

# Ficha Prática 1

António Nestor Ribeiro, António Fernandes, F. Mário Martins  
{anr,af,fnm}@di.uminho.pt  
PPIV (LESI)

2003/04

## Objectivos

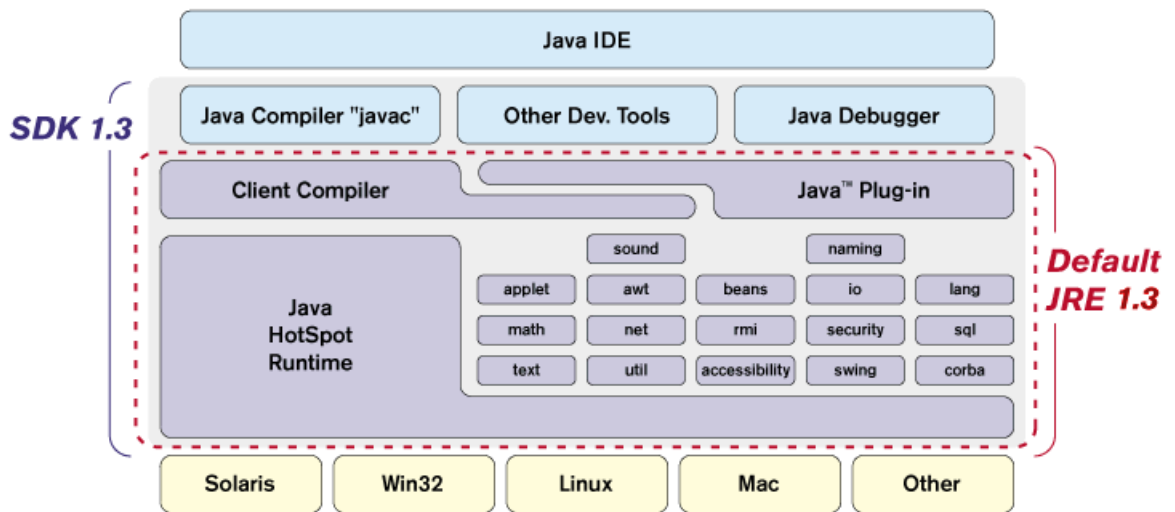
1. Apresentação do ambiente JAVA
2. Tipos de Dados e Estruturas de Controlo
3. Arrays: declaração e utilização
4. Exercícios com arrays

## Horário de Atendimento

- anr  
– 2ª feira - 14h-16h

## Introdução ao ambiente Java

### Java 2 quê?



## Execução de um programa Java

```
public class Teste1 {
    public static void main(String args[]) {
        // ...
        // código
        //...
    }
}
```

O ficheiro deve ter o mesmo nome da classe. Deve chamar-se `Teste1.java`.

**Compilação:** `javac Teste1.java`

Se correu bem, ie. não havia erros, gerou um ficheiro `Teste1.class`

**Execução:** `java Teste1`

## Tipos Primitivos

Tipo	Valor	val. p/ Om.	Tamanho	Gama
boolean	true, false	false	1	
char	caracteres	\u0000	16	\u0000 a \uFFFF
byte	inteiros	0	8	-128 a +127
short	inteiros	0	16	-32768 a +32767
int	inteiros	0	32	-2147483648 a +2147483647
long	inteiros	0	64	$\approx -1E+20$ a $+1E+20$
float	virg. flut.	0.0	32	$\approx \pm 3.4E+38$ a $\pm 1.4E-45$
double	virg. flut.	0.0	64	$\approx \pm 1.8E+308$ a $\pm 5E-324$

## Variáveis

*tipo variável*[= valor];

## Constantes

**final** *tipo variável* = valor;

## Alguns Operadores

Prioridade	Operação	Tipo dos Operandos	Associatividade	Descrição
1	++	aritméticos	D	pré/pós incremento
	--	aritméticos	D	pré/pós decremento
	+, -	aritméticos	D	sinal unário
	!	boolean	D	negação
	(tipo)	todos	D	conversão (cast)
2	*	aritméticos	E	multiplicação
	/	aritméticos	E	divisão
	%	aritméticos	E	resto
3	+	aritméticos	E	soma
	-	aritméticos	E	subtração
5	<, <=	aritméticos	E	menor (ou igual)
	>, >=	aritméticos	E	maior (ou igual)
6	==	primitivos	E	iguais
	!=	primitivos	E	diferentes
7	&	arit., char	E	e (bit a bit)
	&	boolean	E	e (booleano)
9		arit., char	E	ou (bit a bit)
		boolean	E	ou (booleano)
10	&&	boolean	E	e (condicional)
11		boolean	E	ou (condicional)

