

Exercício – Criar o primeiro programa em java

REGRAS PARA TODOS OS EXERCÍCIOS EM JAVA

- DEVE CRIAR UMA PASTA COM O SEU NOME E COLOCÁ-LA NO DIRETORIO C:
- NÃO USAR ESPAÇOS NEM CARACTERES "INVÁLIDOS" PARA O JAVA.
- TER SEMPRE ATENÇÃO, AOS NOMES DADOS AOS FICHEIROS QUE DEPOIS IRÃO CRIAR A CLASS PUBLICA. PARA QUE NÃO HAJA ERROS ACONCELHO A USAR SEMPRE O MESMO NOME QUE SERÁ APRESENTADO NOS EXEMPLOS E GUARDAR COMO TAL TAMBÉM.
- LÓGICO TUDO SEMPRE DENTRO DA PASTA COM O VOSSO NOME.
- LEMBRANDO QUE AO EXECUTAR O JAVA TEM DE ABRIR O DIRETORIO EM QUESTÃO, LOGO O VOSSO NOME APARECERÁ NO PRINT.
- TODOS OS EXERCÍCIOS SERÃO FEITOS ATÉ ORDEM CONTRÁRIA NUM BLOCO DE NOTAS, COM EXTENSÃO JAVA

Comentários

Aprenda a comentar!

Apesar de não executar nenhuma ação os comentários de um programa são muito importantes. Através dos comentários vocês fazem com que o vosso programa seja mais claro e compreensível aos outros e a si mesmo daqui a algum tempo. Podem acreditar.

Vai se sentir bem melhor quando encontrar os seus próprios programas comentários.

No Java são permitidos três tipos de comentários:

» Comentário de linha

» Comentário de múltiplas linhas

» Comentário de documentação

Comentário de linha

Os comentários de linha permitem ao programador descrever o que ocorre numa determinada linha.

O Java ignora tudo o que se encontra depois do símbolo `//` até o final da linha.

Vejamos alguns exemplos:

```
public class Comentarios1{  
    public static void main (String args[]){  
        System.out.println("Olá !");    // Imprime a frase "Olá !"   
    }    // Fim do método main  
} // Fim da declaração de classe Comentarios1
```

As frases "Imprime a frase "Olá!"" , "Fim do método main" e "Fim da declaração de classe Comentarios" são comentários de linha.

O Java sabe que deve ignorar estas frases pois antes dela encontrou os símbolos `//`.

Experimente. Execute e coloque um print num ficheiro word. Pode copiar o código deixado acima.

Comentários de múltipla linha

Os comentários de múltipla linha permitem que tudo o que se encontra entre o início e o fim dos limites do comentário sejam ignorados pelo Java.

```
/*  
Classe "Comentários2"  
  
Todo este texto esta sendo ignorado pelo Java e tem como única função explicar o  
funcionamento do código do programa.  
*/  
  
public class Comentarios2{  
    public static void main (String args[]){  
        System.out.println("Olá. De novo !");  
    }  
}
```

No exemplo acima tudo o que estiver entre os caracteres "/*" e "*/" será ignorado pelo Java e tem como finalidade comentar a aplicação.

Experimente. Execute e coloque um print num ficheiro word. Pode copiar o código deixado acima.

Comentário de documentação.

O comentário de documentação é semelhante ao comentário de múltiplas linhas.

Ele difere nos delimitadores utilizados e na sua função. São utilizados como delimitadores os símbolos `"/**"` – delimitador inicial - e `"*/"` – delimitador final. O objetivo do comentário de documentação é ser utilizado pelo programa **javadoc** para gerar uma documentação para o programa. Mais tarde entraremos em detalhes sobre como usar o **javadoc** para gerar documentação.

Definir uma classe

Todo programa Java deve ter pelo menos uma definição de classe. Uma definição de classe consiste numa palavra reservada **public** (como já falamos) seguida da palavra reservada **class**, seguida do nome da classe que por convenção deve começar com letras maiúsculas. O nome da classe é um identificador. Um identificador pode conter letras, dígitos, sublinhados e o símbolo de cifrão. Um identificador não pode começar com um dígito. Um identificador não deve começar com um cifrão.

