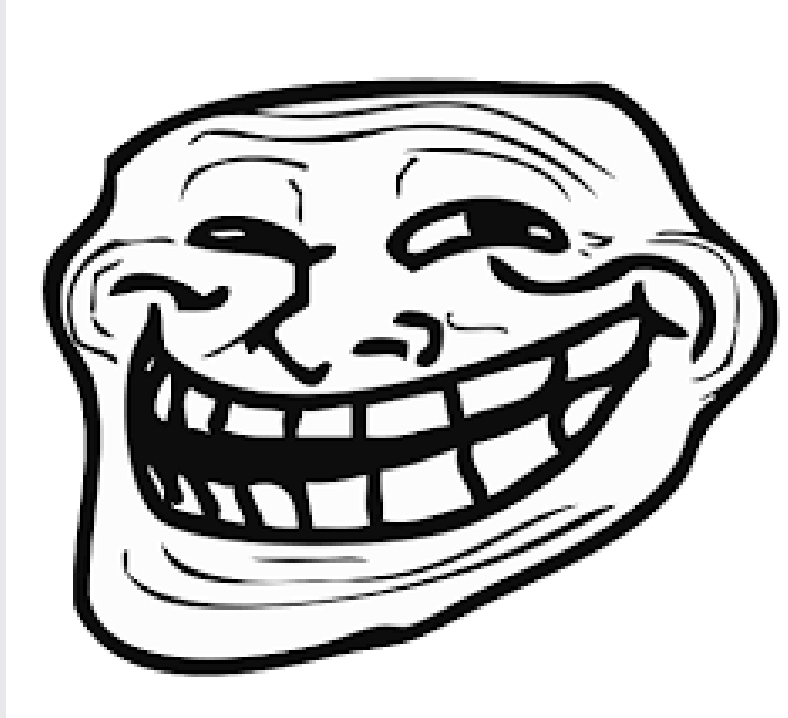
- É uma representação gráfica de algoritmos onde formas geométricas diferentes implicam ações (instruções, comandos) distintos.
- Tal propriedade facilita o entendimento das ideias contidas nos algoritmos e justifica sua popularidade.
- Esta forma é aproximadamente intermediária à descrição narrativa e ao pseudocódigo (subitem seguinte), pois é menos imprecisa que a primeira e, no entanto, não se preocupa com detalhes de implementação do programa.

Revisando...

- Fluxograma
 - Nota-se que os fluxogramas convencionais preocupam-se com detalhes de nível físico da implementação do algoritmo.
 - Por exemplo, figuras geométricas diferentes são adotadas para representar operações de saída de dados realizadas em dispositivos distintos, como uma fita magnética ou um monitor de vídeo.

Revisando...

- Pseudocódigo



Revisando...

- Pseudocódigo
 - Enquanto as linguagens de programação permitem expressar um algoritmo de um modo formal, o pseudocódigo permite fazê-lo de um modo mais informal, sem a preocupação de obedecer a um léxico e a uma gramática rígida.
 - No entanto, apesar desta liberdade, dois programadores distintos devem ser capazes de estar de acordo quanto a um dado algoritmo escrito em pseudocódigo e ser capazes de escrever dois programas funcionalmente equivalentes.

Revisando...

- Pseudocódigo
 - O pseudocódigo pode ser escrito em qualquer linguagem natural (humana) na forma de uma lista ordenada (ou numerada) de passos.
 - O algoritmo descrito na forma de um fluxograma pode ser escrito em pseudocódigo como:
 1. Início
 2. Passo 1
 3. Passo 2
 4. Passo 3
 5. Fim

Constantes e Tipos de Dados

- Constantes são valores que não se alteram em um algoritmo (valores fixos)
- Os dados são os valores que serão utilizados para a resolução de um problema
 - Fornecidos pelo usuário do programa
 - Originados de processamentos (cálculos)
 - Outros programas, etc
- **Classificados em:**
 - Literal
 - Numérico (inteiro / real)
 - Lógico

$$\frac{(n1 + n2)}{2}$$

Constantes e Tipos de Dados

Primitivos		Específicos para a Linguagem Java	
Tipos de Dados	Definição	Tipos de Dados	Capacidade
Literal – também conhecido como texto ou caractere	Poderá receber letras, números e símbolos	char / String	16 bits (2 bytes)
Inteiro	Poderá receber números inteiros positivos ou negativos	int	32 bits (4 bytes) - 2.147.483.648 a 2.147.483.647
Real – também conhecido como ponto flutuante	Poderá receber números reais, isto é, com casas decimais, positivos ou negativos	float / double	32 bits (de $-3,4E-38$ até $+3,4E+38$) 64 bits (8 bytes) (de $-1,7E-308$ até $+1,7E+308$)
Lógico – também conhecido como booleano	Poderá receber verdadeiro (1) ou falso (0)	boolean	8 bits (true ou false)

