



# CURSO DE PROGRAMAÇÃO EM JAVA

Aula 5   
Expressões

1.

Expressões

# Expressões

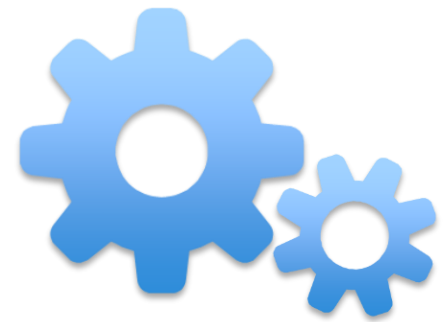
Uma Expressão em um programa é a parte da instrução que produz um valor, normalmente através do uso de operandos (valores) e operadores (indicam a operação a ser realizada).

Os operadores se organizam basicamente em três grupos:

- Aritméticos;
- Relacionais;
- Lógicos.



# Expressões

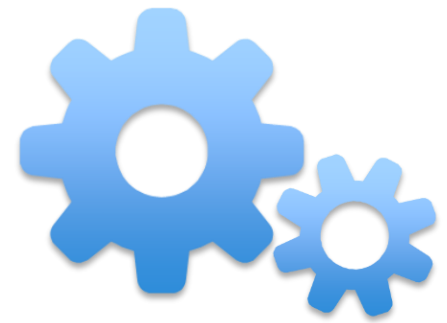


- Toda expressão escrita em Java tem um tipo que pode ser deduzido da estrutura da expressão e dos tipos de seus valores literais, variáveis e métodos.
- Se escrevermos uma expressão em um contexto em que o tipo da expressão não é apropriado, pode acontecer:
  - Erro em tempo de compilação
  - O contexto faz uma conversão implícita (quando possível) para um tipo aceitável pelo contexto

2.

# Conversão de Tipos

**Quais os tipos de converção?**

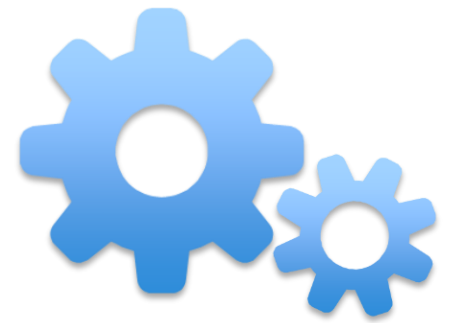


# Casting

- Conversão realizada de maneira explícita através do operador de cast
- Permite qualquer tipo de conversão em tempo de compilação, podendo gerar exceções

```
public static void main(String args[]){  
    float f = 12.5f;  
    int i;  
  
    //i = f; ERRO DE COMPILAÇÃO. Operador de cast necessário  
    i = (int) f;  
    System.out.println("Valor de f: " + f);  
    System.out.println("Valor de i: " + i);  
}
```

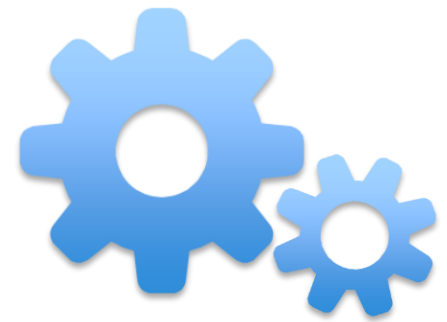
# Atribuição



- Ocorre quando o tipo de uma expressão é convertido para o tipo da variável que recebe seu valor

```
int intOper1 = 2;  
int intOper2 = 3;  
float resultFloat = 0.0f;  
  
//resultado armazenado em uma variável do tipo float  
//(conversão implícita)  
resultFloat = intOper1 + intOper2;
```

