

# Tecnologias de Informação 2022 / 2023 SI

ISEL – DEETC – LEIM

Trabalho Prático 2

## Introdução

Os exercícios deste trabalho deverão ser resolvidos usando a linguagem de programação *Java*. A documentação do *Java* poderá ser encontrada em <https://docs.oracle.com/en/java/javase/index.html>.

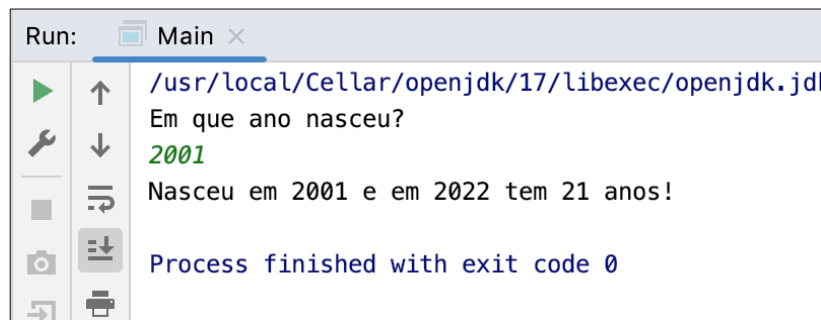
O trabalho deverá ser entregue via *moodle* e consiste num ficheiro *zip* que deverá conter os programas e um *pdf* com o relatório. O ficheiro *zip* deverá ter o seguinte formato: **TI-TP2-A50001-A50002.zip**, com os respetivos números dos dois alunos do grupo. A data limite para entrega do trabalho será no dia **25 de Novembro de 2022**.

Nota: qualquer plágio detetado implica a anulação de todos os trabalhos envolvidos.

## Programas elementares e com decisões

Para a resolução dos exercícios deste grupo não deverá recorrer nem a ciclos nem a *arrays*.

1. Implemente um programa (TP2-01-Idade) que peça o ano de nascimento ao utilizador e calcule a sua idade com base no ano actual. Pretende-se uma interação na consola semelhante ao seguinte:



```
Run: Main x
/usr/local/Cellar/openjdk/17/libexec/openjdk.jdk
Em que ano nasceu?
2001
Nasceu em 2001 e em 2022 tem 21 anos!
Process finished with exit code 0
```

2. Crie um programa (TP2-02-FormulaResolvente) que peça três números ao utilizador e encontre os zeros da função  $ax^2+bx+c$ , recorrendo à fórmula resolvente.

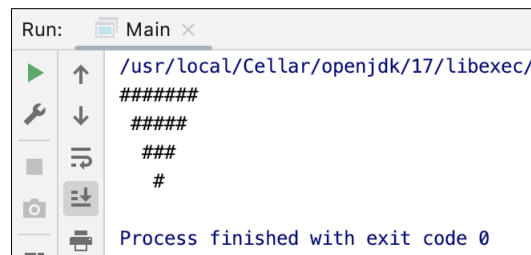
3. Implemente um programa (TP2-03-Votar) que peça o nome do utilizador e o seu ano de nascimento. O programa deverá validar se o utilizador é maior de idade e se pode votar ou não. O programa deverá informar “O ABC pode votar” ou “O ABC não pode votar”, em que “ABC” deverá ser substituído pelo nome do utilizador.
4. Implemente um programa (TP2-04-MaiorDeTres) que peça ao utilizador três números e mostre qual o número maior, o menor e o do meio. Não trate de forma especial as situações em que possa haver números iguais.
5. Implemente um programa (TP2-05-Dados) em que o computador manda um dado (número aleatório entre 1 e 6) e o jogador terá de acertar no número. Utilize a classe *Random* e o método *nextInt(int n)*.
6. Implemente o programa (TP2-06-PedraPapelTesoura) que permita jogar ao jogo da “Pedra, papel ou tesoura”. O programa deverá começar por gerar aleatoriamente um valor {0, 1, 2} que corresponde a {“Papel”, “Pedra”, “Tesoura”}. Deve então colocar numa variável do tipo *String* a palavra correspondente ao valor aleatório gerado. Depois, deve pedir ao utilizador para introduzir a sua escolha, devendo este escrever “Papel”, “Pedra” ou “Tesoura”. Esse valor deve ser lido para outra variável do tipo *String*. Por fim, deve indicar quem ganhou (o programa ou o utilizador) seguindo as regras do jogo: a tesoura ganha ao papel (cortando-o), a pedra ganha à tesoura (quebrando-a), e o papel ganha à pedra (embrulhando-a).

