ALGORITMOS E ESTRUTURAS DE DADOS

Tópicos essenciais de
algoritmia e programação

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

VARIÁVEIS

- Localização de memória capaz de reter um valor;
- Essa localização é representada por um "nome" que funciona como um identificador da zona de memória onde está guardado o valor;
- Uma variável pode ser imaginada como uma caixa capaz de albergar um e apenas um valor a cada momento;
- Além do nome, essa caixa tem um tamanho medido em bytes e um tipo de dados não alterável que define o tipo de valor que pode lá ser guardado.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

CONSTANTES

- Significado e modo de funcionamento muito similar às variáveis, mas o valor é apenas possível de definir uma vez.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

TIPOS DE DADOS

PSEUDO-CODIGO		TIPOS DE DADOS MAIS USADOS NA LINGUAGEM C#	
INTEIRO	byte	1Byte	De 0 a 255
	int	4Bytes	De -2147483648 a 2147483647
	uint	4Bytes	De 0 a 4294967295
	short	2Bytes	De -32768 a 32767
	ushort	2Bytes	De 0 a 65535
	long	8Bytes	De -922337203685477508 a 922337203685477507
	ulong	8Bytes	De 0 a 18446744073709551615
REAL	float	4Bytes	De -3.402823e38 a 3.402823e38
	double	8Bytes	De -1.79769313486232e308 a 1.79769313486232e308
	decimal	16Bytes	De ±1.0 × 10e−28 a ±7.9 × 10e28
TEXTO OU CARACTER	string		Unicode → 1 Caracter = 2 Bytes
	char	2Bytes	Unicode → 1 Caracter = 2 Bytes
LOGICO	bool	1Byte	[verdadeiro|falso] *precisaria apenas de 1bit, mas a unidade mínima endereçável de memória é o Byte*

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES MATEMÁTICOS

Operator	Description	Example (x=100, y=50)
+	Used to add two values or operands	x+y=150
-	Used to subtract second value from the expression	x-y=-50
*	Used to Multiply two operands or values	x*y=5000
/	Used to divide numerator by de-numerator	x/y=2
%	Used to get the remainder after an integer division	x%y=0

http://www.tutorialsleader.com/csharp/c-sharp-operators.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES RELACIONAIS

Operator	Description	Example (x=100, y=50)
==	Used to check if two operands are equal or not. It returns true in case if both operands are equal else it returns false.	x==y returns false
!=	It works opposite of == operator. Returns true if two values are not same. It returns false if two values are same.	x!=y returns true
>	Used to check if the left value is greater than the value of right side. It returns true if the left side value is greater.	x>y returns true
<	Used to check if the left value is smaller than the value of right side. It returns true if the left side value is smaller.	x>y returns false
>=	Used to check if the left value is either greater or equal to the value of right side. It returns true if any conditions match.	x>=y returns true
<=	Used to check if the left value is smaller or equal to the value of right side. It returns true if any condition matches.	x<=y returns false

http://www.tutorialsleader.com/csharp/c-sharp-operators.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES RELACIONAIS (EXERCÍCIO)

Calcule o resultado das seguintes expressões, tendo em conta os valores atribuídos.

- a = 3; b = 7; c = 4;
  - (a + b) > b
  - b >= (a + 2)
  - c == b - a
  - (b + a) <= c
  - (c + a) > b

Implemente código em C# para validar os resultados apresentados na consola.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES RELACIONAIS (EXERCÍCIO)

```
static void Main(string[] args)
{
    // variaveis
    string nome;
    int idade, maioridade;
    bool maior;       // false, true

    // atribuir valores
    nome = "manuel";
    idade = 19;
    maioridade = 18;

    // comparar a idade com a maioridade e atribuir o
    // resultado logico à variavel maior
    maior = idade >= maioridade;

    // mostrar o numero atual no ecra
    Console.WriteLine("O {0} é maior de idade?: {1}", nome, maior);

    // pausa antes de fechar a aplicação
    Console.ReadKey();
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES LÓGICOS

Operator	Description	Example (a=true, b=false)
&&	The AND operator returns true if both operands have TRUE value.	a&&b returns false
||	The OR operator returns true if any operator from both has a TRUE value.	a||b returns true
!	NOT operator is used to reverse the operand's value. It converts TRUE into FALSE and vice versa.	!(a&&b) returns true

http://www.tutorialsleader.com/csharp/c-sharp-operators.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES LÓGICOS (EXERCÍCIO)

Determine o resultado das seguintes expressões, considerando as variáveis do tipo lógico e as respetivas atribuições.

- a = verdadeiro; b = falso; c = falso; d = verdadeiro;
  - a || (b && c) || (!d || !a)
  - !(!a || b) && (c || d)
  - b && !c || a && d
  - !a && !!c && d

Implemente código em C# para validar os resultados apresentados na consola.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES LÓGICOS (EXERCÍCIO)

