

FIA/P GRADUAÇÃO

# DOMAIN DRIVEN DESIGN

Prof. Me. Thiago T. I. Yamamoto

#07 - CONVERSÕES E TOMADA DE DECISÕES



# TRAJETÓRIA

---



- ✓ Orientação a Objetos
- ✓ Introdução ao Java
- ✓ IDE e Tipos de Dados
- ✓ Classes, atributos e métodos
- ✓ Encapsulamento
- ✓ Construtores
- ✓ Conversões e Tomada de decisões

# #07 - AGENDA

---



- Conversões
  - Parse
  - String
  - Implícitas/Explícitas
  - Casting
- Tomada de decisão
  - Operadores relacionais
  - Operadores lógicos
  - If/Else
  - Switch



# CONVERSÕES

- É possível **transformar** um valor contido numa **String** num valor de um **tipo primitivo**. Para isto utilize os métodos **parse** disponíveis em quase todas as **classes** que representam os tipos primitivos:

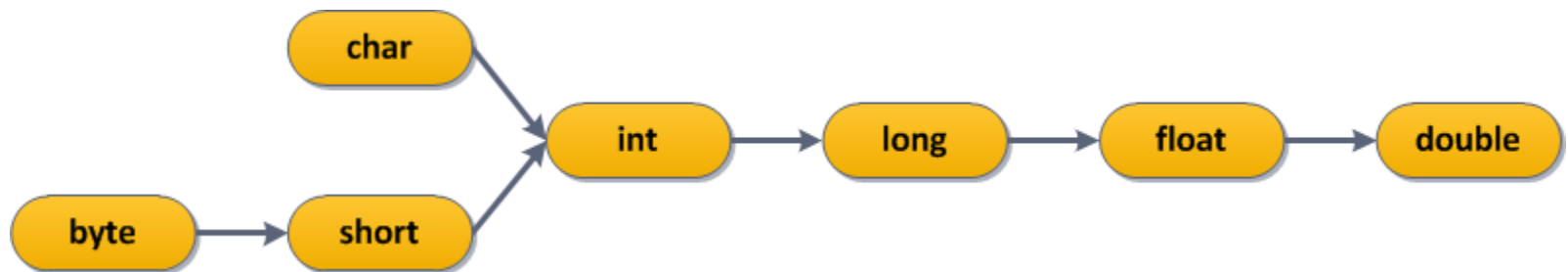
```
byte valorByte = Byte.parseByte("20");  
short valorShort = Short.parseShort("-3");  
int valorInt = Integer.parseInt("490");  
long valorLong = Long.parseLong("5000000");  
float valorFloat = Float.parseFloat("43.70");  
double valorDouble = Double.parseDouble("-421.894");  
boolean valorBoolean = Boolean.parseBoolean("true");
```

- É possível criar uma **String** a partir de um **valor contido num tipo primitivo**. Para isto utilize o método **valueOf** da classe **String**:

```
String v1 = String.valueOf(3); //byte
String v2 = String.valueOf(54); //short
String v3 = String.valueOf(-783); //int
String v4 = String.valueOf(9862L); //long
String v5 = String.valueOf(56.76f); //float
String v6 = String.valueOf(-127.54d); //double
String v7 = String.valueOf(true); //boolean
String v8 = String.valueOf('a'); //char
```

- A **conversão automática** de tipos ocorre quando você atribui um valor a uma variável de um tipo diferente ao do valor original:

```
int i;  
double d;  
i = 10;  
d = i; //OK, um int eh menor que um double
```





- A conversão automática de tipos **NÃO** ocorre quando você atribui um valor a uma variável de outro tipo cujo tamanho é menor em relação ao do tipo original:

```
double d;  
short s;  
d = 1.2345d;  
s = d; //ERRO - possible loss of precision
```



- É possível utilizar o recurso de *Casting* para ordenar ao Java fazer uma **conversão específica**;
- Para "fazer um cast" é só colocar o tipo desejado entre **parênteses** antes do valor a ser convertido;

```
//neste caso o cast nao eh requerido
```

```
double d = 10.5;
```

```
int i = (int) d;
```

```
//neste caso o cast eh requerido
```

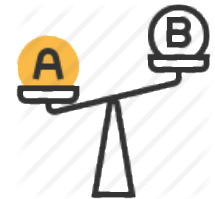
```
int vlInteiro = 25;
```

```
byte vlByte = (byte) vlInteiro ;
```