## Programas com ciclos

---

A resolução dos seguintes exercícios requer a utilização de ciclos (*for*, *while*, *do-while*, etc.)

7. Implemente o programa (TP2-07-ListaNumeros) que peça dois números ao utilizador e uma das seguintes *strings* (“par”, “impar”, “todos”). Caso o utilizador escolha “par”, o programa deverá mostrar todos os números pares entre os números introduzidos. Se escolher “impar”, lista apenas os ímpares, e se escolher “todos”, deverá listar todos os números entre os dois introduzidos.
8. Implemente o programa (TP2-08-Normalizar) que peça ao utilizador um nome e que o mostre normalizado. Tenha atenção às palavras como “do”, “dos”, “e”, etc., que normalizadas, deverão ficar em minúsculas. Como exemplo, o nome “JOSÉ RODRIGUES DOS SANTOS E CÉU” ficará normalizado como “José Rodrigues dos Santos e Céu”.
9. Implemente o programa (TP2-08-TrianguloInvertido) que peça um número *n* e desenhe um triângulo de cardinais invertido com *n* linhas. Ex: para *n=4* linhas, o programa deverá desenhar o seguinte triângulo invertido.



10. Implemente o programa (TP2-10-Divisao) que peça dois números inteiros positivos ao utilizador e retorne a divisão inteira do primeiro número pelo segundo número usando apenas subtrações. O programa deverá imprimir o resultado da divisão inteira e o resto da divisão.

## Programas com arrays

---

Nestes exercícios deverá utilizar *arrays*.

11. Implemente o programa (TP2-11-RandomArray) que peça um número  $n$  ao utilizador, crie um *array* com  $n$  posições, e coloque um número aleatório (entre 0 e 100) em cada uma das posições do *array*. Por fim, deverá mostrar todos os números e o seu somatório.
12. Crie um programa (TP2-12-BubbleSort) que crie um *array* com números aleatórios (entre 1 e 100) e ordene os valores no *array* por ordem crescente. Deverá usar o algoritmo *bubble sort* ([https://pt.wikipedia.org/wiki/Bubble\\_sort](https://pt.wikipedia.org/wiki/Bubble_sort)) para ordenar o *array*.
13. Crie um programa (TP2-13-MergeSorted) que crie dois *arrays* ordenados (pode mesmo criá-los já ordenados no próprio código Java) e faça a junção dos dois *arrays* num terceiro *array*, que deverá, por sua vez, também ficar ordenado. Ex: Para os *arrays*  $A=[1, 3, 5, 6]$  e  $B=[1, 2, 4, 6]$ , a junção dos *arrays* deverá resultar em  $[1, 1, 2, 3, 4, 5, 6, 6]$ .
14. Implemente um programa (TP2-14-SomaElementos) que crie dois *arrays* (pode mesmo criá-los no próprio código Java), peça um número ao utilizador, e encontre dois números no *array* cuja soma seja igual ao número inicial inserido pelo utilizador. Caso não exista, deverá informar o utilizador.
15. Faça um programa (TP2-15-SomaMatrizes) que crie duas matrizes  $n \times n$  e faça a soma das matrizes. Pode pedir o  $n$  ao utilizador e preencher a matriz com números aleatórios.  
Ex de soma de matrizes:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 3 & 2 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 1 & 2 \\ 3 & 2 & 1 \\ 1 & 3 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 3 & 3 \\ 5 & 3 & 4 \\ 4 & 5 & 3 \end{bmatrix}$$

(fim de enunciado)