



METODOLOGIA

- › Interprete o documento calmamente e com atenção.
- › Acompanhe a execução do exercício no seu computador.
- › Não hesite em consultar o formador para o esclarecimento de qualquer questão.
- › Não prossiga para o ponto seguinte sem ter compreendido totalmente o ponto anterior.
- › Caso seja necessário, execute várias vezes o exercício até ter compreendido totalmente o processo.

CONTEÚDO PROGRAMÁTICO

1. [Compilação e Execução](#)

1. Compilação e Execução

Para percebermos a sintaxe do Java e os conceitos do exercício anterior, vamos fazer um pequeno exercício. Vamos aprender a compilar e executar um programa a partir da linha de comandos.

- › **Abra o Notepad++**
- › **Introduza** o seguinte texto:

```
/*
Este bloco representa a estrutura base de um programa JAVA
*/

public class Exemplo {

    public static void main (String[] args) {

        // Escrita de dados no ecrã
        System.out.println("Exemplo de um programa em JAVA: Hello World!!");

    } // fim do método main

} // fim da classe Exemplo
```

› **Grave** o ficheiro com o nome `Exemplo.java` na sua pasta de aluno P:

📄 O nome dos ficheiros Java tem que ser igual ao nome da classe que contêm com a extensão **java**. De ter em conta que o uso de letras maiúsculas e minúsculas têm implicações nesta regra.

› **Abra** uma **linha de comandos**

📄 Todos os comandos que se seguem são instruções para o interpretador de linha comandos (uma aplicação do sistema operativo). **Estes têm de terminar obrigatoriamente pressionando a tecla** `ENTER`.

› **Execute** o seguinte **comando** na **linha de comandos** que abriu para mudar para a sua pasta de aluno:

P:

- › **Execute** o seguinte **comando** na **linha de comandos** para compilar o seu programa:

```
javac Exemplo.java
```

📄 Caso o sistema não apresente nenhuma mensagem, significa que o programa foi compilado com sucesso. Confirme que foi criado o ficheiro `Exemplo.class` na sua pasta de aluno **P**:

- › **Execute** o seguinte **comando** na **linha de comandos** executar o seu programa:

```
java Exemplo
```

📄 Para executar programas, deverá omitir a extensão `.class`, pois esta é assumida automaticamente.

- › **Analise** o resultado

Vamos agora analisar o programa `Exemplo` em detalhe:

Os caracteres `/*` e `*/` envolvem uma secção de comentários, isto é, instruções que não são compiladas, servindo apenas ajudar os programadores a perceber o código do programa.

```
/*  
    Esta classe representa a estrutura base de um programa JAVA
```

```
*/
```

Os caracteres `//` envolvem uma linha de comentários. Tudo o que está para a frente destes caracteres e até ao final da linha é considerado comentário.

```
// Escrita de dados no ecrã  
  
} // fim do método main  
  
} // fim da classe Exemplo
```

Os caracteres `{` e `}` representam o início e fim de uma secção de código, respetivamente. Uma classe java deve estar declarada dentro de uma secção “class”:

```
class <NOME DA CLASSE> {  
  
    <Código da classe aqui>  
  
}
```

📖 O termo `public` diz respeito ao âmbito de execução da classe, do método ou da variável à qual está associado. Mais à frente iremos aprofundar este tema.

Cada classe deverá ter um método `main` se se tratar do ficheiro de arranque de um programa, e deve cumprir **sempre** a seguinte definição:

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

📖 Os detalhes desta definição serão abordados nos próximos módulos.

📄 No Java existe diferença entre maiúsculas e minúsculas, ou seja, “String” e “string” representam coisas diferentes e a última irá gerar um erro.

› **Altere** o código da classe `Exemplo.java`, escrevendo “string” em vez de “String” como parâmetros de entrada do método `main`

› **Remova** o ficheiro `Exemplo.class` e compile o programa `Exemplo.java`

📄 Irá obter um erro de compilação. Confirme que o ficheiro `.class` não foi gerado.