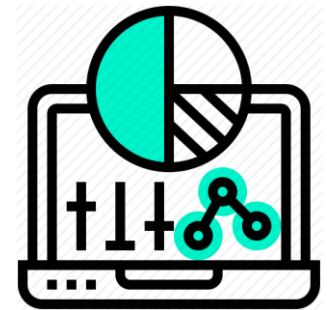


# TOMADA DE DECISÃO

Operador	Função
==	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a



Operador	Função
&&	E
	Ou
!	Não



- Sintaxe:

```
if (<condição lógica>) {  
    <instruções>  
}
```

```
if(x > 10){  
    System.out.println("X eh maior que 10");  
}
```



- Sintaxe:

```
if (<condição lógica>) {  
    <instruções>  
}  
else {  
    <instruções>  
}
```

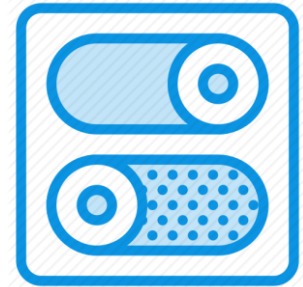
```
if(x > 10){  
    System.out.println("X eh maior que 10");  
} else {  
    System.out.println("X eh menor ou igual a 10");  
}
```

```
if(x > 10){  
    System.out.println("X eh maior que 10");  
} else {  
    System.out.println("X eh menor ou igual a 10");  
}
```

```
if(x > 10){  
    System.out.println("X eh maior que 10");  
} else if(x == 10) {  
    System.out.println("X eh igual a 10");  
} else {  
    System.out.println("X eh menor que 10");  
}
```



```
switch (<variável>) {  
    case <constante 1> :{  
        <instruções>  
        break;  
    }  
    case <constante 2> :{  
        <instruções>  
        break;  
    }  
    case <constante 3> :{  
        <instruções>  
        break;  
    }  
    default:{  
        break;  
    }  
}
```



```
switch(opcao){  
    case 1: {  
        System.out.println("Opcao 1(Novo) - selecionada");  
        break;  
    }  
    case 2: {  
        System.out.println("Opcao 2(Imprimir) - selecionada");  
        break;  
    }  
    case 3: {  
        System.out.println("Opcao 3(Sair) - selecionada");  
        break;  
    }  
    default: {  
        System.out.println("Opcao invalida, selecione 1,2 ou 3");  
    }  
}
```

# PRÁTICA

---



Crie um projeto chamado: **TomadaDecisao**, dentro dele resolva os descansos 1, 2 e 3, criando apenas as **classes de teste** (método `main()`):

1. Crie uma classe **TesteTamanho** em Java que peça o nome, a altura e o peso de 2 pessoas e apresente o nome da mais pesada e o nome da mais alta.
2. Crie uma classe **TesteEleitores** que leia a idade de 3 pessoas e informe a sua classe eleitoral:
  - não eleitor (abaixo de 16 anos)
  - eleitor obrigatório (entre 18 e 65 anos)
  - eleitor facultativo (16 , 17 anos ou maior que 65 anos)

Ao final da execução exibir o total de eleitores obrigatórios.

3. No projeto “Elevador” que criamos anteriormente, implemente nos métodos (ou crie novos métodos) a programação para que atendam as seguintes necessidades:

- O **construtor** deve definir os valores dos atributos `capacidade` e `quantidadeAndar`;
- O método **`entra()`** deve adicionar o valor passado como parâmetro ao atributo **`quantidadeAtualPessoas`**, desde que a capacidade do elevador não seja excedida. Após executar o procedimento citado, o método deve retornar a valor do atributo `quantidadeAtualPessoas`;

- O método **sai()** deve subtrair do atributo **quantidadeAtualPessoas** o valor passado como parâmetro, desde que o resultado desta operação seja um número maior ou igual a 0. Após executar o procedimento citado, o método deve retornar a quantidade de pessoas restantes no elevador;
- O método **sobe()** deve adicionar 1 ao atributo **andarAtual**, desde que o resultado desta operação não seja maior que o valor do atributo **quantidadeAndar**. Após executar o procedimento citado, o método deve retornar o valor **true** quando o valor for adicionado e **false** no caso contrário;
- O método **desce()** deve subtrair 1 do atributo **andarAtual**, desde que o resultado desta operação não seja menor que 0 (zero). Após executar o procedimento citado, o método deve retornar o valor **true** quando o valor for subtraído e **false** no caso contrário;
- Teste com os novos métodos!!!!

# Copyright © 2020 - 2022 Prof. Thiago T. I. Yamamoto

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).

*"A lógica pode levar de um ponto A a um ponto B.  
A imaginação pode levar a qualquer lugar"*  
Albert Einstein