```
static void Main(string[] args)
{
    // Declarar variaveis e atribuir valores exemplo
    string nome = "António";
    int idade = 22;              // SUGESTÃO: testar outros valores
    float altura = 1.65F;        // chega aos pedais? (testar outros valores)
    bool cartaConducao = false;  // tem carta de condução? (outros valores)
    bool consegueConduzir;

    /*
     * Basta que uma pessoa tenha carta de condução ou que tenha no minimo
     * 22 anos que se assume conseguir conduzir. No entanto, nunca deve ter altura
     * inferior a 1.5m senão, não chega aos pedais!
     */
    consegueConduzir = (altura >= 1.5F) && ((idade >= 22) || (cartaConducao == true));

    // escreve o resultado
    Console.WriteLine("O {0} consegue conduzir?: {1}", nome, consegueConduzir.ToString());

    // aguarda o pressionar de qualquer tecla para ver o resultado
    Console.ReadKey();
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES DE ATRIBUIÇÃO

Operator	Description	Example If (a=100, b=50)
=	= Operator works from Right to left. It assigns values from right side operand to left side operand.	a = b Results a =50
+=	+= Operator is called add AND operator. First, it sums both operands(left and right) value and then it assigns the resultant value to the left operand.	a += b works same as a = a + b which Results a=150
-=	-= named as Subtract AND assignment operator. First, it subtracts the right operand's value from the left operand value and then it assigns the resultant value to the left operand.	a -= b works same as a = a - b which Results a = 50
*=	*= named as Multiply AND assignment operator. First, it multiplies both operands(left and right) value and then it assigns the resultant value to the left operand.	a *= b works same as a = a * b which Results a = 5000
/=	/= named as divide AND assignment operator. First, it divides left operand value from the right operand value and then it assigns the resultant value to the left operand.	a /= b works same as a = a / b which Results a = 2
%=	%= named as modules AND assignment operator. First, it divides left operand value from the right operand value and then it assigns modulus value to the left operand.	a %= b works same as a = a % b which Results a=0

Operator	Description	Example If (a=100, b=50)
<<=	<<= named as left shift AND assignment operator. First, it left shifts, left side operand value to that number of times specified on the right side. and then it assigns the resultant value to the left operand.	a <<= 2 works same as a = a << 2 which Results a = 400
>>=	>>= named as right shift AND assignment operator. First it right shifts, left side operand value to that number of times specified on the right side, and then it assigns the resultant value to the left operand.	a >>= 2 works same as a = a >> 2 which Results a = 25
&=	&= named as AND assignment operator. It performs a bitwise logical AND operation then it assigns the resultant value to the left operand.	a &= b works same as a = a & b which Results a = 32
^=	^= named as exclusive OR assignment operator. It performs a bitwise exclusive-OR operation and then it assigns the resultant value to the left operand.	a ^= b works same as a = a ^ b which Results a = 86
|=	|= named as OR assignment operator. It performs a bitwise OR operation and then it assigns the resultant value to the left operand.	a |= b works same as a = a | b which Results a = 118

http://www.tutorialsleader.com/csharp/c-sharp-operators.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

OPERADORES DE ATRIBUIÇÃO (EXERCÍCIO)

Determine o resultado das seguintes expressões, considerando as variáveis do tipo lógico e as respetivas atribuições.

- a = 3; b = 2; c = -1;
  - a += b;
  - a += c;
  - b *= c
  - c *= c
  - a /= c + b

Implemente código em C# para validar os resultados apresentados na consola.

