

Java

Tipos Primitivos

Existem 3 maneiras básicas - mais de 3, porém, 3 maneiras básicas – de fazer declaração de uma Variáveis em Java.

Existem 4 Famílias que podem ser utilizadas na declaração de Variáveis.

Valore Numéricos

```
int idade = 3;
```

```
/*
```

Idade recebe 3. Neste caso, já é realizada a Declaração da Variável e a Atribuição.

```
*/
```

```
int idade = (int) 3;
```

```
/*
```

Para especificar tipos de valores específicos utiliza-se Typecast. O tipo entre parentes seguido do valor numérico, (int) 3, informa considere esse 3 como inteiro.

```
*/
```

```
Integer idade = new Integer(3);
```

```
/*
```

Desta forma, utilizamos Classes. Nesta opção, a Variável é um Objeto; que veremos mais a frente.

No momento, vale entender que sempre que tivermos a “new” dentro e uma declaração, esse “new” automaticamente está criando um Objeto; então temos que ter uma Classe referenciando este Objeto.

Não podemos dizer que “idade” é do Tipo “int”, pois a palavra “new”, logo depois do Identificador “idade”, indica que “idade” não é uma Variável, mas sim um Objeto.

Neste caso, também, notamos que o valor 3 será considerado como um valor Inteiro e colocado dentro do Objeto Idade.

Damos de “Wrapper Class” ao nome desta Classe; que quer dizer Invólucro (de envolver).

```
*/
```

```
int idade = 3;
```

Typecast

```
int idade = (int) 3;
```

Wrapper Class

```
Integer idade = new Integer(3);
```

```
float sal = 1825.54f;
```

```
/*
```

A Variável “sal”, neste caso, para guardar o valor do salário é Real.

float, sal recebe 1824.54 com um pequeno “f” no final, que indica que o número é float, ou seja, que o número é Real. Neste tipo de declaração, se não colocar o “f” no final, o próprio Java acusará erro.

```
*/
```

```
float sal = (float) 1825,54 ;
```

```
*/
```

float sal recebe o Typecast de float de 1825.54; desta forma, será informado que 1825.54 é para ser considerado como float.

```
*/
```

```
Float sal = new Float(1825.54);
```

```
/*
```

Utilizando Classes Wrapper Class, declara-se Float sal recebe new Float 1825.54.

```
*/
```

```
int idade = 3;
```

```
float sal = 1825.54f;
```

Typecast

```
int idade = (int) 3;
```

```
float sal = (float) 1825.54;
```

Wrapper Class

```
Integer idade = new Integer(3);
```

```
Float sal = new Float(1825.54);
```

Declaração de Variáveis Caracteres

```
char letra = 'G';
```

```
/*
```

O observe que no Java o 'G' possui Aspas Simples ('G').

A utilização do Tipo char só ocupa uma única letra. Para que possa ocupar mais de uma letra, será necessário utilizar uma Classe; que logo, logo, veremos.

```
*/
```

```
char letra = (char) 'G';
```

```
/*
```

Para declarar utilizando Typecast, informe char letra recebe o Typecast de 'G'; desta forma, será informado que o 'G' deve ser considerado como do Tipo char.

```
*/
```

```
Character letra = new Character('G');
```

```
/*
```

A terceira maneira de declarar char é utilizando a Classe Wrapper (Wrapper Class) que é "Character".

Desta forma, letra recebe new Character G.

```
*/
```

```
int idade = 3;
```

```
float sal = 1825.54f;
```

```
char letra = 'G';
```

Typecast

```
int idade = (int) 3;
```

```
float sal = (float) 1825.54;
```

```
char letra = (char) 'G';
```

Wrapper Class

```
Integer idade = new Integer(3);
```

```
Float sal = new Float(1825.54);
```

```
Character letra = new Character('G');
```

Declaração de Tipos Booleans

```
boolean casado = false;
```

```
/*
```

```
Boolean casado recebe falso.
```

```
*/
```

```
boolean casado = (boolean) false;
```

```
/*
```

```
Utilizando Typecast, boolean casado recebe boolean de falso.
```

```
*/
```

```
Boolean casado = new Boolean(false);
```

```
/*
```

```
Utilizando Classes Involucras (Wrapper Class), boolean casado recebe new boolean falso.
```

*/

```
int idade = 3;  
  
float sal = 1825.54f;  
  
char letra = 'G';  
  
boolean casado = false;
```

Typecast

```
int idade = (int) 3;  
  
float sal = (float) 1825.54;  
  
char letra = (char) 'G';  
  
boolean casado = (boolean) false;
```

Wrapper Class

```
Integer idade = new Integer(3);  
  
Float sal = new Float(1825.54);  
  
Character letra = new Character('G');  
  
Boolean casado = new Boolean(false);
```

Como informado no segundo parágrafo deste texto, os 4 tipos NÃO SÃO 4 TIPOS, MAS SIM 4 FAMÍLIAS.