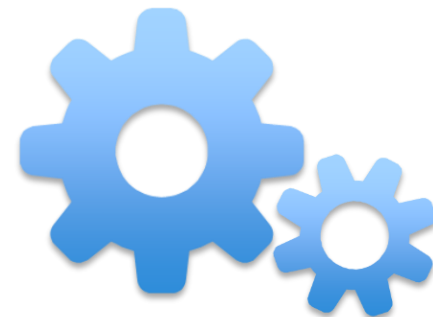
# String



- Possibilita que qualquer tipo de dado seja representado na forma de String

```
int resultInt = 0;  
float resultFloat = 0.0f;  
  
System.out.println("Resultado da operação (int): " +  
resultInt);  
System.out.println("Resultado da operação (float): "  
+ resultFloat);
```





# Conversões com Strings

- Conversões de tipos primitivos de dados para Strings

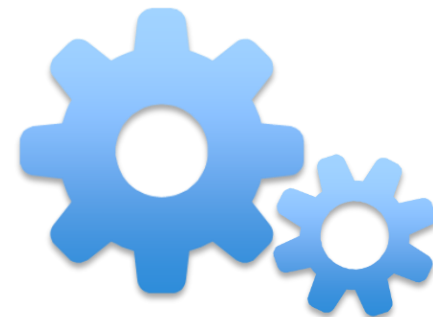
***String s = String.valueOf (tipo);***

Onde tipo pode ser uma variável do tipo:

- int, long, float, double, boolean.

- Conversão de String para tipos primitivos (Wrappers):

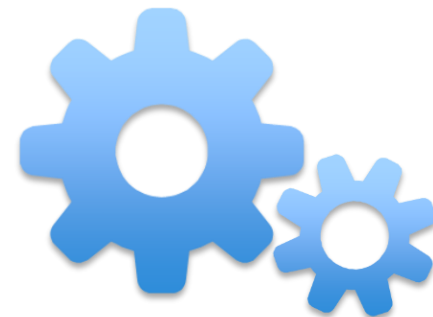
```
int i = Integer.parseInt(String);  
long l = Long.parseLong(String);  
double d = Double.parseDouble(String);
```



# Promoção Numérica

- Converte implicitamente um os operandos de uma expressão para um tipo em comum, de tal forma que a expressão possa ser operada

```
int intOper1 = 2;  
float floatOper1 = 3.6f;  
float resultFloat = 0.0f;  
  
//operador intOper1 é implicitamente tratado como  
float  
resultFloat = intOper1 + floatOper1;
```

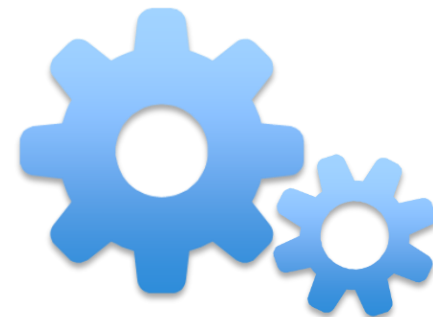


# Promoção Numérica

- Conversão automática

byte → short → int → long → float → double

```
int intOper1 = 2;  
float floatOper1 = 3.6f;  
float resultFloat = 0.0f;  
  
//operador intOper1 é implicitamente tratado como  
float  
resultFloat = intOper1 + floatOper1;
```



# Promoção Numérica

- De maior para menor precisão necessita o uso de casting

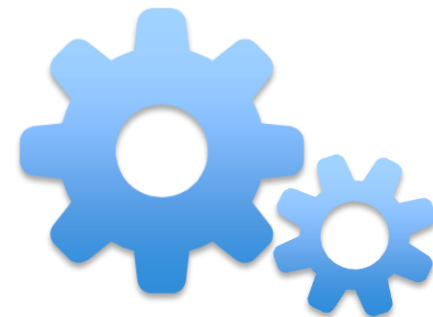
```
double d = 9997;  
int i = (int) d;
```

```
// i tem valor 9997
```

```
double d = 99.97;  
int i = (int) d;
```

```
// i tem valor 99
```

```
// Informações podem ser perdida
```



