

## FICHA 2 – TABELAS

As tabelas ou *arrays* são objetos constituídos por uma sequência de dados do mesmo tipo (int, char, float, double, etc.).

1. Para criar uma tabela faz-se:

```
int[] numeros = new int [10];
```

ou:

```
int[] numeros;           // definição
numeros = new int[10];    // criação
```

2. Os elementos de uma tabela são identificados pelo nome da tabela e por um índice que indica a sua posição na tabela. `numeros[i]` representa o elemento na posição `i`. O primeiro elemento da tabela tem o índice 0.

3. A criação de uma tabela e a inicialização dos seus elementos também pode ser feita em simultâneo:

```
int[] tab = {1, 3, 5, 7, 9};
```

4. A dimensão máxima da tabela é dada pelo campo `length`. Após a sua criação não é possível alterar o valor deste campo.

5. É possível passar a tabela como argumento de um método. O que é passado é uma referência para a tabela, e o seu valor pode vir alterado após a execução do método. Isto é, não é necessário devolver uma tabela que se queira alterar, e caso não se pretenda que as alterações no método alterem a tabela original, deve-se criar uma cópia primeiro (ver ponto 9).

6. A criação de uma tabela bidimensional é feita de forma semelhante:

```
double[][] x = new double[5][5];
```

7. Os elementos da tabela são identificados pelo seu nome e por dois índices que indicam a sua posição na tabela. `x[i][j]` representa o elemento da tabela `x` que se localiza na linha `i` coluna `j`. Os índices de linha e de coluna começam em zero.

8. Numa tabela bidimensional regular o campo `length` indica o número de linhas e `x[0].length` o número de colunas.

9. A cópia de tabelas é mais eficientemente realizada usando o método:

```
public static void System.arraycopy(Object src, int
                                   srcPosition, Object
                                   dst, int dstPosition,
                                   int length)
```

que copia uma tabela `src`, a partir da posição `srcPosition`, para outro `dst`, a partir da posição `dstPosition`. O número de elementos copiados é dado por `length`.

10. Uma tabela bidimensional irregular pode ser criada e inicializada através de:

```
int[][] irregular = {{1, 2, 3}, {1, 4}, {1, 6, 8, 4}};
```

## Exercícios

### 1. Tabela de números aleatórios

Escreva um programa que crie uma tabela com 10 números inteiros aleatórios entre 1 e 100. De seguida deverá imprimir no ecrã todos os números aleatórios gerados (indicando a posição de cada número na tabela) e apresentar a média destes números. Utilize o método ***Math.random()*** para gerar números aleatórios no intervalo [0,1).

### 2. Junta tabelas

Crie um método que receba duas tabelas de  $n$  números inteiros (cada) e que devolva uma tabela com base na informação contida nas duas tabelas originais. A nova tabela é construída através da intercalação dos elementos das duas primeiras.

Exemplo:

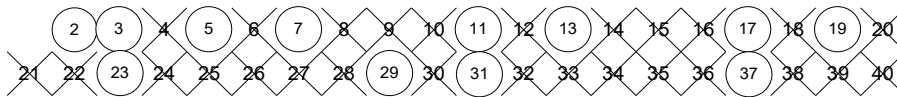
$$\begin{array}{|c|c|c|} \hline 1 & 4 & 7 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|} \hline 9 & 2 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline 1 & 9 & 4 & 2 & 7 & 3 \\ \hline \end{array}$$

### 3. Produto de matrizes

Crie um método que receba duas matrizes retangulares e que devolva o seu produto. Quando não for possível efetuar a multiplicação deve apresentar uma mensagem elucidativa.

### 4. Peneira de Eratosthenes

Uma forma simples e eficiente de calcular todos os números primos até um certo valor  $n$  é o método da Peneira de Eratosthenes. O processo é simples: escrevem-se todos os valores entre 2 e  $n$  (limite máximo). Em seguida faz-se um círculo em volta do 2, marcando como primo e riscam-se todos os seus múltiplos. Continua-se a fazer um círculo em volta do menor inteiro que encontrar, eliminando todos os seus múltiplos. Quando não restarem números sem terem círculos à volta ou traços por cima, os números com círculos à volta representam todos os primos até  $n$ . A figura seguinte apresenta o método para  $n=40$ .



- Crie um método que implemente a Peneira de Eratosthenes. Este método deve ter como parâmetro de entrada um número inteiro correspondente ao limite máximo e deve devolver uma tabela contendo todos os números primos encontrados.
- Desenvolva um programa que, dados  $x$  números inteiros inseridos pelo utilizador, identifique todos os números primos (o valor de  $x$  deve ser pedido ao utilizador). Na elaboração do programa deve utilizar o método desenvolvido na alínea anterior.