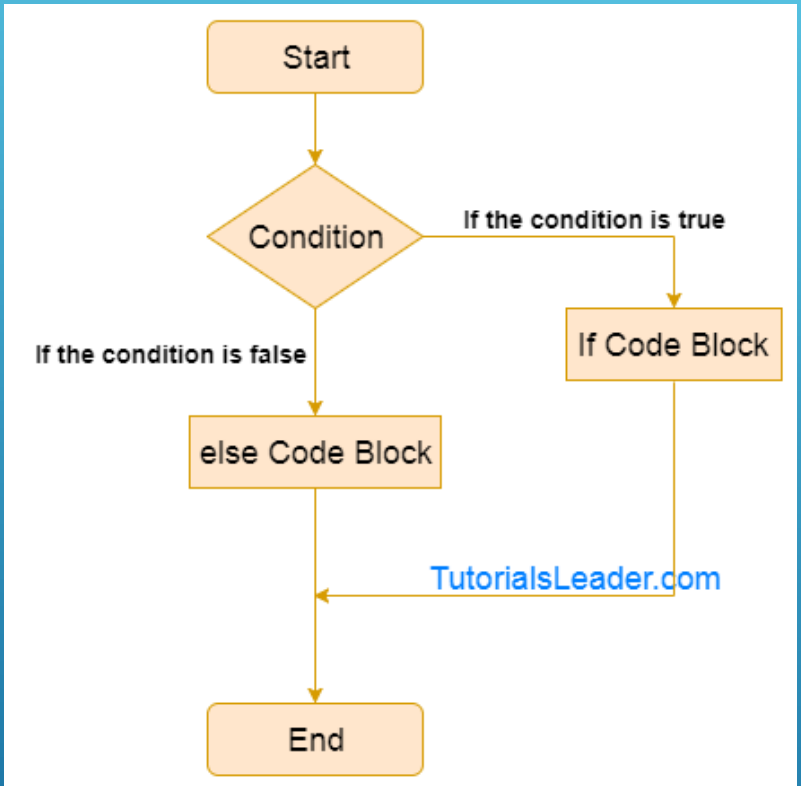
OPERADORES DE ATRIBUIÇÃO (EXERCÍCIO)

```
static void Main(string[] args)
{
    // declarar variáveis e atribuir valores
    int numero = 5;
    int valor = 10;

    // incrementar 10 unidades a numero
    numero = numero + 10;

    // somar o valor a numero
    numero += valor;

    // mostrar o numero atual no ecra
    Console.WriteLine("O valor atual é: {0}", numero);

    // retirar o valor 3
    numero -= 3;

    // mostrar o numero atual no ecra
    Console.WriteLine("O valor atual é: {0}", numero);


    // aguardar o pressionar de uma tecla
    Console.ReadKey();
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO

INSTRUÇÕES DE DECISÃO | IF… ELSE…




```
static void Main()
{
    int a = 50, b = 20, c = 30;

    if (a > b)
    {
        if (a > c)
        {
            Console.WriteLine("a is greater than b and c");
        }
        else
        {
            Console.WriteLine("a is greater than b but less than c");
        }
    }
    else
    {
        if (a > c)
        {
            Console.WriteLine("a is less than b but greater than c");
        }
        else
        {
            Console.WriteLine("a is less than b and c");
        }
    }
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO | IF… ELSE… (EXERCÍCIO)

ENUNCIADO

Devido à crise, o governo decidiu criar uma sobretaxa de IRS extraordinária de 3,5% sobre os salários cujo valor seja superior ao salário mínimo nacional. Esta nova taxa, apenas deve incidir sobre o valor que remanesce depois de subtraído o salário mínimo ao salário do funcionário.

Sugira um algoritmo para resolver o problema, acautelando a exceção prevista para os funcionários que apenas aufiram o salário mínimo nacional.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO | IF… ELSE… (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// definir constantes
const double salarioMinimo = 485.0;

// declarar variáveis e atribuir valores
string nome;
double salario, valorSobretaxa;


// limpar consola
Console.Clear();

// recolher nome do funcionario
Console.Write("Insira o nome do funcionario: ");
nome = Console.ReadLine();

// recolher o salario do funcionario
Console.WriteLine("");
Console.Write("Insira o salário do funcionario: ");
salario = float.Parse(Console.ReadLine());
```

```
if ((salario - salarioMinimo) > 0)
{
    // calcular o valor da sobretaxa
    valorSobretaxa  = (salario - salarioMinimo) * (3.5/100);

    // retirar o valor ao salário
    salario -= valorSobretaxa;
}

// mostrar o valor do salário
Console.WriteLine("");
Console.WriteLine("Salário de {0}: {1} euros.", nome, salario);

// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO | IF… ELSE… (EXERCÍCIO)

ENUNCIADO

Na Unidade Curricular de português foi definida a nota mínima de 8,5 valores para o primeiro teste e de 9 valores para o segundo. Foi também decidido que a nota final à unidade seria a média aritmética entre as notas desses dois testes.

