

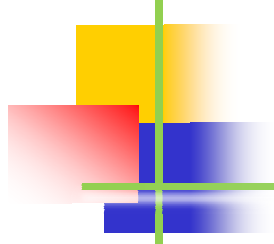
deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

# Programação 1

Departamento de Electrónica, Telecomunicações e Informática  
Universidade de Aveiro

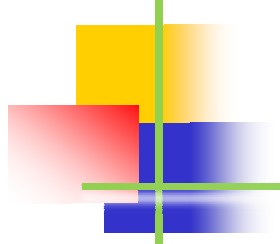
<http://moodle.ua.pt/>



- Arrays de Strings
- Arrays de registos (classes)
- Arrays bidimensionais
- Exemplos



- Vimos até aqui que podíamos armazenar numa variável do tipo array vários valores do mesmo tipo.
- Podemos também ter arrays de tipos referência, sendo que numa variável deste tipo podemos armazenar várias referências de um certo tipos de dados:
  - arrays de Strings;
  - arrays de registos;
  - arrays bidimensionais (arrays de arrays).
- A criação deste tipo de arrays necessita de uma declaração semelhante à dos arrays de tipos primitivos, mas cada elemento dos array necessita depois de ser “criado” segundo as regras correspondentes ao tipo de dados em causa.



# Arrays de Strings



deti

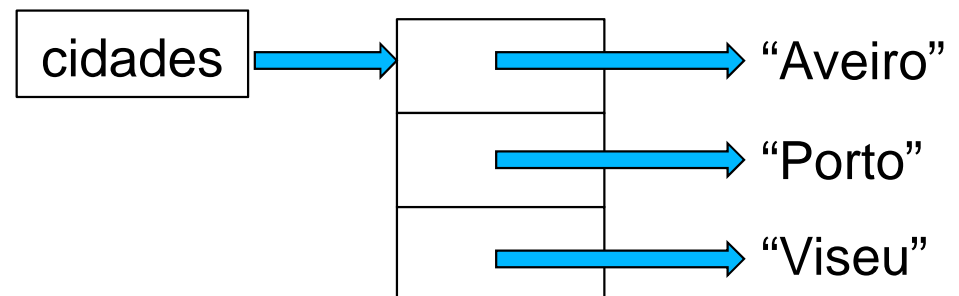
universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

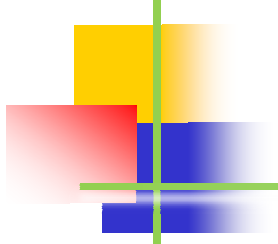
- É então possível criar uma sequência de Strings, ou seja, uma estrutura bidimensional de caracteres.
- A declaração de uma sequência de Strings cria um array de referências nulas para String que depois serão preenchidas por instruções de atribuição...

```
String cidades[];  
cidades = new String[3];  
cidades[0] = "Aveiro";  
cidades[1] = "Porto";  
cidades[2] = "Viseu";
```

// ou

```
String cidades[] = {"Aveiro", "Porto", "Viseu"};
```





# Exemplo

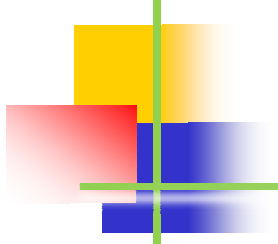


deti

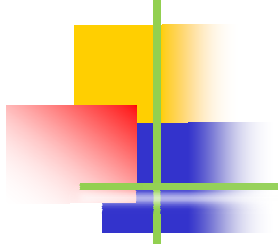
universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
// ler frases até aparecer a palavra fim
public static int lerFrases(String frases[]){
    String s = new String(); int n = 0;
    do{
        System.out.print("Frase: "); s = sc.nextLine();
        if(!s.equalsIgnoreCase("fim")){
            frases[n] = s; n++;
        }
    }while(!s.equalsIgnoreCase("fim") && n < frases.length);
    return n;
}

