Aula 3
Fluxo de controlo
Scanner

Iniciativa Conjunta:

Com o apoio de:

Instruções e Blocos de Instruções

Uma instrução é uma ação na execução do programa que pode mudar o seu estado (variáveis).
Em Java uma instrução termina sempre com um ponto-e-vírgula (;).

```
int n = 0;

int p =0;
```

Um bloco de instruções é um conjunto de instruções entre chavetas que será executado sequencialmente (p.e., numa função)

```
{
  n = n + 1;
  p = n % 2;
}
```

Estruturas de Controlo

Uma estrutura de controlo é um elemento de um programa que controla a execução de instruções, por exemplo

1. Executar uma instrução caso se verifique determinada condição;
2. Executar uma instrução caso se verifique determinada condição e outra se não se verificar essa condição
3. Executar uma instrução um determinado número de vezes
4. Executar continuamente um conjunto de instruções enquanto determinada condição se verifica

De grosso modo, as estruturas de controlo dividem-se em duas categorias: seleção (1 e 2) e repetição (3 e 4)

Estruturas de Seleção (if-else)

- Uma estrutura de seleção permite condicionar a execução de uma ou mais instruções em função de determinada condição booleana (verdadeiro/falso).

- Em Java, a estrutura de seleção mais comum é o if-else

```
if (condição) {
    instrução;
    ...
}
```

Caso a **condição** se verifique (i.e., é verdadeira), então o bloco de instruções é executado

Estruturas de Seleção (if-else)

- Uma estrutura de seleção permite condicionar a execução de uma ou mais instruções em função de determinada condição booleana (verdadeiro/falso).

- Em Java, a estrutura de seleção mais comum é o if-else

```
if (condição) {
    instrução;
    ...
} else {
    instrução;
    ...
}
```

Caso a **condição NÃO** se verifique (i.e., é verdadeira), então o bloco de instruções é executado

Exemplo (if-else)

Função que devolve o mínimo entre dois valores inteiros

```
static int min (int a, int b) {
      if (a < b) {
        return a;
      } else {
        return b;
      }
}
```

Exercício A (if-else)

- Arredondar um número decimal para inteiro.
  - Nesta função é necessário aplicar a aproximação e converter um decimal para inteiro, descartando a parte decimal, sem usar o Math.round().

    Exemplos:

    round (9.4) -> 9
    round (9.7) -> 10

Exercício B (if-else)

- Implemente a seguinte função *getTemperature* de forma a:
  - Imprima “Está gelado” se a temperatura for inferior a 0;
  - Imprima “Está frio” se a temperatura for inferior a 15;
  - Imprima “Está quente” se a temperatura for superior a 50;
  - Imprima “Está a ferver” se a temperatura for superior a 80;
  - Imprima “Está normal” para os casos que sobram.

```
public static void getTemperature(int temp) {



}
```

Estrutura de Seleção (switch)

- Uma declaração ‘switch’ é normalmente mais eficiente que um conjunto de ‘ifs’.
- A decisão entre optar por cada declaração baseia-se na legibilidade, velocidade e expressão que a declaração está a testar.

```
int variável = 5;

switch (variável) {
    case 1:
        //instrução
        ...
        break;
}
```

Estrutura de Seleção (switch)

```
public void teste (char c) {
        switch (c){
                case ( ‘A’ ) :
                    // instrução
                    break;
                case ( ‘B’ ) :
                    // instrução
                    break;
                default:
                   // caso não seja nenhum dos casos
        }
}
```

Exercício C - Switch

Repetir o exercício B, agora usando switch-case em vez de ifs.

Switch x If-else

Switch	If-else
Testa expressões baseadas em apenas um único número inteiro, valor enumerado, ou objeto string.	Pode testar expressões baseadas em intervalos de valores ou condições.
Ótimos para valores de dados fixos.	Melhor para valores booleanos.
Maior velocidade de processamento quando o número de casos a analisar for superior a 5.	Quando o número de casos é pequeno (<5) não são percetíveis diferenças de velocidade de processamento.
O código é mais fácil de ler.	Uma grande sucessão de ifs pode tornar-se confuso e vulnerável a erros.
Adicionar ou remover valores é mais simples e fácil de alterar.	

Scanner - Output

- Para mostrar informação no ecrã, podemos usar o objeto que normalmente representa o canal de escrita de dados no ecrã, o **System.out**

```
public class Program { public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
         System.out.println("Como te chamas?");
        String name = keyboard.nextLine();
        System.out.println("Quantos anos tens?");
        int age = keyboard.nextInt();
        System.out.print("Olá " + name + "! ");
        System.out.println("Tens " + age + " anos.");
}
```

Scanner - Input

- Para obter informação introduzida pelo utilizador através do teclado pode ser utilizado um objeto do tipo Scanner
    - Para usar este tipo de objetos temos de fazer:
        import java.util.Scanner;

- O objeto do tipo Scanner tem de estar associado ao objeto que representa o **canal de leitura de dados** normalmente associado ao teclado, o **System.in**

```
public class Program { public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        String name = keyboard.nextLine();
         int age = keyboard.nextInt();
}
```

Exercício D

- Criar uma função que nos diga se um número inteiro dado pelo utilizador é positivo, negativo ou zero.

Exercício D

- Criar uma função que nos diga se um número inteiro dado pelo utilizador é positivo, negativo ou é zero.

```
public static void main(String[] args)
  {
      Scanner in = new Scanner(System.in);
      System.out.print("Input number: ");
      int input = in.nextInt();

      if (input > 0)
      {
          System.out.println("Number is positive");
      }
      else if (input < 0)
      {
          System.out.println("Number is negative");
      }
      else
      {
          System.out.println("Number is zero");
      }
  }
```

Exercício E

Após pedir três números inteiros ao utilizador, o programa imprime o maior valor.

Exemplo:

a = 25;
b = 61;
c = 18;

→ Resultado = 61;

Exercício E

Após pedir três números inteiros ao utilizador, o programa imprime o maior valor.

```java
import java.util.Scanner;
public class Exercise3 {


  public static void main(String[] args) {
  Scanner in = new Scanner(System.in);

  System.out.print("Input the 1st number: ");
  int num1 = in.nextInt();

  System.out.print("Input the 2nd number: ");
  int num2 = in.nextInt();

  System.out.print("Input the 3rd number: ");
  int num3 = in.nextInt();


  if (num1 > num2)
   if (num1 > num3)
    System.out.println("The greatest: " + num1);

  if (num2 > num1)
   if (num2 > num3)
    System.out.println("The greatest: " + num2);

  if (num3 > num1)
   if (num3 > num2)
    System.out.println("The greatest: " + num3);
 }
}
```

https://www.w3resource.com/java-exercises/conditional-statement/index.php