A aprovação de um aluno é apenas possível se a nota final (média entre os dois testes) for igual ou superior a 10 valores. Se o aluno tiver uma nota final igual ou superior a 17 valores terá que fazer defesa oral. Nestes casos, a nota final do aluno será sempre inserida diretamente pelo professor. Caso não compareça à defesa, o professor atribuirá a nota final de 17 valores.

Desenvolva um algoritmo para resolver o problema e sugira uma implementação.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

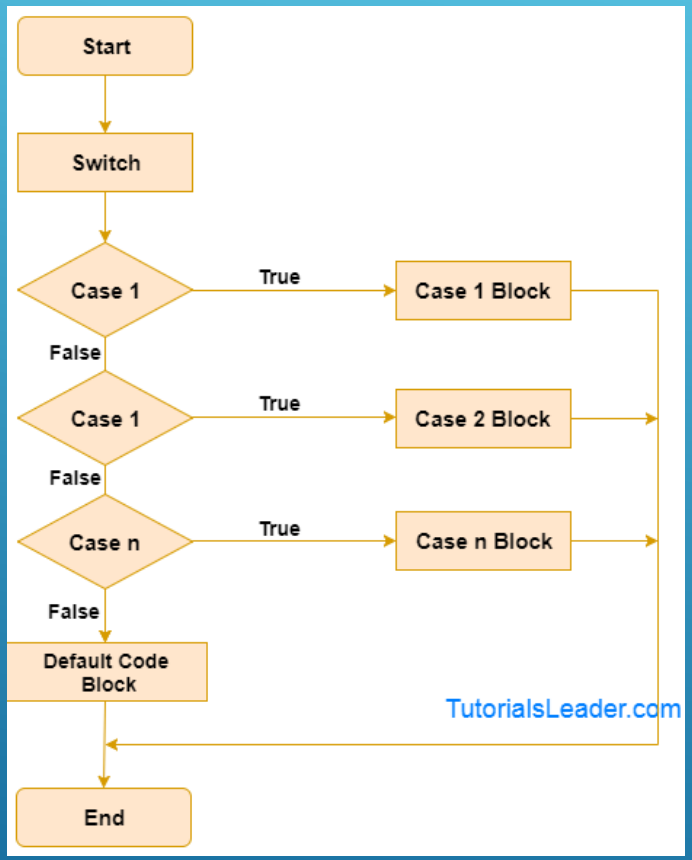
INSTRUÇÕES DE DECISÃO | IF... ELSE... (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// definir variáveis
float nota1, nota2, notaFinal;
string nome;


// limpar consola
Console.Clear();

// recolher nome do aluno
Console.Write("Insira o nome do aluno: ");
nome = Console.ReadLine();

// recolher nota primeiro teste
Console.WriteLine("");
Console.Write("Insira a nota do primeiro teste: ");
nota1 = float.Parse(Console.ReadLine());

// recolher nota segundo teste
Console.WriteLine("");
Console.Write("Insira a nota do segundo teste: ");
nota2 = float.Parse(Console.ReadLine());
```

```
// decidir aprovação do aluno
if (nota1 >= 8.5 && nota2 >= 9)
{
    // calcular notaFinal
    notaFinal = (nota1 + nota2) / 2;

    // verificar nota final
    if (notaFinal >= 10 && notaFinal < 17)
    {
        Console.WriteLine("");
        Console.WriteLine("O aluno {0} está aprovado com a nota: {1}.", nome, notaFinal);
    }
    else if (notaFinal >= 17)
    {
        Console.WriteLine("");
        Console.Write("Introduza a nota final do aluno (defesa oral) {0}: ", nome);
        notaFinal = float.Parse(Console.ReadLine());
        Console.WriteLine("");
        Console.WriteLine("A classificação final do aluno {0}, é: {1}.", nome, notaFinal);
    }
    else
    {
        // reprovado
        Console.WriteLine("");
        Console.WriteLine("O aluno {0} está reprovado com a nota {1}!", nome, notaFinal);
    }