Exercício 1

- Identificar o tipo de dado/constante:
 - false
 - 21
 - “O resultado é: ”
 - 3.1415
 - “true”
 - 6,78
 - ‘h’

Operadores

- Utilizados para representar expressões de cálculos, comparação, condição e atribuição
- Basicamente três tipos de operadores:
 - Operadores Aritméticos
 - Operadores Relacionais
 - Operadores Lógicos

Operadores Aritméticos

Operação	Operador	Expressão	Resultado
Adição	+	6 + 4	10
Subtração	−	7 − 9	-2
Multiplicação	*	12 * 3	36
Divisão	/	44 / 2	22
Módulo (Resto da Divisão)	%	10 % 3	1

Operadores Aritméticos

$$\begin{array}{r} 10 \overline{) 3} \\ 3 \end{array}$$

Operador **/** (divisão)

div = 10 / 3

$$\begin{array}{r} 10 \overline{) 3} \\ - 9 \\ \hline 1 \end{array}$$

Operador **%** (módulo)

resto = 10 % 3

Exemplo

- Qual o resultado da avaliação da expressão

$$10 - 4 * 2 + 1$$


- 1, 13, 18, 3?
- Depende da precedência dos operadores!

Precedência de Operadores

Menor



+ -

/ *

()

Maior

Exemplos

$$6 * 4 + 5 - 3$$

$$3 + (2 - 1) * 5$$

Operadores Relacionais

- Estabelecem uma relação de comparação entre valores ou expressões
- Resultam sempre em um valor lógico
 - VERDADEIRO/FALSO (**true/false**)

Operadores Relacionais

Operador	Representação	Exemplo
Maior que	$>$	$a > b$: Se o valor de a for maior do que o valor de b, retornará true. Senão, retornará false
Maior ou igual a	$>=$	$a >= b$: Se o valor de a for maior ou igual ao valor de b, retornará true. Senão, retornará false
Menor que	$<$	$a < b$: Se o valor de a for menor do que o valor de b, retornará true. Senão, retornará false
Menor ou igual a	$<=$	$a <= b$: Se o valor de a for menor ou igual ao valor de b, retornará true. Senão, retornará false
Igual a	$==$	$a == b$: Se o valor de a for igual ao valor de b, retornará true. Senão, retornará false
Diferente de	$!=$	$a != b$: Se o valor de a for diferente do valor de b, retornará true. Senão, retornará false

Operadores Lógicos

- Utilizados para concatenar ou associar expressões que estabelecem uma relação de comparação entre valores
- O resultado dessas expressões é sempre um valor lógico (**true/false**)
- **Tabela Verdade**
 - São tabelas que representam todas as possíveis combinações das variáveis de entrada de uma função, e os seus respectivos valores de saída.

Operadores Lógicos

- **Operador E (&&)**

- Somente resulta em VERDADEIRO (true), se TODAS as expressões condicionais forem verdadeiras

- **Operador OU (||)**

- Se apenas UMA expressão condicional for verdadeira, o resultado é VERDADEIRO (true)

- **Operador NÃO (!)**

- Se a expressão condicional for falsa, o resultado é verdadeiro. Caso contrário é falso

Operadores Lógicos

- Tabela-verdade para os operadores && e ||

A	B	A && B	A B
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

&& – somente resulta em VERDADEIRO quando todas as sentenças avaliadas forem verdadeiras

|| – somente resulta em FALSO quando todas as sentenças avaliadas forem falsas

Operadores Lógicos

- O operador ! faz a negação de uma sentença:

A	! A
V	F
F	V

- Precedência dos operadores: !, && ou ||

Exercício 2

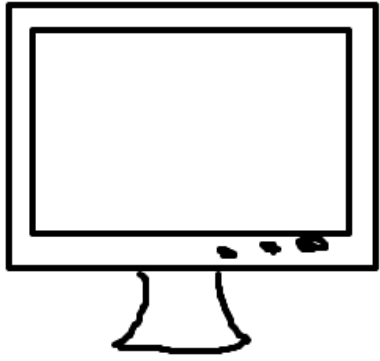
- Suponha $w = \text{false}$, $x = \text{true}$ e $z = \text{true}$. Resolva as expressões:

– $x \parallel z \&\& w$

– $z \&\& x \&\& x$

– $z \parallel z \parallel w \&\& x$

– $x \&\& z \&\& w \parallel !w$



Estrutura de um programa/ Comando de Saída

Estrutura de um programa Java

- Como Java é uma linguagem orientada a objeto, todo o código de um programa deve estar definido dentro de **uma classe**
- Especificamente deve estar definido dentro de um método especial da classe chamado **main()**
- O Método **main()** é o ponto de partida na execução de um programa em java

Estrutura de um programa em Java

Início →

```
public class PrimeiroPrograma
{
    public static void main(String[] args)
    {
        // Aqui serão digitados os passos do algoritmo
        /* Os comentários podem
           ser em blocos
        */
        instrução1;
        instrução2;
        instrução3;
        instrução4;
    }
}
```

Fim →

TAB TAB

servem para documentar o algoritmo e são ignorados pelo compilador

O resultado do processamento de um algoritmo pode ser armazenado em uma **variável** e/ou pode ser exibido diretamente em um **dispositivo de saída**

- Monitor é o dispositivo de saída padrão



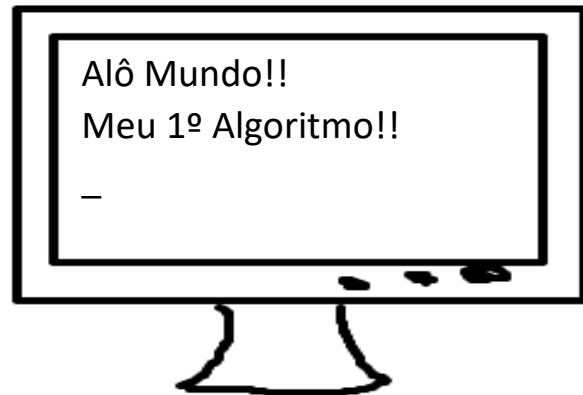
Comando de Saída

Comando de Saída

- Utilizado para escrever dados e resultados na saída padrão (monitor)

System.out.println(" ");

```
1 public class AloMundo {  
2     public static void main(String[] args) {  
3         System.out.println("Alô Mundo!!");  
4         System.out.println("Meu 1º Algoritmo!!");  
5     }  
6 }
```



Comando de Saída

- Para escrever mais de um tipo de dado vamos usar o operador de concatenação (+)

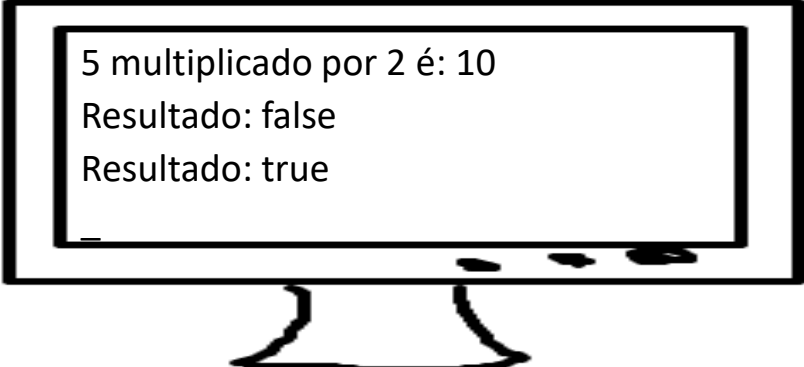
```
1 public class Altura {  
2     public static void main(String[] args) {  
3         System.out.println("Minha altura é: " + 1.75 + " metros");  
4     }  
5 }
```



Comando de Saída

- Também é possível realizar processamentos e comparações no comando de saída

```
1  public class Processamento{
2      public static void main(String[] args){
3          System.out.println("5 multiplicado por 2 é:" + (5*2) );
4          System.out.println("Resultado:" + (1023 == 1203) );
5          System.out.println("Resultado:" + ((1 < 4) && !(3!=3)) );
6      }
7  }
```



5 multiplicado por 2 é: 10
Resultado: false
Resultado: true

Vamos praticar?