Veja abaixo um exemplo de uma definição de classe:

```
public class Classe1 {  
}
```

Perceba que além das regras citadas acima a definição de classe contém um par de chaves. O que estiver contido dentro deste par de chaves é o que chamados de corpo de definição da classe.

No corpo de definição da classe estarão os métodos (ações) necessários para o funcionamento das classes, assim como, os demais itens necessários para “dar vida” à classe.

“Dar vida” a uma classe

1) Palavra reservada – Uma palavra reservada é uma palavra que o Java utiliza internamente. As palavras reservadas não devem ser utilizadas como nome de classes ou nome de variáveis. Estudaremos sobre variáveis mais a frente.

2) Um identificador não deve começar pelo uso de cifrão porque o Java costuma utilizar identificadores iniciados por cifrões internamente. Quando for dar nome a uma classe procure sempre começar com letras maiúsculas.

Para que uma classe execute determinada tarefa a classe deve ter ao menos uma definição de método (definição de ação).

Veja um exemplo de definição de método:

```
Public static void main (String args[]){  
}
```

Por enquanto não entraremos em detalhes sobre o que significa cada parte da definição do método. O importante é vocês saberem que toda classe daqui pra frente terá na definição do seu corpo a definição de um método main.

Exibir uma mensagem

Para exibir uma mensagem no Java utilizaremos a instrução “println”. Esta instrução permite imprimir a mensagem desejada e ainda posiciona o cursor no início da próxima linha da tela.

Vamos ver o exemplo de uma instrução que exibe a mensagem “Bem-vindo a programação Java!”.

```
System.out.println("Bem-vindo a programação Java!");
```

Outra forma de imprimir mensagens

Também pode imprimir a mesma mensagem da seguinte forma:

```
System.out.print("Bem-vindo a programação Java!");
```

A diferença é que o cursor não será posicionado no início da linha seguinte e sim imediatamente após o ponto de interrogação.

Juntar tudo

Juntando todos os tópicos deste capítulo teremos o nosso primeiro programa em Java. Este programa permite a impressão de uma mensagem de boas vindas na tela do computador.

```

/*
 *
 */

public class BemVindo{

    public static void main (String args[]) {

        System.out.println("Bem-vindo a programação Java!");

    } // método main

} // class BemVindo

```

**Experimente. Execute e coloque um print num ficheiro word. Insira um comentário entre os /* * **

Atenção:

» O Java diferencia letras maiúsculas e letras minúsculas. Tenha cuidado para digitar o texto exatamente como esta acima.

» Ao terminar de digitar o texto salve-o com o mesmo nome da classe e com a extensão Java. O texto acima deve ser salvo em um arquivo chamado "BemVindo.java".

Entender o que cada linha faz

```
/*
```

"Avisa" ao Java que irá iniciar um comentário de várias linhas. Tudo o que estiver escrito deste ponto até encontrar o conjunto de caracteres "*/" deverá ser ignorado pelo Java.

BemVindo – Imprime uma mensagem de boas vindas

Comentário – A Indicar algo

```
*/
```

"Avisa" ao Java que o comentário de várias linhas esta encerrado.

```
public class BemVindo{
```

Determina o início da definição da classe BemVindo.

```
public static void main (String args[]) {
```

Determina o início da definição do método main.

```
System.out.println("Bem-vindo a programação Java!");
```

Imprime a mensagem "Bem-vindo a programação Java!"

```
} // método main
```

Encerra a definição do método main com o caractere "}" e faz um comentário de uma única linha "// método main".

```
} // class BemVindo
```

Encerra a definição da classe BemVindo com o caractere "}" e faz um comentário de uma única linha "// class BemVindo".

Compilar e executando o nosso programa

No prompt de comando digite *javac* e o nome do arquivo para compilar o nosso programa.

```
javac BemVindo.java
```

Para executar o programa vá ao prompt de comando e digite *java* e nome da classe, sem a extensão.

```
java BemVindo
```

Pronto! Escrevemos o nosso primeiro programa em Java.

Para finalizar tente fazer o seguinte:

» Escreva um programa que exiba a frase, algo e utilize três instruções *System.out.print*.

Execute e coloque um print num ficheiro word.