}
else
{
    // reprovado
    Console.WriteLine("");
    Console.WriteLine("O aluno {0} está reprovado!", nome);
}
```

INSTRUÇÕES DE DECISÃO | SWITCH




```
static void Main()
{
    int a = 2;

    switch (a)
    {
        case 0:
            Console.WriteLine("The value of a is zero");
            break;
        case 1:
            Console.WriteLine("The value of a is one");
            break;
        case 2:
            Console.WriteLine("The value of a is two");
            break;
        case 3:
            Console.WriteLine("The value of a is three");
            break;
        case 4:
            Console.WriteLine("The value of a is four");
            break;
        case 5:
            Console.WriteLine("The value of a is five");
            break;
        default:
            Console.WriteLine("The value is not defined in the switch statement");
            break;
    }
}
```

http://www.tutorialsleader.com/csharp/c-sharp-switch-statement.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO | SWITCH (EXERCÍCIO)

ENUNCIADO

Um clube de futebol pretende um programa que lhe facilite o processo de classificação de atletas em categorias. A tabela apresentada abaixo, define as categorias existentes e também os intervalos de idade que especificam a categoria para cada atleta.

Com o objetivo de promover as inscrições na categoria Juvenil, o clube oferece a taxa de inscrição aos atletas com 11 e 12 anos. Certifique-se que o utilizador é lembrado desse facto na sua implementação.

De	ATÉ	Categoria
11	15	Juvenil
16	20	Júnior
21	25	Profissional

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO | SWITCH (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// variáveis
string nome;
int idade;

// limpar consola
Console.Clear();

// solicitar e recolher dados do atleta
Console.Write("Insira o nome do atleta: ");
nome = Console.ReadLine();
Console.Write("Insira a idade do atleta: ");
idade = int.Parse(Console.ReadLine());
```

```
// decidir categoria e informar o utilizador
switch (idade)
{
    case 11:
        Console.WriteLine("LEMBRETE: inscrição gratuita (atletas de 11 e 12 anos)!");
        Console.WriteLine("O atleta {0} pertence à categoria {1}", nome, "juvenil");
        break;

    case 12:
        Console.WriteLine("LEMBRETE: inscrição gratuita (atletas de 11 e 12 anos)!");
        Console.WriteLine("O atleta {0} pertence à categoria {1}", nome, "juvenil");
        break;

    case 13:
    case 14:
    case 15:
        Console.WriteLine("O atleta {0} pertence à categoria {1}", nome, "juvenil");
        break;

    case 16:
    case 17:
    case 18:
    case 19:
    case 20:
        Console.WriteLine("O atleta {0} pertence à categoria {1}", nome, "junior");
        break;

    case 21: case 22: case 23: case 24: case 25:
        Console.WriteLine("O atleta {0} pertence à categoria {1}", nome, "profissional");
        break;

    default:
        Console.WriteLine("Categoria não prevista!!!");
        break;
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO | SWITCH (EXERCÍCIO)

ENUNCIADO

Para facilitar o controlo no consumo de energia, é necessário criar uma pequena aplicação que estime o valor da fatura mensal de um cliente, com base nos Kw estimados de consumir num mês (estimativa de consumo num mês). Tenha em atenção que, o custo por Kw varia em função do tipo de cliente:

- Particulares (continente) – 0,1865 €
- Particulares (ilhas) – 0,1875 €
- Pequenas empresas – 0,1754 €
- Médias e grandes empresas – 0,1592 €
- Estado e organismos públicos – 0,1311 €

Note que aos valores indicados é necessário acrescer IVA à taxa de 23%. Desenvolva um algoritmo que torne simples a resolução deste cálculo..

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO | SWITCH (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// declarar variaveis
string opcaoEscolhida;
int kwEstimativa;
double valorEstimado;

// limpar consola
Console.Clear();