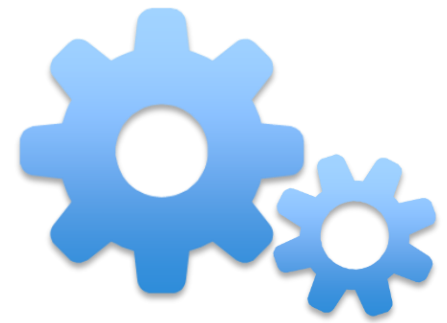
# Invocação de métodos

- Parâmetro real é implicitamente convertido para o tipo do parâmetro formal

```
//parâmetros do método: float op1 e float op2
public static float soma(float op1, float op2){
    return op1 + op2;
}

public static void main(String args[]){
    int opInt = 5;
    float opFloat = 2.7f;
    //parâmetros passados: int opInt e float opFloat
    System.out.println("Resultado da soma: " + soma(opInt, opFloat));
}
```

# Conversões entre tipos numéricos

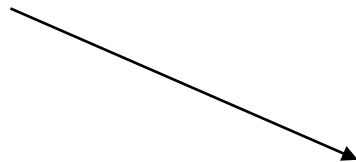


- Ponto flutuante
  - Em Java, todo literal de ponto flutuante é double, a não ser que seja explicitamente declarado como float

~~float 3.14;~~

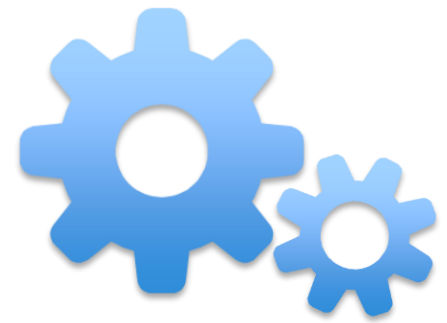


float 3.14f;



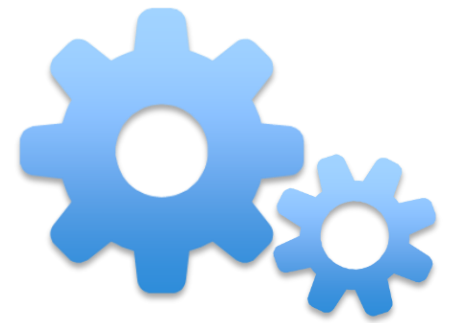
Erro de compilação:  
possible loss of precision

# Conversões entre tipos numéricos



- Para operações numéricas:
  - Se um dos operandos é do tipo double, o outro será tratado como um double no escopo da operação;
  - Senão, se um dos operandos for float, o outro será tratado como float;
  - Senão, se um dos operandos é do tipo long, o outro será tratado como long.
  - Senão, os operandos serão tratados como inteiros.

# Conversões entre tipos numéricos



- Por exemplo:

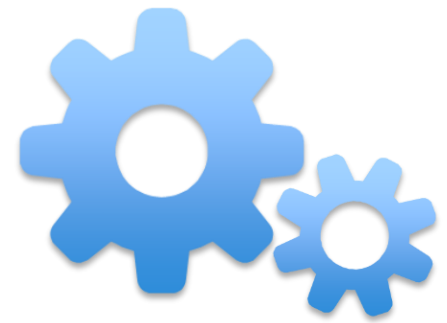
```
double d1 = 0.2;  
float f1 = 0.3f;  
<tipo> soma = d1 + f1;
```

Qual o tipo de dado de **<tipo>** para receber a soma acima?

**double** soma = d1 + f1;



# Conversões entre tipos numéricos



- Char
  - É necessário o uso de cast para converter inteiro em char
  - char pode ser considerado inteiro sem o uso de casting.

```
int i1 = 100;  
char c1 = 'c';  
char c2 = (char) i1;  
int i2 = c1;
```

```
System.out.println(c2);  
System.out.println(i2);
```



d  
99



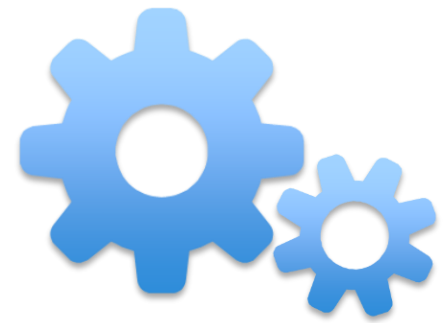
} **Tabela ASCII**

# ASCII TABLE



Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

# Conversões entre tipos numéricos



- Qual a saída do trecho abaixo?

```
char c1 = 71;  
char c2 = 82;  
char c3 = 69;  
char c4 = 77;  
char c5 = 73;  
char c6 = 79;  
  
System.out.println(c1 + c2 + c3 + c4 + c5 + c6);  
System.out.println(c1 + "" + c2 + "" + c3 + "" + c4 + "" + c5 + "" + c6);
```