Exercícios

3. Faça um algoritmo para imprimir em cada linha: o seu nome completo, o seu RA e o seu curso
4. Faça um algoritmo para imprimir a sua idade, o seu peso e o seu sexo. Indicar o sexo como sendo: f – feminino ou m – masculino.
5. Faça um algoritmo para apresentar o resultado lógico das seguintes expressões:
 - $((120 - 30) = (3 \times 30))$
 - $(\text{não} ((20 \bmod 4) = 1) \text{ ou } (9 \neq 9))$
 - $((5 \bmod 2 / 2) > 3)$
 - $(a = A)$

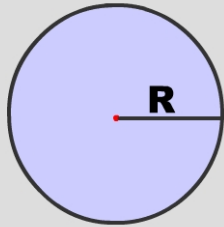
Exercícios

6. Faça um algoritmo para apresentar o cálculo da média aritmética das seguintes notas 8.0, 7.5, 4.5 e 9.
7. Construir um algoritmo apresentar o cálculo da área de um quadrado de 350 m de lado.

Exercícios

8. Construir um algoritmo para imprimir o cálculo da área de um círculo com raio de 5 cm. ($\pi=3,14159$)

ÁREA DO CÍRCULO



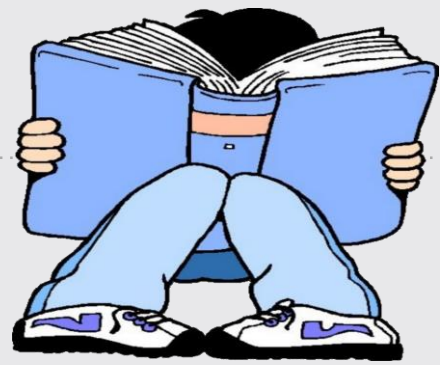
$$A = \pi \cdot R^2$$

Exercícios

9. Faça um algoritmo que calcule o valor de cada expressão a seguir:

- $(20 - 15)/2$
- $2*(5/20) + 30/(15*2)$
- $35 / (6 + 2)$
- 23 modulo 4

Leitura para casa



- Ler o capítulo 1 do livro:
- **Fundamentos da Programação de Computadores**
– Ana Fernanda Gomes Ascencio
- <http://anhembi.bv3.digitalpages.com.br/users/publications/9788564574168/pages/1>

.....

Variáveis/Comando de Entrada

.....

Prof. Fernando Zuher



Universidade
Anhembi Morumbi

LAUREATE INTERNATIONAL UNIVERSITIES



Variáveis

- Espaços de memória (recipientes) que são usados para armazenar os dados de um programa
- O valor armazenado na variável pode ser alterado!
- Porém, a cada instante somente um único valor é assumido

Regras para Nomear Variáveis

- É necessário nomear a variável (identificador)
- O nome da variável deve ser único e deve seguir algumas regras

1

Não utilizar espaços entre as letras ()

2

Não iniciar o nome da variável com número

3

Não utilizar palavras reservadas da linguagem

4

Não utilizar caracteres especiais (acentos, símbolos, ? / : @ ç #, etc)

5

Ser sucinto e utilizar nomes coerentes

Regras para Nomear Variáveis

- Fique atento!!!
 - Os nomes de variáveis abaixo são iguais???

idade

Idade

IdaDe

- Não, pois variáveis em Java são case-sensitive!
 - nomes com letras maiúsculas são diferente de nomes com letras minúsculas!!