// Solicitar e recolher valores do utilizador
Console.WriteLine("===== Estimativa do custo mensal de energia =======");
Console.WriteLine(" by Célio Carvalho");
Console.WriteLine("");
Console.WriteLine("     Escolha o seu tipo de cliente:");
Console.WriteLine("        > 1 - Particulares (continente)");
Console.WriteLine("        > 2 - Particulares (ilhas)");
Console.WriteLine("        > A - Pequenas empresas");
Console.WriteLine("        > B - Medias e grandes empresas");
Console.WriteLine("        > C - Estado e organismos públicos");
Console.WriteLine("");
Console.Write("                    Opcao: ");

// opcao escolhida
opcaoEscolhida = Console.ReadLine();
Console.WriteLine("");

// kw estimados de consumir
Console.Write("      Insira os Kw que estima consumir no mês: ");
kwEstimativa = int.Parse(Console.ReadLine());
```

```
// calcular valor estimado tendo em conta o tipo de cliente
switch (opcaoEscolhida)
{
    case "1":
        // Particulares (continente) - 0,1865
        valorEstimado = kwEstimativa * 0.1865;
        break;

    case "2":
        // Particulares (ilhas) - 0,1875
        valorEstimado = kwEstimativa * 0.1875;
        break;

    case "A": case "a":
        // Pequenas empresas - 0,1754
        valorEstimado = kwEstimativa * 0.1754;
        break;

    case "B": case "b":
        // Medias e grandes empresas - 0,1592
        valorEstimado = kwEstimativa * 0.1592;
        break;

    case "C": case "c":
        // Estado e organismos publicos - 0,1311
        valorEstimado = kwEstimativa * 0.1311;
        break;

    default:
        // caso não previsto
        valorEstimado = -1;
        break;
}

// acrescentar o valor do iva
valorEstimado += (valorEstimado * (23.0 / 100));

// mostrar resultado ao utilizador
Console.WriteLine("");
Console.WriteLine("==== RESULTADOS ================");

if (valorEstimado > 0)
    Console.WriteLine("Valor mensal estimado: {0} Eur", valorEstimado.ToString("###,##0.00"));
else
    Console.WriteLine("((tipo de cliente inserido inválido!))");

// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO

ENUNCIADO

O índice de massa corporal (IMC) de alguém é calculado através da divisão do peso em Kg pela sua altura em m2. Assim, por exemplo, uma pessoa de 1,67m que pese 55Kg tem um IMC = 19,72 porque,

$$IMC = \frac{\text{peso}}{\text{altura2}} = \frac{55\text{Kg}}{1{,}67\text{m} * 1{,}67\text{m}} = 19{,}72$$

- Desenhe e implemente um algoritmo que recolha os dados referentes a uma pessoa e calcule o seu IMC.

- Evolua o algoritmo anterior para que mostre ao utilizador de forma automática a interpretação do índice introduzido.

IMC	Interpretação
Até 18,5 (inclusive)	Peso abaixo do recomendado
De 18,5 até 25 (inclusive)	Peso normal
De 25 a 30 (inclusive)	Peso acima do recomendado
Mais que 30	Peso muito acima do recomendado (obesidade)

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO

ENUNCIADO

Numa determinada empresa há a necessidade de criar uma aplicação que identifique o tipo de cliente em função do seu volume de compras anual (ver quadro seguinte).

TIPO CLIENTE	ATÉ (inclusive)
NORMAL	5.000,00 €
PROFISSIONAL	20.000,00 €
EMPRESARIAL	999.999,99 €

Desenvolva um algoritmo capaz de automatizar esta identificação e, de seguida, sugira uma implementação.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE DECISÃO

ENUNCIADO

Uma loja de venda de animais pretende criar uma aplicação que automatize o processo de cálculo de PVP (preços de venda a publico). A margem a ser praticada varia em função da família dos artigos vendidos, (conforme tabela seguinte).

CÓDIGO	FAMILIA	MARGEM
1	COBRAS	3,0 %
4	RATOS	4,0 %
9	CAES	2,5 %

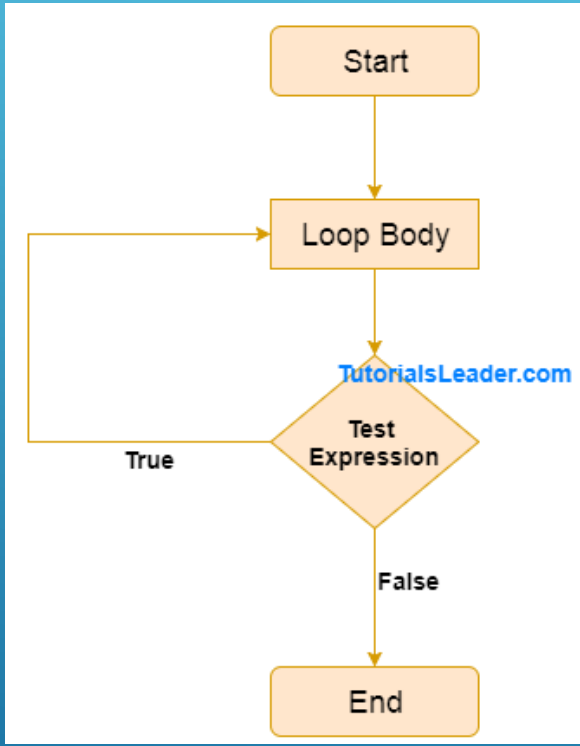
Desenvolva e implemente um algoritmo capaz de automatizar esta identificação, de forma a mostrar o PVP de um artigo com e sem IVA (23%) atendendo à família a que pertence.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO | DO{…} WHILE(…);




