**LOGICA DE PROGRAMAÇÃO EM JAVA**

**Variáveis, constantes e atribuição.**

Variável: é uma estrutura que permite armazenar dados na memória durante a execução do programa, para o procedimento de informações. Java é uma linguagem **fortemente tipada**, ou seja, cada variável obrigatoriamente deve ter um tipo declarado antes que possa ser utilizada.

Exemplo:  
int contador = 0;  
boolean flag = true; nome↧  
String name = “John”; tipo -> int[] array = new int[5]; <- atribuição

**Principais Tipos em JAVA.**

boolean : Tipos de dado que representa dois estados.  
Verdadeiro (true) ou Falso (false)  
Ex: boolean flag = true

char : Tipos de dado que armazena um caractere.  
Ex: char vogal = ‘a’;

int : Tipos de dado que armazena um valor inteiro  
Ex: int contador = 0;

long : Semelhante ao inteiro, porém reserva mais memória de armazenamento.  
Ex: long serialId = 19273675489L; - O L no final é obrigatório

float: Tipos de dados que armazena um número de pronto flutuante.  
Ex: float pi = 3,14f;

double: Tipos de dados semelhante ao float, porém pode armazenar números maiores.   
Ex: double saldo = 7834534369363853634636;

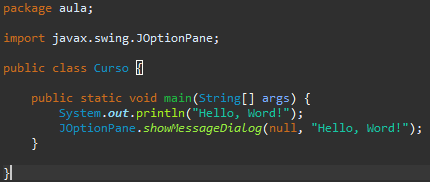
String: Classe utilizada para o armazenamento de dados em forma de texto.  
Ex: String txt = “João”;

**OBS: char utiliza aspas simples e String utiliza aspas duplas.**

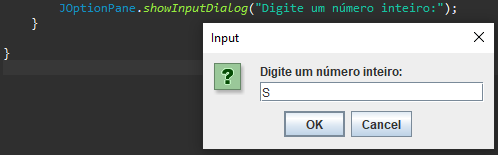
**Constantes**

Constantes: Valores que não irá se modificar durante o fluxo do programa  
Ex: final float pi = 3.14f;

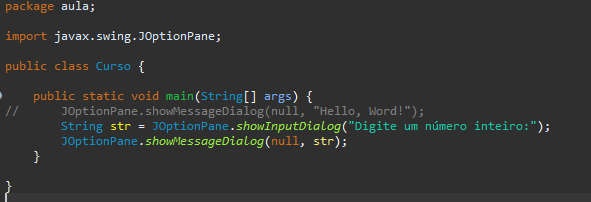
**Eclipse Comandos**

****

**-** O JOptionPane.*showMessageDialog* mostra somente a mensagem em uma janela flutuante.



- O JOptionPane.*showInputDialog Mostra uma janela para digitar.*



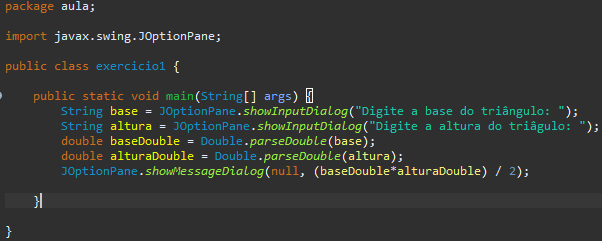
- JOptionPane.*showMessageDialog*(null, str); mostra o valor de uma variável e mostra um print.



- Transforma uma string em inteiro.



- Convertendo para Double de uma forma simplificada.



- Recebe base e altura e exprime a área do triângulo.

**Operadores aritméticos**

Soma: +  
Subtração: -  
Multiplicação: \*  
Divisão: /  
Módulo: %  
Incrementar: ++  
Decrementar: --

**Operadores relacionais**

Maior: >  
Menor: <  
Menor-igual: <=  
Maior-igual: >=  
Igual: ==  
Diferente: !=  
Negação: =!

