



Módulo 5



A decorative graphic on the left side of the slide. It features a large cyan hexagon with the number '1' inside. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and three connecting lines.

1

Tipos complexos, boxing (autoboxing), unboxing e casting

Tipos complexos de variáveis



Tipos complexos (Wrappers)

Primitivos	Wrappers
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double



A decorative graphic on the left side of the slide. It features a large central hexagon with a blue-to-teal gradient, containing the white number '2'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and teal. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines.

2

Boxing (Autoboxing) e unboxing



Boxing (Autoboxing)

Boxing ou Autoboxing é a conversão de tipos primitivos para classes wrappers correspondente.

Exemplos:

```
Boolean a = true;
```

```
Byte b = 1;
```

```
Character c = 'c';
```

```
Short s = 1;
```

```
Integer i = 1;
```

```
Long l = 1L;
```

```
Float f = 1.0F;
```

```
Double d = 1.0;
```





Unboxing

Unboxing é quando você quer converter o valor de um objeto Wrapper para o seu correspondente tipo primitivo.

Exemplos:

```
boolean a = new Boolean("True");  
byte b = new Byte("1");  
char c = new Character('c');  
short s = new Short((short)1);  
int i = new Integer(1);  
long l = new Long(1);  
float f = new Float(1.0f);  
double d = new Double(1.0);
```



A decorative graphic on the left side of the slide. It features a large central hexagon with a blue-to-cyan gradient, containing the number '3'. Surrounding this central hexagon are several smaller hexagons of varying shades of blue and cyan. Some of these smaller hexagons contain white icons: a lightbulb, a thumbs-up, a smartphone, a magnifying glass, and a gear. There is also a network-like icon with a central node and radiating lines, and a speech bubble icon. The entire graphic is set against a dark blue background.

3

Casting



Casting de primitivos


Casting é um recurso, que possibilita mudar um valor de um tipo para o outro. As vezes isso é feito automático, que chamamos de casting IMPLÍCITO e também pode ser feito por você, chamado de casting EXPLÍCITO.





Casting de primitivos

Tipo	Tamanho
boolean	1 bit
byte	8 bits
char	16 bits
short	16 bits
int	32 bits
float	32 bits
double	64 bits
long	64 bits



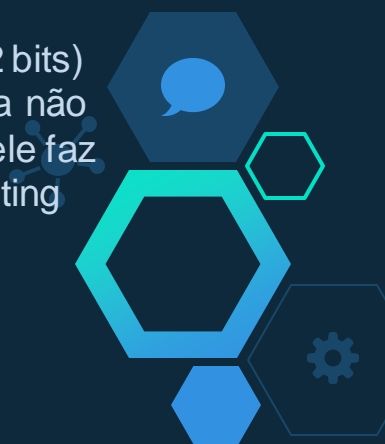


Casting explícito

```
int num1 = 10;  
short num2 = num1; (Erro de compilação)
```

Por que no caso acima o java não nos permite fazer `num2 = num1` direto ?

Por que `num1` é do tipo `int`, que é maior que `short`. Um inteiro(32 bits) é maior que um `short`(16 bits) à nível de armazenamento. O java não permite que um tipo maior seja convertido para um tipo menor, ele faz isso como medida de segurança, para isso é preciso fazer o casting explícito.





Casting explícito

```
int num = 10;  
short num2 = (short) num;
```

Neste código ele faz a conversão de um número maior para um número menor, mas neste ponto você está mostrando para ele que você sabe o que está fazendo.





Casting

Observações sobre tipos primitivos:

- boolean - Não pode ser moldado para nenhum outro tipo, tanto implicitamente quanto explicitamente.
- char - Nenhum outro tipo pode ser moldado para char.
- long é semelhante a um inteiro, mas com uma possibilidade maior de números.





Casting implícito

```
byte b = 1;  
short s = b;
```

Este é um casting implícito porque não foi preciso deixar claro que estamos convertendo um byte para short, porque byte é menor que short. Toda variável que é menor pode ser feito o casting implícito para qualquer outro tipo maior.





Casting

PARA:	byte	short	char	int	long	float	double
DE:	byte	short	char	int	long	float	double
byte	----	<i>Impl.</i>	(char)	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
short	(byte)	----	(char)	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
char	(byte)	(short)	----	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
int	(byte)	(short)	(char)	----	<i>Impl.</i>	<i>Impl.</i>	<i>Impl.</i>
long	(byte)	(short)	(char)	(int)	----	<i>Impl.</i>	<i>Impl.</i>
float	(byte)	(short)	(char)	(int)	(long)	----	<i>Impl.</i>
double	(byte)	(short)	(char)	(int)	(long)	(float)	----

