

Revisão de Lógica(Java)

Comandos básicos:

```
JOptionPane.showMessageDialog(null,"Olá Mundo!");
```

Mostra uma mensagem ao usuário, necessita do "null" antes de mensagem que será exibida, caso queira mostrar o valor de uma variável:

```
int A=10;
```

```
JOptionPane.showMessageDialog(null,"A variável A vale: "+A);
```

A mensagem será: A variável A vale 10

Mas eu quiser mostrar o valor de uma variável entre o texto?

```
int A=10;
```

```
JOptionPane.showMessageDialog(null,"O numero "+A+" está armazenado na  
variavel A!");
```

Resultado: O numero 10 está armazenado na variável A!

```
int x=JOptionPane.showInputDialog("Digite um valor para X!");
```

Recebe o valor de uma variável, ou seja, o que o usuário digitar, definirá o valor da variável. Só que este comando está errado, por que? Falta o comando Integer.valueOf, porque sem ele, mesmo que o usuário digite um número, "X" vai armazenar um texto, assim, você não poderia realizar calculos com "X".

Forma correta:

```
int x=Integer.valueOf(JOptionPane.showInputDialog("Digite um valor para X!"));
```

Caso "X" seja double:

```
double x=Double.valueOf(JOptionPane.showInputDialog("Digite um valor para  
X!"));
```

Mas se eu quiser receber um texto, ou seja, armazenar um texto em uma variável "String"? Simples, não precisa de conversão nem nada é só fazer assim:

```
String x=JOptionPane.showInputDialog("Digite um texto para X!");
```

Operadores:

Os operadores lógicos são aqueles que ajudam no processo de construção da lógica do programa, os que foram usados durante as aulas:

Aritméticos:

+ ← Realiza soma.

*** ← Realiza multiplicação**

/ ← Realiza divisão

% ← Resto da divisão(será explicado nos extras)

Verificação:

>← Verifica se uma variável é maior que a outra

>= ← Verifica se uma variável é maior igual a outra

< ← Verifica se uma variável é menor que a outra

<= ← Verifica se uma variável é menor igual a outra

Lógicos:

&& ← O “e” da tabela verdade, só é verdade quando as duas condições forem verdadeiras.

|| ← O “ou” da tabela verdade, só é falso quando tudo for falso. Se um for verdade tudo é verdade.

Comandos de verificação:

```
if(A>B){
```

```
JOptionPane.showMessageDialog(null,"A é maior que B!");
```

```
}
```

```
else if(A<B){
```

```
JOptionPane.showMessageDialog(null,"B é maior que A!");
```

```
}
```

```
else{
```

```
JOptionPane.showMessageDialog(null,"A e B são iguais!");
```

```
}
```

Traduzindo, **SE** A for maior que B, será exibida a mensagem “A é maior que B!”, **MAS CASO** A for menor que B, outra mensagem será exibida “B é maior que A!”, mas **se nenhuma dessas afirmações forem verdadeiras**, significa que A é igual a B. Se eu tiver apenas uma condição, utilizo só o if, exemplo:

```
if(A==10){
```

```
JOptionPane.showMessageDialog(null,"A é igual a 10!");
```

```
}
```

Nesse caso apenas verifiquei **SE A é igual a 10**, mas se eu quisesse verificar também se A for maior que 10?

```
if(A==10){
```

```
JOptionPane.showMessageDialog(null,"A é igual a 10!");
```

```

}
else if(A>10){
JOptionPane.showMessageDialog(null,"A é maior que 10!");
}

```

Se a for 10, o programa exibe a mensagem de que A é igual a 10, mas caso a variável seja **MAIOR** que 10, ele ignora o if e passa para o else if, e verifica se A é maior que 10, caso seja verdade, exibe a mensagem. Mas se A não for 10, e também for menor que esse número, posso fazer com o que o programa exiba a mensagem de que A nem é 10 e é menor que esse número? Sim! Basta utilizar o else.

```

if(A==10){
JOptionPane.showMessageDialog(null,"A é igual a 10!");
}
else if(A>10){
JOptionPane.showMessageDialog(null,"A é maior que 10!");
}
else{
JOptionPane.showMessageDialog(null,"A não é 10 e é menor que esse número");
}

```

O **else** é utilizado quando nenhuma das afirmações são verdadeiras, é como se fosse a última opção, o diferencial do **else** é que ele não precisa de condição, apenas dos comandos dentro das chaves.

Comandos de repetição:

São comandos que repetem outros comandos enquanto uma determinada condição for cumprida. Por exemplo "Correrei até me cansar", você irá correr, correr e correr, e só vai parar quando estiver cansado, "estar cansado" é a condição, "correr" é a ação que será realizada. Existem 3 comandos de repetição: Do, Do While e For.