**Operadores lógicos**

E: &&  
OU: ||  
ex: s == ‘f’ || s == ‘F’;

**Formatação**

DecimalFormat dc = new DecimalFormat("0. ##");

String sf = dc.format(*imc*(peso, altura));

JOptionPane.*showMessageDialog*(null, sf);

**Estrutura de decisão**

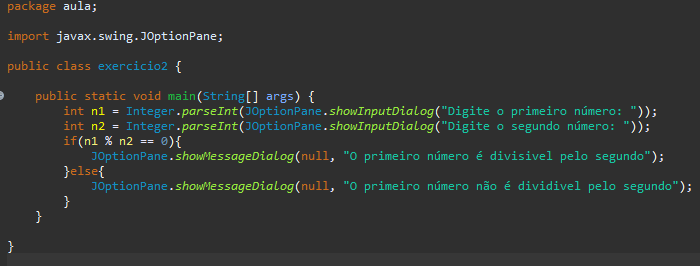
If(alguma coisa){  
 //faça algo

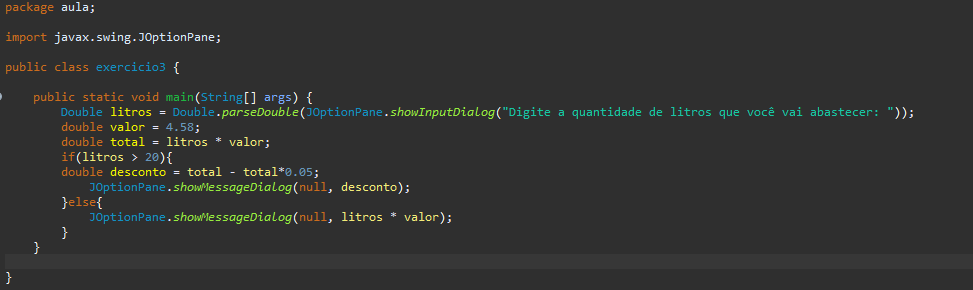
}else if(outra coisa){  
 //faça algo

}else{  
 //faça algo  
}

**OBS: em Java e em outras linguagens o escopo da estrutura é definido pelas chaves, ou sejam, tudo que está entre as chaves será executado caso a condição seja satisfeita.**

**Exercico:**

- O primeiro número é dividido pelo segundo



- Saber quantos litros para abastecer

**Estrutura de decisão**

Switch(sexo){  
 case’m’:  
 //faça algo  
 break;  
 case’f’:

//faça algo

break;  
 default:  
 break;

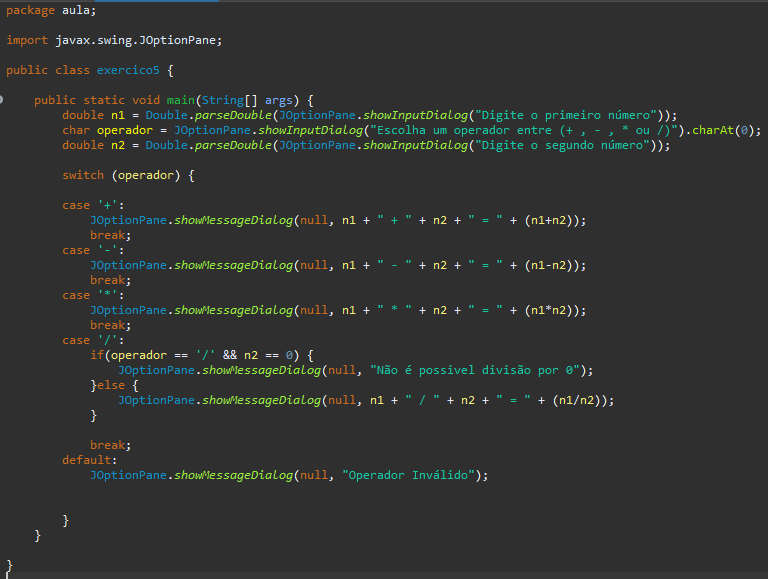
}

**OBS: Normalmente utilizado quando deseja-se verificar uma variável que pode assumir valores bem definidos.**

**Exercicios**

****

**-** O programa pede para digitar seu sexo e exprime ele.  
.toLowerCase() – Deixa a letra em maiúscula  
.charAt() – Pega uma letra da palavra que está em determinada posição de acordo com o número que foi colocado.



