

Curso Profissional: Programação de Informática**PSD – 11.º ano: UFCD 0816 - Programação de sistemas distribuídos - JAVA****Ficha de Trabalho 7****Ano letivo 22/23****arrays**

Para utilizar um array é necessário 2 passos:

- (1) Declaração de uma referência para o array:

```
tipoDeDados [] Nomearray;    // modo Java - melhor
```

ou

```
tipoDeDados Nomearray [];    // funciona é como na linguagem C - não é o melhor
```

- (2) Armazenar o array (espaço em memória):

```
Nomearray = new tipoDeDados [DimArray];
```

(1) e (2) podem ser usados numa única instrução:

```
tipoDeDados[] Nomearray = new tipoDeDados[DimArray];
```

Exemplo:

```
float[] numeros = new float[5];
```

```
numeros[0] = 234.2;    // primeira posição de um array é 0 (índice 0)
```

- Podemos inicializar os valores de um array na sua declaração:

```
double[] minhaLista = {1.9, 2.9, 3.4, 3.5};
```

//a propriedade length dá-nos a dimensão do array

```
for (int i = 0; i < minhaLista.length; i++)  
    System.out.println(minhaLista[i]);
```

// Escreve todos os elementos do array, com for each <-novidade

```
/* Sintaxe: for (type var: array){  
    Instruções que utilizem a variável var  
} */
```

```
for (double elemento: minhaLista)  
{  
    System.out.println(elemento);  
}
```

```
// criar um array de objetos de uma classe
```

```
Public class Conta
```

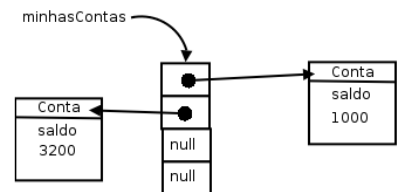
```
{
    float saldo;
    public static void main(String args[])
    {
```

```
        Conta[] minhasContas;
        minhasContas = new Conta[4]; // criação de 4 espaços em memória para guardar a referência
                                         a uma conta, i.é. criação do array de classes
```

```
        minhasContas[0] = new Conta(); // criação de cada objeto do array de classes
        minhasContas[1] = new Conta();
        minhasContas[0].saldo = 1000.0;
        minhasContas[1].saldo = 3200.0;
```

```
        System.out.println(minhasContas[0].saldo+" " +minhasContas[1].saldo);
```

```
    }
}
```



```
// atributo de uma classe que é um array de objetos de outra classe(empregados)
```

```
public class Empresa
```

```
{
    Funcionario[] empregados;
    String cnpj;
}
```

```
public class Funcionario {
```

```
    string nome;
    float salario;
}
```

```
public class TestaEmpresa {
```

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

```
    {
        Empresa emp = new Empresa(); //objeto da classe empresa
```

```
        emp.empregados = new Funcionario[2]; // Armazenar o array
```

```
        emp.cnpj="AMonte S.A.";
```

```
        emp.empregados[0] = new Funcionario(); //criação do 1.º objeto da classe funcionario
        emp.empregados[1] = new Funcionario(); //criação do 2.º objeto da classe funcionario
```

```
        emp.empregados[0].salario = 1000;
        emp.empregados[0].nome= "Pedro Dias";
        emp.empregados[1].nome= "Duarte Lima";
        emp.empregados[1].salario = 1700;
```

```
        System.out.println("Designação da empresa: "+emp.cnpj);
```

```
        for (int i=0;i<2;i++)
```

```
            System.out.println(emp.empregados[i].nome+" "+emp.empregados[i].salario);
```

```
        }
```

```
    }
```

1. Elabora um programa que escreva no ecrã os números contidos num array de inteiros. Inicializa o array com os seguintes valores: 2, 4, 8, 10, 15.
2. Altera o programa anterior para solicitar ao utilizador 15 números, guardar esses números no array e imprimi-los no ecrã.

3. Acrescenta código ao programa anterior para indicar, no final, quantos números deste array são positivos e quantos são negativos.

