

Ficha de trabalho #3

Arrays Funções

Arrays

Array: estrutura de dados, com uma ou mais dimensões (matriz: array 2D), usada para guardar dados do mesmo tipo.

Definir array com uma dimensão: `tipo[] nome`
"tipo" pode ser qualquer um dos tipos em Java.

É necessário definir o número de dimensões e o tamanho das dimensões:

```
int[] array1 = new int[4];
```

Define um array de inteiros com 1 dimensão e 4 posições

array1:

0	0	0	0
---	---	---	---

```
double[] notas;
```

Define um array de doubles com 1 dimensão e 3 posições. Como não definimos tamanho, fica a null.

notas:



```
notas = new double[3];
```

Definir o tamanho do array anterior: 3 posições

notas:

0.0	0.0	0.0
-----	-----	-----

```
char[] vogais = {'a','e','i','o','u'};
```

Define um array de caracteres com 1 dimensão e inicializa-o com 5 valores.

vogais:

a	e	i	o	u
---	---	---	---	---

```
String[][] matriz1 = new String [2][3];
```

Define um array de Strings com duas dimensões: 2 linhas e 3 colunas.

frutas:

null	null	null
null	null	null

```
boolean[][] m = {{false,true},{true,false}};
```

Define um array de booleans com 2 dimensões e inicializa-o com valores.

m:

false	true
true	false

Aceder ao conteúdo de um array:

```
double[] abc = {1.0, 12.4, 3.5, 0.3};
```

abc:

1.0	12.4	3.5	0.3
-----	------	-----	-----

```
int x = abc[0];
```

x fica com o valor guardado na posição 0 do array.

abc:

1.0	12.4	3.5	0.3
-----	------	-----	-----

```
abc[2] = 7.6
```

O valor guardado na posição 2 do array passa a ser 7.6.

abc:

1.0	12.4	7.6	0.3
-----	------	-----	-----

```
for(int i=0; i<abc.length; i++){
    abc[i] += 1.0;
}
```

Percorre o array e soma 1.0 ao valor guardado em cada posição.

abc:

2.0	13.4	8.6	1.3
-----	------	-----	-----

Funções

- Conjuntos de instruções para executar uma única ação
- Permitem modularidade e reutilização de código

Sintaxe

```
modifs tipo nome(tipo1 arg1, ...){
    instrução 1;
    instrução 2;
    ...
}
```

Exemplos

```
public static int soma(int a, int b){
    return a + b;
}
```

```
public static void mostra (String s, int n){
    System.out.println(s + n);
}
```

Componentes de uma função:

- Modificadores: public / private; static
- Tipo: void, um dos tipos do java (int, double, String, ...), ou outros (veremos mais tarde)
- Nome: um nome explicativo do que faz a função, iniciado com letra minúscula
- Conjunto de 0 ou mais argumentos
- Instruções: os passos que a função deve executar

O main é uma função

Podemos chamar funções a partir do main ou a partir de outras funções.

```
public static int soma(int a, int b){
    return a + b;
}

public static void mostra (String frase, int numero){
    System.out.println(frase + numero);
}

public static void main(String[] args){
    int soma = soma(3, 2);
    String frase = "O resultado da soma é ";
    mostra(frase, soma);
}
```

Também podemos chamar uma função dentro de si própria (função recursiva).

```
// Função que soma todos os n°s de 1 até um dado valor
public static int somaDe1Ate(int x){
    if(x == 1)
        return 1;
    else
        return x + somaDe1Ate(x-1);
}
```

Arrays

- Escreva um programa que calcule e armazene num array os primeiros 30 números da sequência de Fibonacci. De seguida mostre a sequência de números no ecrã. Os números de Fibonacci são os números na sequência de Fibonacci, caracterizada pelo fato de que cada número após os dois primeiros é a soma dos dois precedentes.

Output
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040

- Considere um prédio de apartamentos com 3 entradas e 4 pisos completamente ocupados. Escreva um programa que leia o nome do proprietário de cada um dos apartamentos e permita saber, para um dado nome inserido pelo utilizador, qual é a respetiva entrada e piso. Caso não exista imprimir a mensagem “Não existe”.