Exercício 1

- Verifique se as variáveis abaixo possuem nomes corretos e justifique as alternativas falsas :

Lado.do.terreno

4media

class

endereço@e-mail&

prova 1

mediaAritmeticaConstrucao_Algoritmos

salario/hora

Declaração de Variáveis

- Para declarar uma variável é necessário indicar o tipo de dado que ela irá armazenar
- Uma variável do tipo int só poderá armazenar valores inteiros

tipo_de_dado *identificador1*;

int quantidade;

tipo_de_dado *identificador1, identificador2*;

int quantidade, numeroPessoas;



Jeliot – declaração de variáveis

Exemplos de Declaração de Variáveis

- `String nome , endereco;`
- `double salario;`
- `int numero;`
- `char sexo;`
- `boolean resultado;`

Exercício 2

- Faça a declaração de variáveis para os seguintes dados:
 - Endereço de um imóvel
 - Resultado de uma divisão entre inteiros
 - Media aritmética de uma disciplina
 - Número de guichês de atendimento
 - Preço de um produto
 - Sexo de uma pessoa
 - Resultado da expressão ($3 < 9$)
 - Indicar uma escolha de S ou N para uma pergunta (Sim/Não)

Atribuição - operador (=)

- Indicar um valor a ser armazenado em uma variável
- A variável irá guardar esse valor até que ele seja modificado
- O tipo de dado DEVE ser compatível com o tipo da variável
- Portanto, se uma variável é do tipo String, somente podemos atribuir um valor do tipo String a ela

identificador1 = valor;

nome = "Mario de Andrade";

idade = 17;

Atribuição - Exemplo

//Declaração de variáveis

```
int A;  
double X, B;  
double valor;  
String nome;
```

//Atribuição de valores

```
A = 85;  
X = 8 + (13 / 5.0);  
B = 100.52;  
valor = 123.456789;  
nome = "Joao da Silva";
```



- Mostrar a declaração variáveis e a atribuição de valores no Jeliot
- Mostrar a modificação de valores

Exercício 3

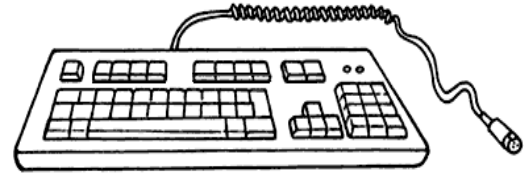
- Faça a declaração de variáveis para os dados a seguir.
- Faça uma atribuição de valor a cada variável declarada;

a-) idade

b-) valor da conta telefônica

c-) nome de empresa

d-) média final na disciplina



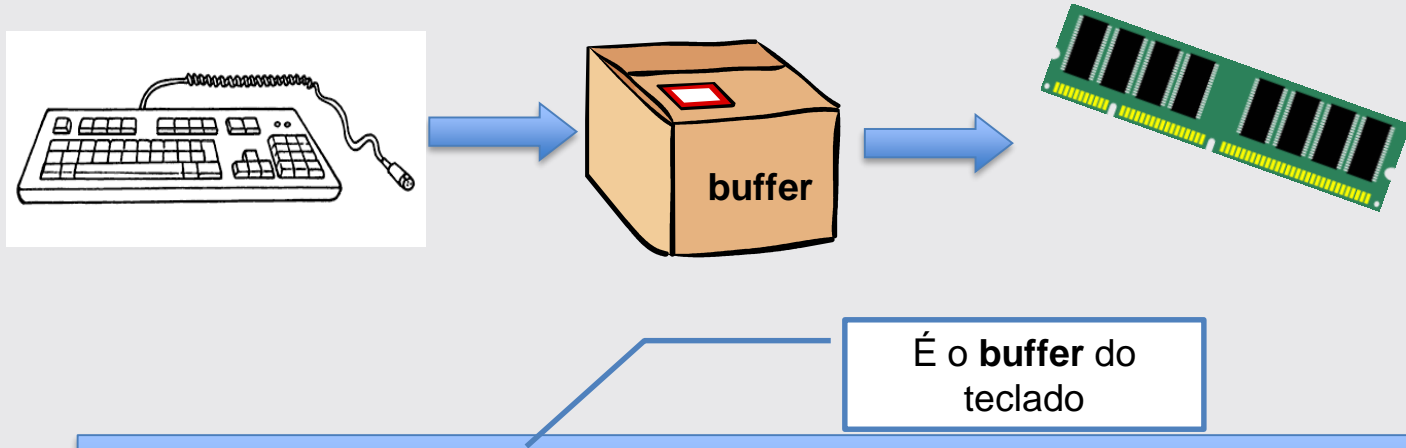
Comando de Entrada

Comando de entrada

- Permite que o usuário informe um dado a ser armazenado em uma variável no programa
- O teclado é o dispositivo de entrada padrão de um computador

