



## Anexo 1 – Noções básicas de Java

Estes exemplos servem para introduzir alguns conceitos da linguagem *Java*. Para além das aulas teóricas, estes conceitos podem ser complementados pela bibliografia disponível na biblioteca, e por *tutorials* e exemplos que podem ser encontrados na Internet. Está também disponível um livro no hiperlink [http://www.ci.fct.unl.pt/arquivo/manuais/java/eckel\\_tjava11.pdf](http://www.ci.fct.unl.pt/arquivo/manuais/java/eckel_tjava11.pdf).

Nos exemplos dados a seguir, criar uma directoria “c:\users\my group name\trab2”, e editar os ficheiros a seguir apresentados.

### Exemplo 1 – O primeiro programa em Java

Ficheiro “Hello.Java”

```
public class Hello {
    static public void main(String args[]) {
        System.out.println("Hello world...");
    }
}
/*
compilar com: javac Hello.java
executar com: java Hello
* caso javac não funcione, compilar com "c:\jdk150\bin\javac Hello.java"
* neste caso, assumindo que o java está instalado na directoria c:\jdk150
* Outra possibilidade é actualizar a PATH do sistema.
*/
```

### Exemplo 2 – Utilização de argumentos na função “main”

Ficheiro “Hello\_args.Java”

```
public class Hello_args {
    static public void main(String args[]) {
        System.out.println("-----");
        for (int i=0;i<args.length; i++) {
            System.out.println("Hello "+args[i]+"...");
        }
        System.out.println("-----");
    }
}
/*
compilar com: javac Hello_args.java
executar com: java Hello_args manel maria jose antonia
*/
```



### Exemplo 3 – Utilização de funções matemáticas

A classe MathVector não constitui um programa. Esta classe é utilizada pela classe Matemática.

#### Ficheiro “MathVector.Java”

```
public class MathVector {
    private double x;
    private double y;
    private double z;
    /** Creates a new instance of Algebra */
    public MathVector(double _x, double _y, double _z) {
        x = _x;
        y = _y;
        z = _z;
    }
    public void somar(double x2, double y2, double z2){
        //argumentos simples
        x+= x2;
        y+= y2;
        z+= z2;
    }
    public MathVector scalarProduct(MathVector v) {
        MathVector result = new MathVector(0,0,0);
        result.x = x * v.x;
        result.y = y * v.y;
        result.z = z * v.z;
        return result;
    }
    public double module() {
        double module = Math.sqrt( Math.pow(x,2)+Math.pow(y,2)+Math.pow(z,2) );
        return module;
    }
    public String toString() {
        String s = new String(" "+x+"x + "+y+"y + "+z+"z");
        return s;
    }
}
```



#### Ficheiro “Matematica.Java”

```
public class Matematica {  
    static public void main(String args[]) {  
        MathVector v = new MathVector(2,3,5);  
        System.out.println("MathVector v =" +v);  
        System.out.println("Module of v =" +v.module() );  
        MathVector v2 = new MathVector(1,4,2);  
        MathVector v3 = v.scalarProduct(v2);  
        System.out.println("MathVector v3 =" +v3);  
        System.out.println("Module of v3 =" +v3.module() );  
    }  
}
```



## Exemplo 4 – Utilização de arrays

Ficheiro “Dados1.Java”

```
public class Dados1 {  
    public void printArray(Object array[]){  
        for(int i=0;i<array.length;i++)  
            System.out.print(array[i]);  
        System.out.println();  
    }  
  
    public void printMatrix(Object matrix[][] ) {  
        System.out.println("matrix={");  
        for(int i=0;i<matrix.length;i++) {  
            System.out.print(" ");  
            printArray( matrix[i] );  
        }  
        System.out.println("");  
        System.out.println("como exercicio fazer como que a matriz seja escrita com  
espacamento....");  
    }  
  
    static public void main(String args[]) {  
        String array1[]={ "what ", "is ", "the ", "matrix", "?"};  
        String array2[]={ "you ", "must ", "see ", "it ", "yourself", "..."};  
        String matrix[][] = {  
            { "a", "f", "1", "b"},  
            { "d", "5", "2", "8"},  
            { "1", "3", "9", "e"},  
            { "f", "a", "c", "a"}  
        };  
        Dados1 d = new Dados1();  
        d.printArray(array1);  
        d.printArray(array2);  
        d.printMatrix(matrix);  
    }  
}
```



## Exemplo 5 – Utilização de listas ligadas

Ficheiro “Dados2.Java”

```
public class Dados2 {  
    static public void main(String args[]) {  
        java.util.LinkedList list = new java.util.LinkedList();  
        list.add("aaa");list.add("ccc");list.add("222");list.add( new Integer(100) ); list.add("zzz");  
        System.out.println("-----");  
        System.out.println( "first(list)="+list.getFirst());  
        System.out.println( "last (list)="+list.getLast());  
        System.out.println("-----");  
        for(int i=0; i<list.size(); i++) {  
            System.out.println( "list["+i+"]="+list.get(i) );  
        }  
        System.out.println("-----");  
    }  
}
```