



Introdução à linguagem Java

Apontamentos sobre as primeiras noções da linguagem Java

Page

Para melhor conhecer a linguagem Java vamos utilizar exemplos de programas

- Assim, vamos começar pelo primeiro programa de todos:
 - O Hello World
 - Este programa simplesmente apresenta no ecrã a frase "hello world"

Java ▾

```
class HelloWorld {  
    // o método main é o sítio onde tudo começa  
    public static main void (String []args ){  
        System.out.print("Hello World");  
    }  
}
```

- Com este programa aprendemos já algumas coisas sobre a linguagem Java:
 - Os ficheiros que contêm o código fonte têm a extensão java
 - Cada ficheiro tem de ter uma declaração do tipo

Java ▾

```
// aqui não há código  
  
class NomeClasse {  
  
    // o código fica aqui dentro  
  
}
```

- Todo o código tem de estar dentro da declaração da classe
- O nome do ficheiro TEM de ser o mesmo da classe
- A primeira instrução a ser executada é a primeira que está dentro do main
 - Este nome não é opcional!

Java ▾

```
class HelloWorld {  
    // o método main é o sítio onde tudo começa  
    public static main void (String []args ){  
        System.out.print("Hello World");  
    }  
}
```

- O Java é case-sensitive:
 - Isto significa que faz a distinção entre maiúsculas e minúsculas

- A função main, como consequência tem de se chamar main e não Main ou MAIN
- Todas as palavras reservadas do Java são em minúsculas
- Um bloco de código (conjunto de instruções) tem de estar dentro de { }
- A { indica que onde começa o bloco
 - No exemplo indica onde começa a class HelloWorld e onde começa o método main
- A } indica onde termina o bloco de código
 - No exemplo indica onde termina a class HelloWorld e onde termina o método main

Java ▾

```
class HelloWorld { // Início do bloco da classe

    public static main void (String []args ){ // Início do bloco do método main
        System.out.print("Hello World");
    } // Fim do bloco do método main
} // Fim do bloco da classe
```

- Qualquer instrução tem de estar dentro de um bloco de código (entre { })
- Mas dentro de um bloco de código { } podem estar várias instruções
- Cada instrução em Java tem de terminar com um ;
- Normalmente só se coloca uma instrução por linha, mas a linguagem não define isso
- O Java tem um conjunto de instruções muito pequeno
- Não tem, por exemplo, nenhuma instrução para escrever no ecrã
- Para muitas tarefas o Java recorre ao uso de métodos que estão armazenados em bibliotecas
 - A biblioteca do Java é muito vasta
- É esse o caso do método print
- De reparar que todos os métodos terão de usar ()
 - Reparar no main
 - O uso de () a seguir a um texto indica que se trata de um método

❑	Será que	seraMetodo();	é um método?
❑	E	estaNaoE	é um método?
❑	E	estaE(3, 5);	é um método?
- O método printf é usado para escrever texto no ecrã
- Em Java considera-se texto o que está dentro de " "
- Para imprimir no ecrã a frase Hello World o código é:

Java ▾

```
System.out.print("Hello World");
```

- os () significam que print é um método
- as " " significam que estamos a trabalhar com texto
- o ; significa que a instrução terminou
- Com isto já temos algumas ferramentas para escrever programas
- Mas ainda existe uma parte que não serve para nada, em termos de programa
- Esse texto serve apenas para quem está a ler o código fonte e é completamente ignorado pelo compilador
- São os **comentários**
- Os comentários são externamente úteis para documentar um programa
- Com eles podem-se
 - Descrever, na linguagem nativa dos programadores, o que faz o código
 - Indicar quais as limitações de um método

- Indicar quem é o autor, versão, etc...
- Em Java existem 2 modos de incluir comentários

/ comentário em bloco */*

// comentário em linha

- Alguns cuidados com os comentários:
 - Comentários dentro de comentários

```
/* este programa simplesmente escreve
no ecrã a frase Hello World

isso é feito na linha
System.out.print( "Hello World" ); /* isto imprime Hello World */

Agora esta linha já não é comentário,
é preciso ter cuidado com isto
*/
```

- Em Java privilegiam-se os *//* em vez dos */* */*



Tipos de Dados