```
static void Main()
{
    int x = 0;
    do
    {
        Console.WriteLine("The value of x is = {0}", x);
        x++;
    } while (x < 10);
}
```

```
The value of x is = 0
The value of x is = 1
The value of x is = 2
The value of x is = 3
The value of x is = 4
The value of x is = 5
The value of x is = 6
The value of x is = 7
The value of x is = 8
The value of x is = 9
```

http://www.tutorialsleader.com/csharp/c-sharp-do-while-loop.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | DO{…} WHILE(…); (EXERCÍCIO)

ENUNCIADO

Pretende-se criar uma aplicação que, perante várias inserções de números naturais (N0) pelo utilizador, calcule a quantidade de números pares inseridos (considerar o zero não válido como par).

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | DO{…} WHILE(…); (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// declarar variaveis necessárias
int contador = 0;
int numero;

// limpar consola
Console.Clear();

// informar utilizador como finalizar o ciclo
Console.WriteLine(">> para terminar insira um numero inteiro negativo <<");

do
{
    // solicitar inserção de novo numero
    Console.Write("insira um numero natural: ");
    numero = int.Parse(Console.ReadLine());

    // validar se o numero inserido cumpre as regras para ser contado:
    //    - não é negativo
    //    - não é zero
    //    - é par
    if (numero > 0 && numero % 2 == 0)
    {
        contador++;
    }

} while (numero >= 0);

// escrever o resultado
Console.WriteLine("");
Console.WriteLine("Quantidade de naturais inseridos, pares maiores que zero: {0}", contador.ToString());

// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | DO{…} WHILE(…); (EXERCÍCIO)

ENUNCIADO

Um centro comercial pretende adequar um pouco mais a decoração do seu espaço de lazer e de alimentação à faixa etária dos seus visitantes. Para isso, no próximo fim-de-semana, irá recolher informação das pessoas que lá entram, com o objetivo de determinar:

- O número total de visitantes femininos e masculinos;
- O número de visitantes com idade compreendida entre os 10 e 16 anos e os 16 e 24 anos.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | DO{…} WHILE(…); (EXERCÍCIO)

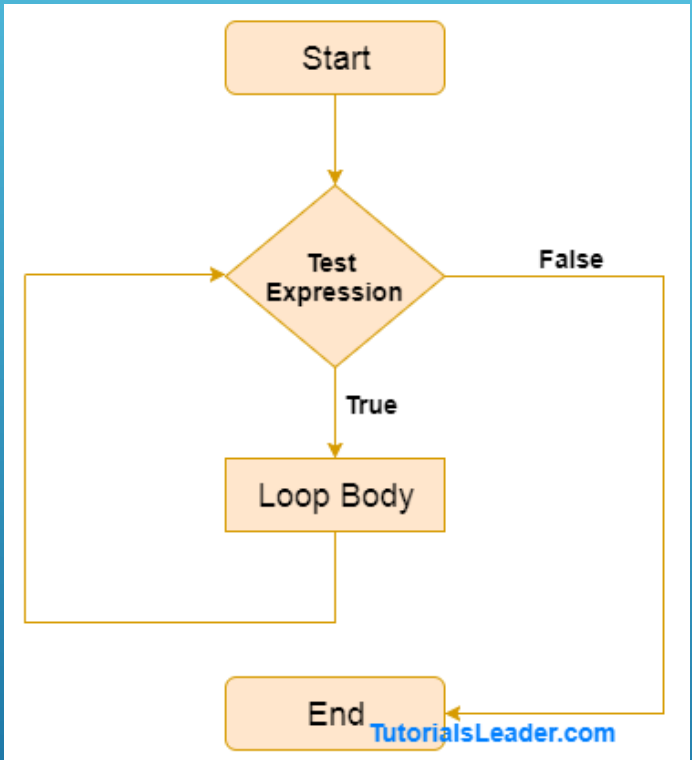
PROPOSTA DE SOLUÇÃO