Input	Output
Ana Bruno Carla Diogo Elsa Filipe Guida Hugo Inês João Lara Manuel Diogo	Andar 1, entrada 0
Ana Bruno Carla Diogo Elsa Filipe Guida Hugo Inês João Lara Manuel Xavier	Não existe

3	João	Lara	Manuel
2	Guida	Hugo	Inês
1	Diogo	Elsa	Filipe
0	Ana	Bruno	Carla
	0	1	2

- Escreva um programa que leia as classificações de exame de uma turma com n alunos (n introduzido pelo utilizador) para um array e calcule e mostre a média da turma e quantos alunos reprovaram à disciplina.

Input	Output
5 12 9 8 12 15	Média: 11.2. Reprovaram 2

- Escreva um programa que leia para um array um conjunto de n valores inteiros (n introduzido pelo utilizador) e determine qual é o maior elemento e menor elemento do conjunto.

Input	Output
5 12 9 8 12 15	Maior: 15 Menor: 8

5. Escreva um programa que leia uma sequência de n números inteiros (n introduzido pelo utilizador) e os mostre pela ordem inversa da inserção.

Input	Output
5 12 9 8 12 15	15 12 8 9 12

6. Escreva um programa que leia as classificações de um teste de avaliação resolvido por n alunos e calcule as frequências absolutas para as notas obtidas. Considere que as classificações variam entre zero (0) e vinte (20) inclusive. O valor de n é fornecido pelo utilizador. Utilize um array para guardar as frequências das notas, de zero a vinte.

Input	Output
8 12 13 10 5 12 10 12 10	5: 1, 10: 3, 12: 3, 13: 1

7. Escreva um programa que preencha um array bidimensional com números inteiros inseridos pelo utilizador. O utilizador deve, previamente, definir as dimensões do array (n° linhas e n° colunas). De seguida, devem ser colocados em dois novos arrays (unidimensionais) a soma de cada uma das colunas e a soma de cada uma das linhas. Por fim, os arrays devem ser visualizados. Exemplo:

Input	Output
3 4 1 2 3 4 5 6 7 8 9 10 11 12	arr: 1 2 3 4 5 6 7 8 9 10 11 12 arrSomaLinhas: 10 26 42 arrSomaColunas: 15 18 21 24

8. Escreva um programa que leia um valor n seguido de n inteiros inseridos pelo utilizador. De seguida imprima os números devidamente ordenados de forma crescente. Sugestão: utilizar o algoritmo de ordenação BubbleSort (<https://www.geeksforgeeks.org/bubble-sort/>).

Input	Output
8 12 13 10 5 12 10 12 10	5 10 10 10 12 12 12 13

9. Escreva um programa que mostre o top três (3) dos melhores salários dos funcionários de uma empresa. Sugestão: Ler os salários para um array, ordenar a informação de forma decrescente e escolher os três primeiros valores (salários mais elevados).

Input	Output
8 1200 1300 1080 500 1200 1000 1250 1050	1300 1250 1200

Funções

10. Escreva um programa para calcular o fatorial de um número n , usando uma função que retorna o fatorial do número passado por parâmetro:

$$f! = f \times (f - 1) \times (f - 2) \times \dots \times 2 \times 1$$

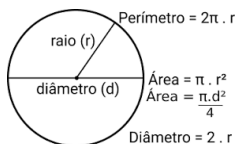
n	fatorial
6	720
5	120

- Usando uma função simples
- Usando uma função recursiva

11. Escreva um programa para testar se um conjunto de palavras são palíndromos, utilizando uma função que recebe uma palavra e determina se esta é um palíndromo (sequências de caracteres que se leem da mesma forma da esquerda para a direita e da direita para a esquerda).

Palavra	É palíndromo?
ana	true
AA	true
Ze	false

12. Escreva um programa para visualizar a área de uma circunferência, usando uma função que, dado o raio, retorna a área da circunferência.



Raio	Área
1	3.141592653589793
2	12.566370614359172

13. Escreva um programa para calcular as combinações, usando uma função que recebe por parâmetro os valores de r e de n e reutiliza a função construída para cálculo do fatorial para retornar o resultado de acordo com a fórmula:

$$C_{(n,r)} = \frac{n!}{r!(n-r)!}$$

n	r	C
10	4	210

14. Escreva um programa para ler um valor n e retornar o valor da série abaixo usando uma função que, dado n , determine o valor da série:

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

n	Série
2	1.5
5	2.2833333333333333

15. Escreva um programa para verificar se os números inteiros positivos introduzidos pelo utilizador são números perfeitos (a leitura termina quando for introduzido um número não positivo). Para isso, escreva uma função que retorne `true` ou `false` caso o número passado por parâmetro seja perfeito ou não, respetivamente. Um número é perfeito se a soma dos seus divisores (exceto o próprio número) é igual ao próprio número.