## Estruturas de Controlo

### Condicionais:

*if/else*  
*switch/case*

### Repetitivas:

*while*  
*do/while*  
*for*

### Outras:

*a ver mais tarde...*

## Arrays

Declaração:

```
int lista_alunos[]; // declaração ‘à la C’
int[] lista_alunos;
int lista_alunos[] = new int[35];
int[] lista_alunos = new int[35];
int pequena_lista[] = {1, 2, 3, 4, 5};
short listas[][] = new short[5][35];
short listas[][] = new short[5][];
listas[0] = new short[25];
listas[4] = new short[28];
```

Tamanho de um array:

```
lista_alunos.length
```

## *Output*

```
System.out.println("Viva o Java!");
```

## Primeiro programa

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

        System.out.println("Viva o Java!")
    }
}
```

## Operações comuns com arrays

1. Imprimir os elementos de um array

```
// declarar, inicializar e imprimir os elementos de um array //

int[] lista = { 12, 2, 45, 66, 7, 23, 99};
System.out.println("----- ELEMENTOS DO ARRAY -----");
for(int i=0; i < lista.length; i++) {
    System.out.println("Elemento "+ i + " = " + lista[i]);
}
System.out.println();
```

```
System.out.println("-----")
System.out.println();
```

## 2. Mínimo de um array - método I

```
int min = Integer.MAX_VALUE; /* inicializado com o MAIOR INTEIRO */
int pos = -1;                /* indice do MENOR; é -1 se array vazio */

for(int i=0; i < lista.length; i++) {
    if (lista[i] < min) {
        min = lista[i]; pos = i;
    }
}

if (pos == -1)
    System.out.println("O ARRAY ESTÁ VAZIO !!");
else {
    System.out.println();
    System.out.println("Menor Elemento = " + min + " no indice " + pos);
}
```

## 3. Mínimo de um array - método II

```
min = lista[0]; /* inicializado com o primeiro da lista */
pos = 0;        /* o 1º menor está de momento no indice 0 !! */
for(int i=1; i < lista.length; i++) {
    if (lista[i] < min) {
        min = lista[i]; pos = i;
    }
}

System.out.println();
System.out.println("Menor Elemento = " + min + " no indice " + pos);
```

## Exercícios

1. (\*) Considere o seguinte código Java:

```
int i=5, j, k;

j = --i;
k = j++;
i += 10;
```

Depois da execução do código, qual será o valor das variáveis *i*, *j* e *k*?

2. (\*) Considere a seguinte declaração de um *array* de inteiros:

```
int[] lista = {5, 2, 10, 15, 7, 23, 14, 5}
```

Escreva código Java que lhe permita:

- (a) Imprimir todos os elementos de *lista*.
  - (b) Calcular qual o índice do primeiro 7 em *lista*.
  - (c) Calcular o somatório de *lista*.
  - (d) Calcular o máximo de *lista*.
3. Considere agora o seguinte código Java:

```
int[] a1={9,4,2,7,1,5,8}, a2;

a2=a1;
for(int i=0;i<a1.length;i++)
    a1[i] += 10;
```

- (a) Depois da execução do código, qual será o valor dos elementos dos *arrays* *a1* e *a2*? E qual o valor de *i*?
- (b) Escreva código Java para:
  - i. Copiar *a1* para *a2*.
  - ii. Comparar *a1* com *a2*.
  - iii. Copiar parte de *a1* (entre os índices *j* e *k*) para um outro *array* *a3*.

4. Considere novamente um *array* de inteiros chamado *lista*. Escreva código Java que lhe permita:

- (a) Ordenar *lista* por ordem crescente.
- (b) Remover o elemento na posição *k* do *array*.
- (c) Implementar um algoritmo de procura binária sobre *lista*.
- (d) Inverter *lista*.

5. (\*) Considere as declarações:

```
short[] [] a1 = {{1,2,3},{4,5,6},{7,8,9}};  
short[] [] a2 = {{1,2,3},{4,5,6},{7,8,9}};  
short[] [] a3 = new short[3][3];
```

Escreva código Java para:

- (a) Calcular o somatório de *a1*.
- (b) Somar *a1* e *a2* para *a3*.
- (c) Multiplicar *a1* e *a2* para *a3*.

6. Considere a declaração:

```
int[] [] turmas = {{.....},  
                  {.....},  
                  {..}  
                  };
```

correspondente a um *array* bidimensional em que cada linha contém os números dos alunos de um dado turno prático de PPIV. Escreva um programa Java que lhe permita:

- (a) Calcular o número total de alunos inscritos.
- (b) Calcular qual a turma mais pequena.
- (c) Calcular o número do aluno mais antigo.
- (d) Ordenar as turmas por ordem crescente de tamanho.