- Pedir 2 números e pedir um operador.

**Estruturas de repetição**

While:

While(condição){  
 //faça algo

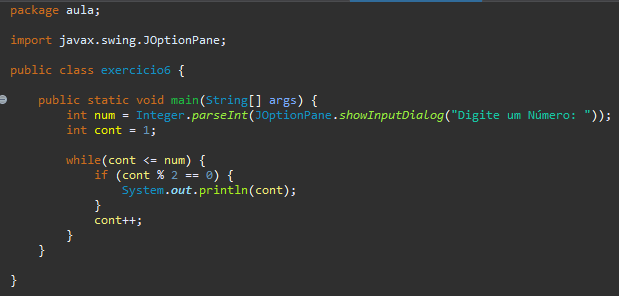
}

do While:

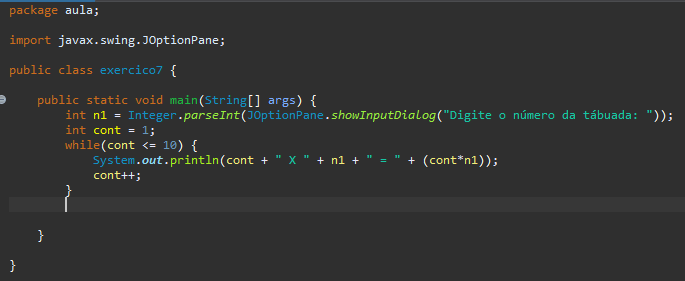
do{  
 //faça algo

}while(condição);

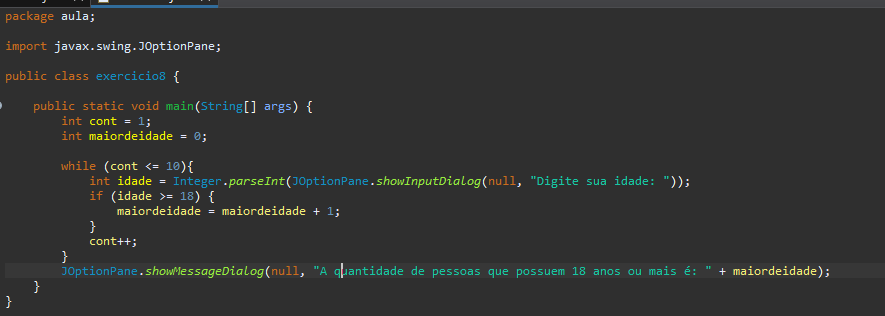
**Exercicios**

****

**-** Contador de números pares.



- Tabuada



- Quantidade de maior de idade.

**Estrutura de repetição**

for:

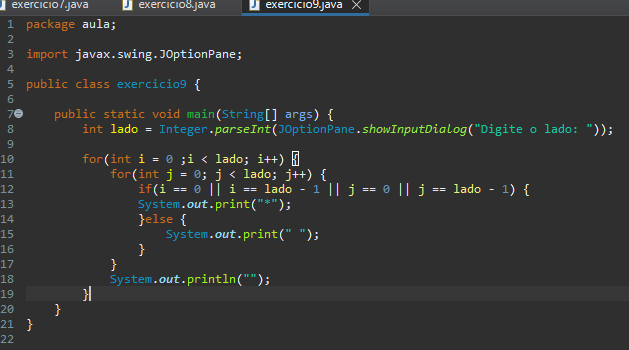
for(int i=0; i < 10; i++){  
 //faça algo  
}

foreach:

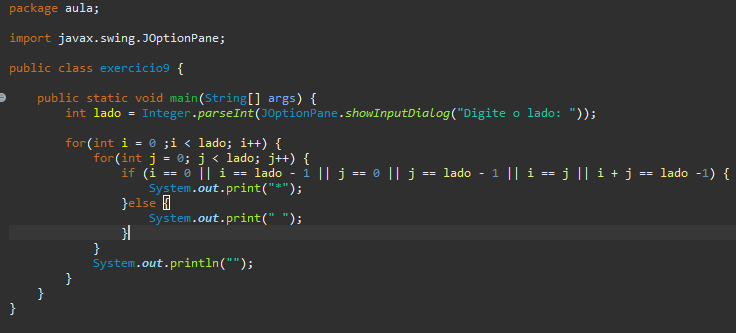
foreach(int item : item){

//faça algo  
}

**Exercicio:**

****

- Fazer uma matriz de \*



- Fazer uma matriz com um x no meio com \*

Funções

public static void main(String[] args){  
 int x = 5;  
 int y = 2;  
 int res = potencia(5, 2);  
}

public static int potencia(int numero, int potencia){  
 int res = 1;  
 for(int i = 0; i < potencia; i++){  
 res\*=numero;  
 }  
 return res;  
}

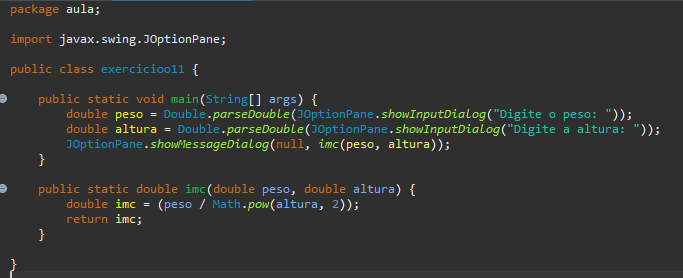
**Exercicios:**

****

- Pede um número e retorna quantos números são pares na posição.



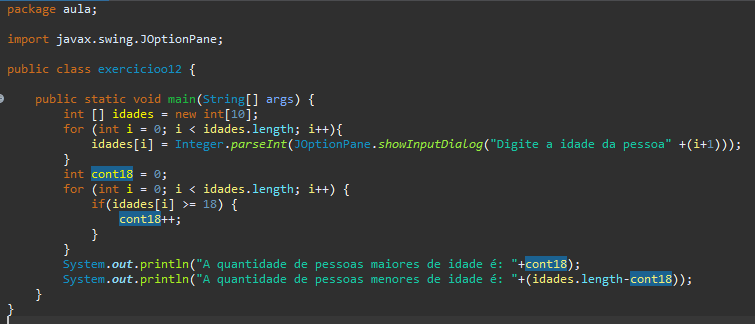
- Função que exprime quantos número pares são em um parâmetro.



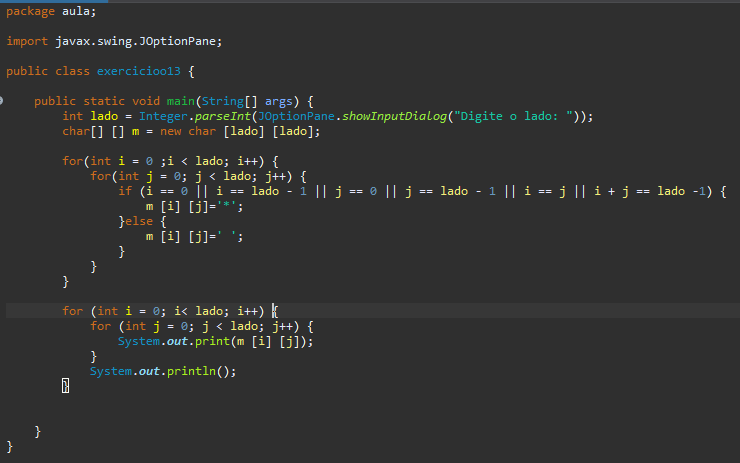
- Descobrir o Imc com função.



- Como formatar um double.



- Saber quantos é maior e menor de idade.



- Fazer os lados de uma forma diferente.