Resposta:

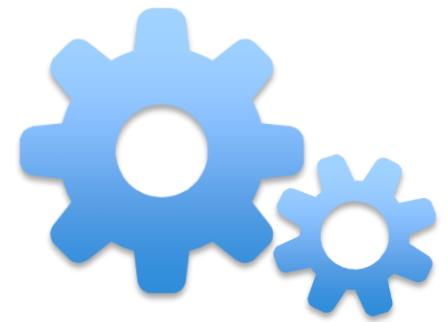
451  
GREMIO

# Tipos de conversão em Java

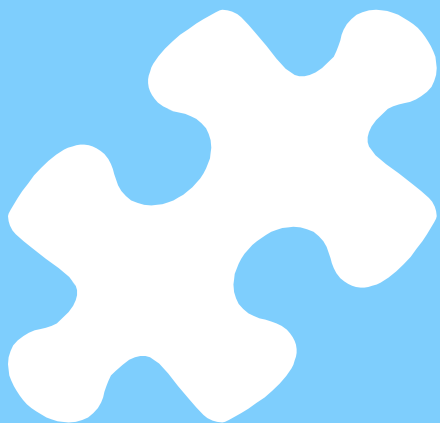


- **Conversão de Casting:** converte o tipo de uma expressão para um tipo explicitamente especificado pelo operador cast. Permite qualquer tipo de conversão em tempo de compilação, podendo gerar exceção em tempo de execução.
  - `int i = (int)12.5f;`
- **Conversão de atribuição:** converte o tipo de uma expressão para o tipo da variável.
  - `float f = i;`
- **Conversão de String:** permite que qualquer tipo seja convertido em uma String:
  - `System.out.println("(int)12.5f = " + i);`

# Tipos de conversão em Java



- **Promoção numérica:** converte um dos operandos de uma expressão para um tipo em comum de tal forma que a expressão possa ser operada:
  - $f = f * i;$
- **Conversão em invocação de método:** aplicado para argumento na chamada de métodos e construtores. Faz a mesma conversão que a conversão de atribuição faz.
  - `double d = Math.sin(f);`



# DESAFIO

E aí, vamos praticar?

# Expressões

Faça um algoritmo para resolver uma equação de segundo grau, considerando que existe solução real.

$$x = \frac{-b \pm \sqrt{\Delta}}{2.a}$$

$$\Delta = b^2 - 4.a.c$$

**Entrada:**

9140474.543020094  
16400517.28328856  
4838856.07944462

**Saída:**

X1 = -0.37  
X2 = -1.42

# Expressões

Faça um algoritmo que leia uma temperatura em graus Celsius e apresente-a convertida em graus *Fahrenheit*. A fórmula de conversão é:  $F = (9 * C + 160) / 5$ , na qual F é a temperatura em *Fahrenheit* e C é a temperatura em *Celsius*;

**Entrada:**

9663.207640922923

**Saída:**

17425.77



# Expressões

Faça um algoritmo que leia uma temperatura em *Fahrenheit* e a apresente convertida em graus *Celsius*. A fórmula de conversão é  $C = (F - 32) * (5 / 9)$ , na qual *F* é a temperatura em *Fahrenheit* e *C* é a temperatura em *Celsius*.

**Entrada:**

6465.16331395284

**Saída:**

3573.98

# Obrigado!

## **Alguma pergunta?**

Você pode me contatar em:  
[ywassef@hotmail.com](mailto:ywassef@hotmail.com)