



# Domain Driven Design with Java

---

## Type Casting

Prof<sup>o</sup> Dr Gustavo Molina

## O que são Tipos de Dados em Java?

- Em Java, um tipo de dado define o tamanho e o tipo de valores que uma variável pode armazenar.
- Existem dois grupos principais de tipos de dados:
  - **Tipos Primitivos**
  - **Tipos por Referência**

# Tipos Primitivos em Java

- Java possui 8 tipos de dados primitivos:
  - byte
  - short
  - int
  - long
  - float
  - double
  - boolean
  - char
- Esses tipos primitivos armazenam valores simples.

## Exemplo: Tipo `byte`

- `byte` é usado para economizar espaço em grandes arrays, sendo um número inteiro de 8 bits.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        byte myByte = 100;  
        System.out.println(myByte);  
    }  
}
```

## Exemplo: Tipo `short`

- `short` é um número inteiro de 16 bits.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        short myShort = 5000;  
        System.out.println(myShort);  
    }  
}
```

## Exemplo: Tipo `int`

- `int` é usado para armazenar números inteiros de 32 bits.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        int myInt = 100000;  
        System.out.println(myInt);  
    }  
}
```

## Exemplo: Tipo `long`

- `long` é usado para armazenar números inteiros de 64 bits.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        long myLong = 150000000000L;  
        System.out.println(myLong);  
    }  
}
```

## Exemplo: Tipo `float`

- `float` é usado para armazenar números decimais de 32 bits.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        float myFloat = 5.75f;  
        System.out.println(myFloat);    // Outputs 5.75  
    }  
}
```



## Exemplo: Tipo `double`

- `double` é usado para armazenar números decimais de 64 bits.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        double myDouble = 19.99d;  
        System.out.println(myDouble);    // Outputs 19.99  
    }  
}
```

## Exemplo: Tipo `boolean`

- `boolean` armazena apenas dois valores possíveis: `true` ou `false`.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        boolean isJavaFun = true;  
        boolean isFishTasty = false;  
        System.out.println(isJavaFun);  
        System.out.println(isFishTasty);  
    }  
}
```

## Exemplo: Tipo `char`

- `char` armazena um único caractere de 16 bits/2 bytes.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        char myChar = 'A';  
        System.out.println(myChar);    // Outputs A  
    }  
}
```

## Tipos por Referência

- Além dos tipos primitivos, Java utiliza tipos por referência para objetos.
- Exemplos: `String`, `Arrays`, `Classes`.
- Eles armazenam a referência à memória onde os objetos estão localizados, não os valores em si.

## Exemplo: Tipo `String`

- `String` é uma classe usada para armazenar uma sequência de caracteres (texto).
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        String greeting = "Hello World";  
        System.out.println(greeting);  
    }  
}
```

# O que é Casting em Java?

- Casting é o processo de conversão de um tipo de dado em outro.
- Em Java, existem dois tipos principais de casting:
  - **Widening Casting (automático):** conversão de um tipo menor para um tipo maior.
  - **Narrowing Casting (manual):** conversão de um tipo maior para um tipo menor.

# Widening Casting

- Ocorre automaticamente quando um valor de tipo menor é atribuído a um tipo maior.
- Exemplos:

```
class Main {  
    public static void main(String[] args) {  
        int myInt = 9;  
  
        double myDouble = myInt; // Widening Casting  
        System.out.println(myInt);           // Outputs 9  
        System.out.println(myDouble);        // Outputs 9.0  
    }  
}
```

# Narrowing Casting

- Deve ser feito manualmente, pois pode resultar em perda de dados.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        double myDouble = 9.78;  
  
        int myInt = (int) myDouble; // Narrowing Casting  
        System.out.println(myDouble); // Outputs 9.78  
        System.out.println(myInt); // Outputs 9  
    }  
}
```



# Comparação entre Widening e Narrowing Casting

- **Widening Casting:**

- Automático
- Sem perda de dados
- Exemplos: byte → short → char → int → long → float → double

- **Narrowing Casting:**

- Manual
- Pode resultar em perda de dados
- Exemplos: double → float → long → int → char → short → byte

# Casting de Objetos

- Em Java, você também pode fazer cast de objetos.
- Exemplo:

```
class Animal {  
    public void sound() {  
        System.out.println("Animal makes a sound");  
    }  
}
```

```
class Dog extends Animal {  
    public void sound() {  
        System.out.println("Dog barks");  
    }  
}
```

## Importante

- Entender o conceito de casting é crucial para evitar erros de compilação e execução em Java.
- Use Widening Casting quando possível para evitar perda de dados.
- Esteja atento ao usar Narrowing Casting, pois pode resultar em perda de precisão ou truncamento de valores.

# Conversão de Tipos em Java

- Em Java, é comum a necessidade de converter entre tipos primitivos e `String`.
- Existem métodos específicos para converter:
  - De tipo primitivo para `String`.
  - De `String` para tipo primitivo.

## Conversão de Tipo Primitivo para `String`

- Para converter tipos primitivos em `String`, usamos o método `String.valueOf()`.
- Este método está disponível para todos os tipos primitivos.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        int myInt = 100;  
        String myString = String.valueOf(myInt);  
        System.out.println(myString);  
    }  
}
```

## Conversão de `String` para Tipo Primitivo

- Para converter uma `String` em um tipo primitivo, utilizamos métodos como `Integer.parseInt()`, `Double.parseDouble()`, etc.
- Cada tipo primitivo possui seu próprio método de conversão.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        String myString = "100";  
        int myInt = Integer.parseInt(myString);  
        System.out.println(myInt);  
    }  
}
```

## Exemplo: String para double

- Para converter uma String em double, utilizamos `Double.parseDouble()`.
- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        String myString = "99.99";  
        double myDouble = Double.parseDouble(myString);  
        System.out.println(myDouble);  
    }  
}
```

## Exemplo: String para boolean

- Para converter uma String em boolean, utilizamos

`Boolean.parseBoolean()`.

- Exemplo:

```
class Main {  
    public static void main(String[] args) {  
        String myString = "true";  
        boolean myBool = Boolean.parseBoolean(myString);  
        System.out.println(myBool);  
    }  
}
```



# Resumo dos Métodos de Conversão

- **Tipos Primitivos para `String`:**

- `String.valueOf(int)`
- `String.valueOf(double)`
- `String.valueOf(boolean)`, etc.

- **`String` para Tipos Primitivos:**

- `Integer.parseInt(String)`
- `Double.parseDouble(String)`
- `Boolean.parseBoolean(String)`, etc.

## Erros Comuns e Exceções

- Se uma `String` não pode ser convertida, Java lança uma `NumberFormatException`.
- É importante tratar exceções ao converter `String` em tipos primitivos.

# Exemplo: Tratamento de Exceções

- Exemplo de tratamento de `NumberFormatException`:

```
class Main {  
    public static void main(String[] args) {  
        String myString = "abc";  
  
        try {  
            int myInt = Integer.parseInt(myString);  
        } catch (NumberFormatException e) {  
            System.out.println("Invalid number format");  
        }  
    }  
}
```

## Conclusão

- A conversão entre tipos primitivos e `String` é uma parte essencial da programação em Java.
- Compreender os métodos de conversão ajuda a evitar erros comuns e a melhorar a eficiência do código.
- Pratique diferentes cenários de conversão para dominar estas técnicas.

Perguntas?

Bora! Perguntas?