While:

```

while(A<10){
int A=Integer.valueOf(JOptionPane.showInputDialog("Digite um valor para A"));
}

```

Enquanto o usuário digitar um número menor que 10, o programa vai repetir a mensagem para digitar um valor, caso o valor digitado seja maior que 10, ele para de repetir. O while primeiro verifica, depois realiza os comandos.

Do while:

```
int n=0;
```

```
do{
```

```
JOptionPane.showMessageDialog(null,“Está tudo errado”);
```

```
n++;
```

```
}
```

```
while(n<5);
```

Traduzindo, faça ou exiba a mensagem “Está tudo errado” e também incremente a variável n, enquanto n for menor que 5, ou seja, essa mensagem será exibida 6 vezes, pois na primeira vez que a mensagem é mostrada, o programa não verifica, pois no do while, primeiro é feito depois é verificado.

For:

```
for(int contador=0;contador<5;contador++){
```

```
JOptionPane.showMessageDialog(null,“Está tudo errado”);
```

```
}
```

O for é como se fosse um while compacto, primeiro você define o contador (int contador=0;) que é o ponto inicial do for, depois você define até quando será repetido os comandos no caso do exemplo acima, enquanto contador for menor que 5(contador<5), e sempre que os comandos são feitos, essa variável contador, incrementa, fazendo assim com que o programa não fique em um loop infinito. Em outras palavras, o programa conta de 0 até 5, e quando chegar no 5, ele para, pois esse número já não é menor que 5(que é a condição dada).

Vetor:

```
int v[ ]= new int[10];  
for(int contador=0;contador<9;contador++){  
v[contador]=Integer.valueOf(JOptionPane.showInputDialog("Digite um  
valor para o vetor")):  
}  
for(int contador=0;contador<9;contador++){  
JOptionPane.showMessageDialog(null,"Valores do vetor:  
"+v[contador]);  
}
```

Vetores são variáveis que podem armazenar uma quantidade de valores determinado pelo criador do algoritmo, para se criar um vetor é simples, basta definir o tipo e nome da variável, e abrir e fechar os colchetes, assim: `int v[]`. `V` é o nome do vetor, mas para definir o tamanho desse vetor, como faço? `Int v[]= new int[10];`

Eu só igualo e coloco "new int(que no caso é o tipo da variável que criei) e [10](Que é o tamanho do meu vetor).

O que mais confunde quando se fala em vetor, são os índices, quando você cria um vetor, você define quantos valores ele vai armazenar, não seu índice máximo, por exemplo, se eu crio um vetor de tamanho 5, ele vai caber 5 valores, certo, porém, o índice máximo desse vetor é 4, veja.

```
Int v[ ]= new int[5];  
v[0]=1;  
v[1]=2;  
v[2]=3;  
v[3]=4;  
v[4]=5;
```

Viu só? Os índices do vetor sempre começam de zero, e como disse anteriormente, é um vetor de tamanho 5, e couberam esse 5 valores.

Para preencher o vetor, eu crio um for. Assim:

```
int v[ ]= new int[10];  
for(int contador=0;contador<9;contador++){  
v[contador]=Integer.valueOf(JOptionPane.showInputDialog("Digite um  
valor para o vetor"));
```

Por que dentro do for tem "**v[contador]**"? Porque esse contador vai ser substituído pelo valor que contador representar no momento, por exemplo, "**contador**" começa em 0, então, no momento **v[contador]** será na verdade **v[0]**, então significa que o valor que o usuário for digitar vai ser armazenado no índice 0, e assim por diante, já que o contador vai incrementando de 1 em 1.

Agora para exibir os números do vetor:

```
for(int contador=0;contador<9;contador++){  
JOptionPane.showMessageDialog(null,"Valores do vetor:  
"+v[contador]);
```

Segue a mesma lógica que expliquei anteriormente, só o que muda é que vou usar o **MessageDialog** para exibir os valores.

EXTRAS

Verificando se um número é par:

```
if(variavel%2==0){  
JOptionPane.showMessageDialog(null,"A variável é par");  
}
```

Mostrando os valores do vetor de forma inversa:

```
for(int contador=9;contador>0;contador--){  
JOptionPane.showMessageDialog(null,"Valores do vetor:  
"+v[contador]);  
}
```

Recebendo apenas valores entre 0 e 10:

```
if(n>=0 && n<=10){
```

```
JOptionPane.showMessageDialog(null,"Nota válida");
```

```
}
```