Aspeto do Output desejado:

```
Insira o 1º número:15
Insira o 2º número:2
Insira o 3º número:5
...
Insira o 15º número: 10

Os números inseridos são: 15, -2, -5, ..., 10
Números positivos: 13
Números negativos: 2
```

4. Reformula o código ao programa anterior, para indicar, no final, quantos números deste array são pares e quantos são ímpares.
5. Escreve um programa que dadas 5 notas de alunos (1 a 20 valores), indique qual a nota mais alta, qual a mais baixa e a média.

Faz a validação para que sejam apenas aceites notas de 1 a 20 e inteiros voltar a pedir os valores até corresponderem aos válidos.

```
int i = Int.parse(valor);
} catch (NumberFormatException e) {
    // Se entrar aqui o valor é inválido.
}
```

6. Num programa, declara um array e inicializa-o com 10 valores inteiros, escolhidos por ti. Imprime o array por ordem crescente, sabendo que a classe Array contém o método sort, que permite ordenar por ordem crescente os valores de um array. Para isso é necessário importar essa classe: `import java.util.Arrays;`

E chamar o método: `Arrays.sort(numeros)`

Imprime também a posição de um dos valores do array, usando o método `binarySearch` (consulta a ficha 6).

7. Cria uma classe de nome Aluno com os atributos seguintes: Numero, nome, notas, em que notas é um array de 6 valores reais.

Cria uma classe de nome **MostraDados** que contenha o método main e que permita ler os dados relativos a um objeto da classe alunos e imprimir essa informação no ecrã.

A impressão das notas é feita dentro de um método, da classe **Aluno**, de nome **ImprimeNotas** que recebe, como parâmetro, o array. (continua→)

Cria ainda um método chamado **AlteraNota**, na classe **Aluno**, que permita alterar qualquer uma das notas do objeto aluno. A nota e a posição da nota no array terá que ser pedida ao utilizador dentro do método **AlteraNota**.

No método main, da classe **MostraDados**, após a impressão das notas no ecrã, deve ser alterada uma das notas e voltar a imprimir as notas, após efetuada a alteração.

Pacote java.time

9. Cria um novo programa (classe **Datas**), com método main, e acrescenta-lhe o seguinte método:

```
private static void mostra_data() {  
  
public class Datas {  
    private static void mostra_data() {  
        Datas data = new Datas();  
        LocalDate hoje = LocalDate.now();  
  
        int ano = hoje.getYear();  
        int mes = hoje.getMonthValue();  
        int dia = hoje.getDayOfMonth();  
        System.out.println(dia + "-" + mes + "-" + ano);  
  
        String nome_mes = hoje.getMonth().name();  
  
        String nome_mes_port=hoje.getMonth().getDisplayName(TextStyle.FULL, new  
        Locale("pt"));  
  
        System.out.println(nome_mes_port);  
  
        System.out.println(dia + "th " + nome_mes + " " + ano+"\n");  
    }  
}
```

2. A introdução do código anterior implica a importação das Classes utilizadas, que podem ser obtidas carregando com o botão do lado esquerdo do rato, na lâmpada que aparece na coluna do lado esquerdo. Nomeadamente:

```
import java.time.LocalDate;  
import java.time.format.TextStyle;  
import java.util.Locale;
```

3. Acrescenta o seguinte código no main(), para chamar o método criado:

```
mostra_data();
```

4. Executa o teu programa.

Bibliografia:

<https://www.w3schools.com/java>
<https://www.tutorialspoint.com/java/>
Jesus, C. (2013). Curso Prático de Java. Lisboa: FCA
Coelho, P (2016). Programação em JAVA – Curso Completo. Lisboa: FCA
https://www.tutorialspoint.com/java/util/java_util_arrays.htm
<https://www.codota.com/code/java/classes/java.time.ZonedDateTime>
<https://mkyong.com/java8/java-8-zoneddatetime-examples/>