public static void imprimirFrases(String frases[], int n){
    for(int i = 0 ; i < n ; i++)
        System.out.printf("[%d] -> %s\n", i, frases[i]);
}
```



- Uma maneira de armazenar informação em aplicações reais consiste na utilização de sequências de registos, normalmente designadas por bases de dados.
- A declaração de arrays de registos é em em tudo semelhante à das sequências de tipos primitivos ou Strings, com a exceção que tem de ser decomposta em duas operações:
  - a primeira consiste em criar a sequência de referências para os futuros elementos do tipo registo;
  - a segunda consiste em criar os elementos propriamente ditos, seguindo a regra para a criação de variáveis do tipo registo.



# Exemplo (1)

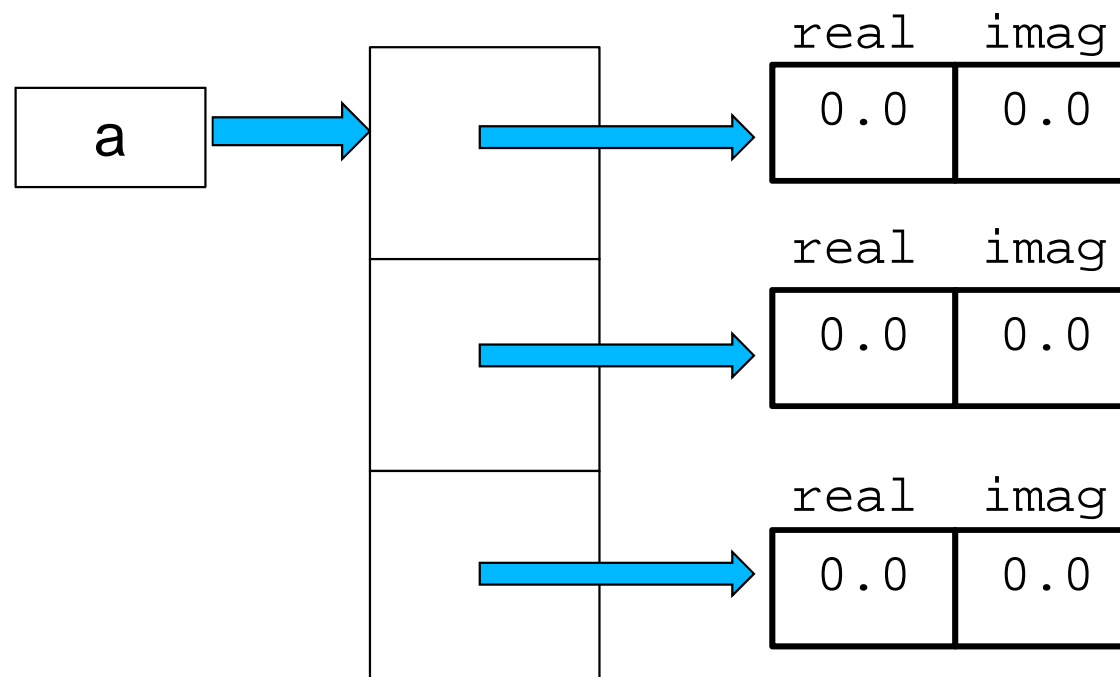
// Declaração de um array de números complexos

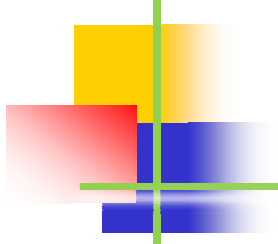
```
Complexo a[] = new Complexo[3]; // Declaração do array
```

```
a[0] = new Complexo(); // Alocação de espaço para pos. 0
```

```
a[1] = new Complexo(); // Alocação de espaço para pos. 1
```

```
a[2] = new Complexo(); // Alocação de espaço para pos. 2
```





## Exemplo (2)

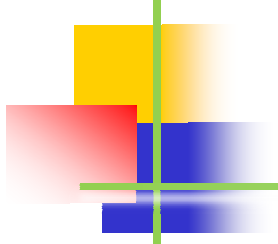


deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
// leitura de pontos até aparecer o (0, 0)
... main ...{
    Ponto2D pontos[] = new Ponto2D[10];
    Ponto2D p; int n = 0;
    do{
        System.out.println("Introduza um ponto:");
        p = lerPonto2D(); // aqui é criada uma nova referência
        if(p.x != 0 || p.y != 0){
            pontos[n] = p; // que depois é armazenada no array
            n++;
        }
    }while((p.x != 0 || p.y != 0) && n < pontos.length);
    imprimePontos(pontos, n);
}
```