Input	Output
6	É perfeito
13	Não é perfeito
28	É perfeito
150	Não é perfeito
496	É perfeito
2000	Não é perfeito
8128	É perfeito
0	

16. Escreva um programa que leia uma sequência de números inteiros positivos até encontrar um número que seja múltiplo de todos os seus algarismos, utilizando uma função que verifica se um número inteiro passado por parâmetro é múltiplo de todos os seus algarismos.

Input	Output
23	Tente outro
54	Tente outro
15	15 é múltiplo de todos os algarismos

17. Escreva um programa que leia uma sequência de números inteiros positivos (a leitura termina quando for introduzido um número não positivo) e indique se são ou não primos, utilizando uma função que determina se um dado número inteiro positivo é ou não primo. Um número primo é apenas divisível por si próprio e por 1.

Input	Output
6	Não primo
5	Primo
13	Primo
12	Não primo
-23	

18. Escreva um programa para implementar o jogo de adivinhar um número, que consiste no seguinte: o programa sorteia um número (gera um número aleatório entre 1 e 100) e o jogador deve tentar adivinhar o número sorteado. A cada tentativa o programa deve indicar se o palpite do utilizador é o correto, ou então se é maior ou menor que o número secreto. No final, deve ainda ser mostrado o número de tentativas necessárias até acertar no número. O programa deve utilizar uma função que recebe um número inteiro secreto por parâmetro e solicita ao utilizador que tente adivinhar qual é esse número, indicando se o palpite do utilizador foi maior, menor ou se acertou no número secreto; e retornando a quantidade de tentativas realizadas até acertar no número secreto.

Funções e Arrays

19. Adaptar o exercício 3, utilizando:

- Uma função para calcular a média dos valores de um array
- Uma função para determinar quantos valores de um array são inferiores a um dado valor

Input	Output
5 12 9 8 12 15	Média: 11.2. Reprovaram 2

20. Adaptar o exercício 4, utilizando:

- Uma função para determinar o maior elemento de um array
- Uma função para determinar o menor elemento de um array

Input	Output
5 12 9 8 12 15	Maior: 15 Menor: 8

21. Escreva um programa para determinar algumas estatísticas sobre vencimentos de funcionários de uma empresa. O número de funcionários pode variar ao longo do tempo, será pelos menos um, mas nunca será superior a 20. O programa deve mostrar o valor da média dos vencimentos assim como o nome dos funcionários com vencimentos inferior à média.

Organize o programa usando as seguintes funções:

- Ler os nomes e vencimentos dos funcionários da empresa. Para cada funcionário deve ser lido o seu nome e respetivo vencimento. A leitura termina quando for inserido o nome "fim".
- Calcular a média dos vencimentos.
- Mostrar a média dos vencimentos (1 C.D.)
- Mostrar os nomes dos funcionários com vencimento inferior a essa média.

Input	Output
quim 3000 ze 2500 marcia 1800 anabela 2200 paulo 1200 fim	Média: 2140,0 Funcionários com salário inferior à média: marcia paulo

22. Escreva um programa que verifique se a sequência de números inteiros inseridos pelo utilizador é sempre crescente. Para este efeito construa:

- Uma função para ler e guardar num array os números inteiros inseridos pelo utilizador. O primeiro valor introduzido é o número de valores que vão ser inseridos.
- Uma função para verificar se a sequência do array é crescente. O método deve retornar true ou false, caso a sequência seja sempre crescente ou não, respetivamente.

Input	Output
5 1 2 3 4 5	É crescente
5 5 4 3 2 1	Não é crescente
5 1 2 5 3 4	Não é crescente

23. Escreva um programa que realize a simulação de uma aplicação financeira ao fim de um ano. A aplicação consiste num depósito bancário inicial com juros capitalizáveis, isto é, no final de cada mês o montante em depósito será acrescido do juro respetivo desse mês. Para este efeito construa:

- Uma função que leia as taxas de juro para cada um dos meses do ano e retorne um array com as mesmas.
- Uma função que receba por parâmetro o valor do depósito inicial e o array com as taxas de juro e, retorne o montante final após um ano de depósito.
- A função main, que chama a leitura das taxas, lê o valor inicial e chama o cálculo do valor final

Input	Output
0,1 0,1 0,1 0,1 0,1 0,1 0,1 0,1 0,1 0,1 0,1 0,1 0,1 1000	3138.4283767209995