



Linguagem Java

Tito Kenzo

Objetivos da Aula

- Sintaxe e convenções
- Estrutura da linguagem
- Tipos primitivos

Java Keywords, Reserved Words, and a Special Identifier

Keywords available since 1.0

Keywords no longer in use

Keywords added in 1.2

Keywords added in 1.4

Keywords added in 5.0

Keywords added in 9.0

Reserved words for literals values

Special identifier added in 10

if	implements	boolean	assert
else	extends	try	enum
continue	interface	catch	module
break	class	finally	requires
for	static	throw	transitive
do	final	throws	exports to
while	return	new	uses
switch	transient	this	provides
case	void	super	with
default	byte	instanceof	opens to
private	short	native	
protected	int	synchronized	true
public	long	volatile	false
import	char	goto	null
package	float	const	
abstract	double	strictfp	var

Notes

✿ Keywords and Literals cannot be used as identifiers (names of classes, variables, methods, and so on).

Classes

- Classe e Objetos são dois conceitos da OO (veremos mais adiante).
- O código Java é estruturado em **Classes**.
- Classes são organizadas em pacotes (package).

```
package meuprojeto;  
class Retangulo {  
    int base;  
    int altura;  
    int calculaArea() {  
        return base*altura;  
    }  
}
```

Sintaxe básica

- Usamos o operador `=` para atribuições:

```
int numero = 1;
```

- Usamos o operador `;` para delimitarmos os comandos:

```
int numeroUm = 10;
```

```
int numeroDois = 13;
```

- Blocos de Código devem ser contidos entre chaves `{ }`.

```
class Bola{  
    //some code  
}
```

- Indentação e espaços ajudam na leitura do código, mas são sintaticamente irrelevantes.

Comentários

- Em Java é possível comentar código de três formas diferentes:
 - **// Comentários de uma linha**
 - **/* comentários de várias linhas */**
 - **/** comentários para documentação */**
É utilizado pela ferramenta javadoc.
Deve preceder a declaração de uma variável, um método ou uma classe. O documento é gerado no formato HTML.

Operadores

Operador	Descrição
=	Atribuição
!	Operador lógico de negação (NOT)
> , < , >= , <=	Operadores lógicos de desigualdade
== , !=	Operadores lógicos de igualdade e diferença
&& ,	E lógico (AND) , OU lógico (OR)
+, - , * , / , %	Adição, Subtração, Multiplicação, Divisão, Módulo
+= , -= , *= , /= , %=	Operação com atribuição. "x += 2"; é o mesmo que "x = x + 2";
++ , --	Incremento (+1), decremento (-1) "x++; " é o mesmo que "x = x+1"


Tipos primitivos

TIPO	DESCRIÇÃO
boolean	Pode assumir o valor true ou o valor false
char	Caractere em notação Unicode de 16 bits. Serve para a armazenagem de dados alfanuméricos. Também pode ser usado como um dado inteiro com valores na faixa entre 0 e 65535.
byte	Inteiro de 8 bits em notação de complemento de dois. Pode assumir valores entre $-2^7=-128$ e $2^7-1=127$.
short	Inteiro de 16 bits em notação de complemento de dois. Os valores possíveis cobrem a faixa de $-2^{15}=-32.768$ a $2^{15}-1=32.767$
int	Inteiro de 32 bits em notação de complemento de dois. Pode assumir valores entre $-2^{31}=-2.147.483.648$ e $2^{31}-1=2.147.483.647$.
long	Inteiro de 64 bits em notação de complemento de dois. Pode assumir valores entre -2^{63} e $2^{63}-1$.
float	Representa números em notação de ponto flutuante normalizada em precisão simples de 32 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável por esse tipo é $1.40239846e-46$ e o maior é $3.40282347e+38$
double	Representa números em notação de ponto flutuante normalizada em precisão dupla de 64 bits em conformidade com a norma IEEE 754-1985. O menor valor positivo representável é $4.94065645841246544e-324$ e o maior é $1.7976931348623157e+308$


Convenções de Nomenclaturas

- Java diferencia maiúsculas de minúsculas.
- Use o padrão “camel-case”.
- Classes devem ser substantivos, com a primeira letra maiúscula.
- Variáveis começam com letra minúscula.
- Constantes devem ser em caixa-alta.
- Métodos devem ser verbos e começar com letra minúscula.

```
package: com.oracle.demos.animals
class:   ShepherdDog
variable: shepherdDog
constant: MIN_SIZE
method:  giveMePaw
```



```
package: animals
class:   Shepherd Dog
variable: _price
constant: minSize
method:  xyz
```



Modificadores de Acesso

Descrevem a visibilidade de classes, variáveis e métodos.

- **public** – visível para qualquer outra classe.
- **protected** – visível para classes que estão no mesmo pacote ou subclasses.
- **<default>** – visível apenas para classes dentro do mesmo pacote.
- **private** – visível apenas dentro a mesma classe.

*default quer dizer que nenhum modificador foi explicitamente usado.

Exemplo

Nome do arquivo deve ser igual ao da classe pública.
"HelloWorld.java"

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

Toda **aplicação** é executada através do método **main**, que possui sempre esta declaração

Compilação

C:\> javac HelloWorld.java //Compilador

- Erros comuns de compilação:
 - **Javac: Command not found** – variável de ambiente PATH não definida corretamente.
 - **HelloWorld.java: 4: Public class HelloWorld must be defined in a file called “HelloWorld.java”** – Apenas uma classe pública em Java pode ser declarada em cada arquivo .java, e o nome deste arquivo deve ter o mesmo nome desta classe pública.

Declaração de Classes

Qualquer identificador válido que já não esteja sendo usando

public, private,
final abstract ...

```
<modificadores> class <nome> {  
    <declaração de atributos>  
    <declaração de construtores>  
    <declaração de métodos>  
}
```

Exemplo

```
public class Retangulo  
{  
    int base, altura;  
    public int calculaArea(){return base*altura;}  
}
```

Declaração de Atributos

`<modificadores> <tipo> <nome> [=<valor inicial>];`

Modificadores
public, final, static

Tipo
byte, short, int, char,
float, double...

Nome
Qualquer identificador
válido e já não esteja
sendo usando

Valor inicial
Pode ser atribuído no
momento da
declaração do atributo

Exemplos

```
public int x;
```

```
static double calc = 45.76;
```

```
String nome = "Rita Cadillac";
```

Declaração de Métodos

```
<modificadores> <tipo retornado> <nome> (<parâmetros>) {  
    <expressão>  
}
```

Tipo retornado
void, byte, short, int,
char, float, double...

Parâmetros
Lista o número e tipo
dos argumentos que o
método pode receber,
separados por vírgula

Exemplo

```
public int calculaArea (int b, int a) {  
    int base = b, altura = a;  
    return base * altura;  
}
```