## Exemplo (3)



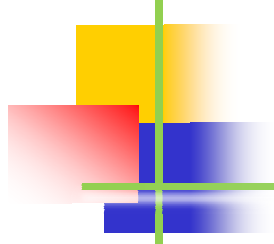
deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
public static Ponto2D lerPonto2D(){
    Ponto2D tmp = new Ponto2D();
    System.out.print("Coordenada x: ");
    tmp.x = sc.nextDouble();
    System.out.print("Coordenada y: ");
    tmp.y = sc.nextDouble();
    return tmp;}

public static void imprimePontos(Ponto2D a[], int n){
    for(int i = 0 ; i < n ; i++){
        System.out.printf("pto %d:(%.1f, %.1f)\n",
            i, a[i].x, a[i].y);
    }
}

class Ponto2D{
    double x, y;}
```



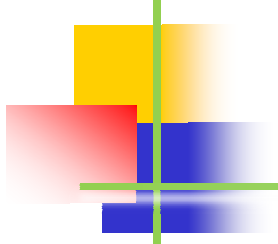
# Arrays bidimensionais



deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

- Existem problemas em que a informação a ser processada é melhor representada através de uma estrutura de dados com um formato bidimensional, como por exemplo uma matriz.
- Uma sequência bidimensional é na prática uma sequência de sequências.
- A sua declaração respeita as regras de uma sequência unidimensional, sendo a única diferença o facto de usar dois operadores sequência seguidos `[ ] [ ]`.
- Pode ser vista como uma estrutura matricial de elementos, composta por linhas e colunas, sendo o acesso a cada um dos elementos feito através de dois índices.



# Exemplo (1)



deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
// Declaração de uma matriz de valores inteiros com 3  
linhas e 3 colunas
```

```
int m[][] = new int[3][3];
```

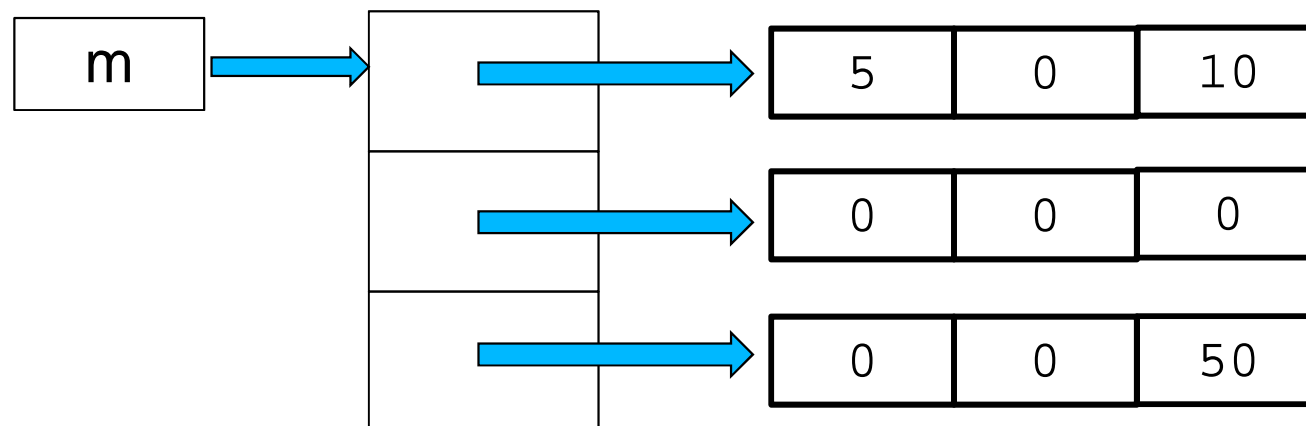
```
m[0][0] = 5; // elemento na linha 0, coluna 0
```

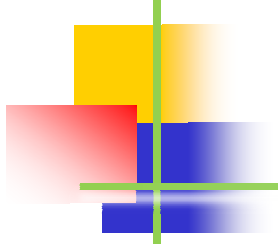
```
m[0][2] = 10; // elemento na linha 0, coluna 2
```

```
m[2][2] = 50; // elemento na linha 2, coluna 2
```

```
m.length // número de linhas
```

```
m[0].length // número de colunas para a linha 0
```





## Exemplo (2)



deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
public static void lerMatriz(int m[][]) {
    for(int l = 0 ; l < m.length ; l++){
        for(int c = 0 ; c < m[l].length ; c++){
            System.out.print("pos [" + l + "][" + c + "]: ");
            m[l][c] = sc.nextInt();
        }
    }
}

public static void imprimirMatriz(int m[][]) {
    for(int l = 0 ; l < m.length ; l++){
        for(int c = 0 ; c < m[l].length ; c++){
            System.out.printf("  %5d", m[l][c]);
        }
        System.out.println();
    }
}
```