Comando de entrada

- Mas, como ler os dados do teclado e guardar na memória do computador?
 - É necessário criar um buffer para guardar os dados digitados no teclado e depois transferí-los para a memória (variável)



```
Scanner entrada = new Scanner(System.in);
```


Comando de entrada

- Após criar o buffer do teclado, basta transferir esse dado para a variável
- A leitura é dada associando o tipo de entrada ao tipo da variável que receberá o dado

Tipo de Dado	Usar
String	<code>entrada.nextLine();</code>
int	<code>entrada.nextInt();</code>
double	<code>entrada.nextDouble();</code>
float	<code>entrada.nextFloat();</code>
char	<code>entrada.next().charAt(0);</code>
boolean	<code>entrada.nextBoolean();</code>

Comando de entrada

- Para usar a classe Scanner é necessário fazer o import de um pacote antes da definição da classe

```
import java.util.Scanner;
```

```
public class Exemplo {
```

```
}
```

Exemplo: Media.java

```
1 package media;
2 import java.util.Scanner;
3
4 public class Media {
5     public static void main(String[] args) {
6         Scanner entrada = new Scanner(System.in);
7
8         String aluno;
9         double nota1, nota2, media;
10
11         System.out.println("Digite seu nome: ");
12         aluno = entrada.nextLine();
13
14         System.out.println("Digite duas notas:");
15         nota1 = entrada.nextDouble();
16         nota2 = entrada.nextDouble();
17
18         media = (nota1 + nota2) / 2;
19
20         System.out.println(aluno + " sua média é: " + media);
21     }
22 }
```

Saída - Media (run)

```
run:
Digite seu nome:
Mariana
Digite duas notas:
8,5
5
Mariana sua média é: 6.75
CONSTRUÍDO COM SUCESSO (tempo total: 33 segundos)
```

Vamos praticar?

Exercícios

4. Faça um algoritmo que leia o nome, a idade e o sexo de uma pessoa. Imprimir todas as informações no final.
5. Faça um algoritmo que leia quatro números inteiros, calcule e mostre a soma desses números.
6. Faça um algoritmo que leia três notas de um aluno, calcule e mostre a média aritmética do aluno.
7. Faça um algoritmo que leia o salário de um funcionário. Sabendo que o salário do funcionário teve um aumento de 25%, calcular e mostrar o novo salário.

Exercícios

8. Construir um algoritmo que calcule a área de um triângulo dado a base e a altura. ($at = (altura * base) / 2$).
9. Construir um algoritmo para ler dois números inteiros e imprimir o seu produto.
10. Construir um algoritmo que leia o ano de nascimento de uma pessoa e o ano atual, calcule e mostre:
 - A idade dessa pessoa;
 - Quantos anos essa pessoa terá em 2020.

Exercícios

11. Sabe-se que um quilowatt de energia custa $1/500$ avos do salário mínimo. Faça um algoritmo que receba o valor do salário mínimo e a quantidade de quilowatts consumida por uma residência. Calcule e mostre:
- O valor, em reais, de cada quilowatt;
 - O valor, em reais, a ser pago por essa residência
 - O valor, em reais, a ser pago com desconto de 15%.

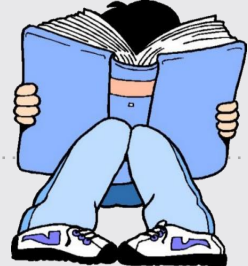
Exercícios

12. Faça um algoritmo que receba um número, calcule e mostre:

- O número elevado ao quadrado;
- O número elevado ao cubo;
- A raiz quadrada do número digitado;
- O número elevado a potência 10;

Método / Constante	Funcionalidade	Como usar
<code>Math.pow(double base, double exp);</code>	Calcula uma potência. Base elevada ao Expoente	<code>Math.pow(2 , 10);</code>
<code>Math.sqrt(double v);</code>	Calcula a raiz quadrada de um valor v	<code>Math.sqrt(25);</code>
<code>Math.PI</code>	Retorna a constante PI (3.1415)	<code>Math.PI</code>

Leitura de Livro !!



- Leitura do capítulo 3 (páginas 19 a 35).
- Livro: Fundamentos da Programação de Computadores.
 - Exercícios para resolver em casa (páginas 36 a 52).
 - Ler o capítulo do livro, fazer anotações sobre as dúvidas e levar as anotações para a próxima aula