### 5. Preenche a matriz

Pretende-se que desenvolva um programa que crie uma tabela bidimensional com 5 linhas e 10 colunas com as seguintes características:

- Preencha as posições da tabela com números aleatório e imprima o conteúdo no ecrã.

```
93 17 62 18 26 99 41 40 2 54
20 87 65 99 74 91 77 64 24 48
0 39 57 8 89 5 2 67 44 50
87 66 94 8 68 41 31 49 41 87
95 39 20 63 39 82 29 34 58 19
```

- Preencha as posições da tabela com números sequenciais a incrementar coluna a coluna.

```
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
```

- Preencha as posições da tabela com números sequenciais a incrementar linha a linha.

```
0 5 10 15 20 25 30 35 40 45
1 6 11 16 21 26 31 36 41 46
2 7 12 17 22 27 32 37 42 47
3 8 13 18 23 28 33 38 43 48
4 9 14 19 24 29 34 39 44 49
```

### 6. Caça aos números repetidos

Pretende-se que desenvolva um programa capaz de simular uma versão simplificada do jogo “Sudoku”. O objetivo do jogo é preencher as células vazias de uma matriz 9x9 com números de 1 a 9 de modo que em cada linha e em cada coluna não haja números repetidos. Assume-se que as células vazias estão preenchidas com o número 0.

Crie um método que, recebendo uma tabela de sudoku como descrito acima, e as coordenadas x, y de uma posição, crie e devolva um vetor contendo os números que nesse momento podem ocupar essa posição da tabela, ou seja todos os que não pertencem à linha e à coluna do elemento em causa.

Nota: Não necessita de fazer o código para gerar a tabela de sudoku, mas assuma que passa como argumento no método.

Exemplo:

9	4	0	1	0	2	0	5	8
6	X	0	0	5	0	0	0	4
0	0	2	4	0	3	1	0	0
0	2	0	0	0	0	0	6	0
5	0	8	0	2	0	4	0	1
0	6	0	0	0	0	0	8	0
0	0	1	6	0	8	7	0	0
7	0	0	0	4	0	0	0	3
4	3	0	5	0	9	0	1	2

Vetor resultante da chamada do método relativo à posição vazia marcada com um X na tabela original:

1 7 8 9

### 7. Quebra-cabeças

Considere que quer resolver um quebra-cabeças. Existe um retângulo com um determinado número de linhas e um determinado número de colunas preenchido com letras. A figura mostra um exemplo de um quebra-cabeças com 5 linhas e 6 colunas:

E	b	a	u	l	q
L	e	r	r	s	s
u	w	u	q	g	r
a	a	l	l	u	a
p	m	h	u	d	j

Desenvolva um método que procure todas as ocorrências de determinada palavra no quebra-cabeças. A palavra pode ocorrer numa linha ou numa coluna. De cada vez que o método encontrar uma ocorrência da palavra, deve escrever no monitor o número da linha e da coluna em que a palavra tem início. O método deve receber como argumentos a tabela

e a palavra a pesquisar. Considerando o exemplo da figura, se a palavra a pesquisar for lua o método deveria escrever:

A palavra lua surge:

- Ao longo da coluna 1 com início na linha 2
- Ao longo da linha 4 com início na coluna 4

### 8. Média dos seguintes

Desenvolva um programa que crie uma matriz de  $N \times N$  e que a preencha com valores inteiros aleatórios, entre 0 e 100. Depois de apresentar a matriz inicial ao utilizador, deve percorrê-la, linha a linha, e substituir cada elemento pela média (parte inteira) de todos os elementos seguintes da matriz. O último elemento,  $M\{N,N\}$ , ficará com o valor ZERO.

Exemplo:

$N=3$

Matriz inicial

2	19	45
16	33	21
76	55	3

Matriz Final:

$$M\{1,1\} = (19+45+16+33+21+76+55+3)/8 = 33$$

$$M\{1,2\} = (45+16+33+21+76+55+3)/7 = 35$$

$$M\{1,3\} = (16+33+21+76+55+3)/6 = 34$$

$$M\{2,1\} = (33+21+76+55+3)/5 = 37$$

$$M\{2,2\} = (21+76+55+3)/4 = 38$$

$$M\{2,3\} = (76+55+3)/3 = 44$$

$$M\{3,1\} = (55+3)/2 = 29$$

$$M\{3,2\} = (3)/1 = 3$$

$$M\{3,3\} = 0$$

33	35	34
37	38	44
29	3	0

### 9. Pares

Crie um método que recebe uma tabela bi-dimensional regular de número inteiros e que devolve uma tabela irregular com os números pares da tabela inicial. Exemplo:

Tabela inicial

2	19	45
16	18	21
76	15	6

Resultado:

2	
16	1
76	6