Diante das informações obtidas, podemos montar a seguinte tabela:

Família	Tipo Primitivo	Classe Invólucro	Tamanho (que ocupa na memória)	Exemplo De atribuição
Lógico	boolean	Boolean	1 bit	true

A Família dos lógicos só possui um Tipo relacionado, sendo o boolean. A Classe Invólucro para boolean é Boolean com “B” maiúsculo.

Uma Variável do Tipo boolean só ocupa um bit na memória, isto é, “0” ou “1”. Por exemplo: True ou False, que será convertido para o valor de 1 bit.

Família	Tipo Primitivo	Classe Invólucro	Tamanho (que ocupa na memória)	Exemplo De atribuição
Lógico	boolean	Boolean	1 bit	True
Literais	char	Character	1 Byte	‘A’

Os literais possuem apenas 1 Tipo Primitivo, mas 2 Classes Invólucas.

O primeiro Tipo Primitivo é o char, que tem Classe Invólucra Character e ocupa 1 Byte de memória; ocupa esse espaço devido a ocupar o tamanho de uma letra, logo um Byte.

Exemplo: a letra ‘A’. Todo tipo char é colocado entre aspas simples.

Família	Tipo Primitivo	Classe Invólucro	Tamanho (que ocupa na memória)	Exemplo De atribuição
Lógico	boolean	Boolean	1 bit	True
Literais	char	Character	1 Byte	‘A’
	-	String	1 Byter/letra	“JAVA”

Sempre que precisar armazenar mais de uma letra, deverá ser utilizada a Classe Invólucra String. No Java NÃO existe a Tipo String, mas sim a Classe Invólucra String.

Os Objetos da Classe Invólucra String ocupam 1 Byte para cada letra. Neste caso, a palavra “JAVA” ocupa quatro Bytes na memória, devido as quatro letras e deve ser utilizada com aspas duplas.

Família	Tipo Primitivo	Classe Invólucro	Tamanho (que ocupa na memória)	Exemplo De atribuição
Lógico	boolean	Boolean	1 bit	True
Literais	char	Character	1 Byte	‘A’
	-	String	1 Byter/letra	“JAVA”
Inteiros	byte	Byte	1 Byte	127

Os Inteiros possuem 4 Tipos relacionados.

O Tipo byter é o menor de todos os números Inteiros. Sua Classe Invólucra é Byte e o tamanho é 1 Byte, desta forma, não aceita números muito grande; sendo o maior 127.

Família	Tipo Primitivo	Classe Invólucro	Tamanho (que ocupa na memória)	Exemplo De atribuição
Lógico	boolean	Boolean	1 bit	True

Literais	char	Character	1 Byte	'A'
	-	String	1 Byte/letra	"JAVA"
Inteiros	byte	Byte	1 Byte	127
	short	Short	2 Bytes	32 767
	int	Integer	4 Bytes	2 147 483
	long	Long	8 Bytes	2 ⁶³

Para valores até 32 767 utiliza-se o Tipo byte, que possui a Classe Invólucra Byte e tamanho de 2 Bytes.

Para valores até 2 147 483 utiliza-se o Tipo int, que possui a Classe Invólucra Integer e tamanho de 4 Bytes.

Para valores maiores que 2 147 483, mais especificamente valor de 2⁶³, utiliza-se o Tipo long, que possui a Classe Invólucra Long e tamanho de 8 Bytes.

Família	Tipo Primitivo	Classe Invólucro	Tamanho (que ocupa na memória)	Exemplo De atribuição
Lógico	boolean	Boolean	1 bit	True
Literais	char	Character	1 Byte	'A'
	-	String	1 Byte/letra	"JAVA"
Inteiros	byte	Byte	1 Byte	127
	short	Short	2 Bytes	32 767
	int	Integer	4 Bytes	2 147 483
	long	Long	8 Bytes	2 ⁶³
Reais	float	Float	4 Bytes	3.4e ⁺³⁸
	double	Double	8 Bytes	1.8e ⁺³⁰⁸

Na Família dos reais temos o Tipo float, que possui a Classe Invólucra Float com tamanho de 4 Bytes em memória; sendo atribuído como (3,4 * e)⁺³⁸ e o Tipo double que é o dobro do Tipo Float.

No Java, o Tipo da Variável é muito importante; chamamos de uma Linguagem Fortemente Tipada.

Em muitos casos será necessário fazer Conversões entre os tipos, através de Conversões ou Especificações.

Por exemplo:

Podemos dizer que o valor 3, da idade acima, pode ser Real (3.0) ou Inteiro (3). Ou seja, há como Especificar que o valor 3 é Real e é Inteiro ao mesmo tempo.