```
// declarar variaveisa
char genero;
int idade, contadorFemininos = 0, contadorMasculinos = 0,
    contadorIdade10Ate16 = 0, contadorIdade16Ate24 = 0;

do
{
    // limpar consola
    Console.Clear();

    // informar utilizador de como terminar aplicação
    Console.WriteLine("=== REGISTO DE NOVO VISITANTE ==============(terminar com Z)==");
    Console.WriteLine("");

    // recolher género
    Console.Write("     -> Genero (M/F): ");
    genero = Console.ReadKey().KeyChar;

    // tratar somente se M ou F
    if (genero == 'M' || genero == 'm' || genero == 'F' || genero == 'f')
    {

        // recolher idade
        Console.WriteLine("");
        Console.Write("     -> Idade (0:99): ");
        idade = int.Parse(Console.ReadLine());

        // processar genero
        if (genero == 'M' || genero == 'm')
            contadorMasculinos++;
        else
            contadorFemininos++;

        // processar idade
        if (idade >= 10 && idade < 16)
            contadorIdade10Ate16++;
        else if (idade >= 16 && idade < 24)
            contadorIdade16Ate24++;

    }

} while (genero != 'Z' && genero != 'z');
```

```
// mostrar resultados
Console.WriteLine("");
Console.WriteLine("");
Console.WriteLine("====== RESULTADOS ========================== by Célio Car
Console.WriteLine("Visitantes = femininos + masculinos = {0} + {1} = {2}",
    contadorFemininos.ToString(),
    contadorMasculinos.ToString(),
    (contadorFemininos + contadorMasculinos).ToString()
    );
Console.WriteLine("Idade [10:16[ = {0}  ***  Idade [16:24[ = {1}",
    contadorIdade10Ate16.ToString(),
    contadorIdade16Ate24.ToString()
    );

// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO | WHILE(…) {…}




```
static void Main()
{
    int i = 0;
    while (i < 10)
    {
        Console.WriteLine("Value of i is : {0}", i);
        i++;
    }
}
```

```
Value of i is : 0
Value of i is : 1
Value of i is : 2
Value of i is : 3
Value of i is : 4
Value of i is : 5
Value of i is : 6
Value of i is : 7
Value of i is : 8
Value of i is : 9
```

http://www.tutorialsleader.com/csharp/c-sharp-while-loop.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | WHILE(…) {…} (EXERCÍCIO)

ENUNCIADO

O organismo público responsável pela meteorologia em Portugal necessita de fazer um estudo acerca da temperatura na cidade do Porto.

Crie uma aplicação capaz de recolher durante n dias a temperatura, e no final, informar qual o pico de calor ocorrido, assim como a média das temperaturas recolhidas. Em cada dia apenas será efetuada uma única recolha.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | WHILE(…) {…} (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// declarar variáveis necessárias
int dias = 0;
double leitura, maximo = -999, acumuladoLeituras = 0;

// limpar ecra
Console.Clear();

// comunicar a condição de paragem ao utilizador
Console.WriteLine("== REGISTO DE TEMPERATURA ========= (terminar com
Console.WriteLine("");

// solicitar a primeira leitura ao utilizador
Console.Write("Insira a temperatura do dia 1: ");
leitura = double.Parse(Console.ReadLine());
```

```
"== REGISTO DE TEMPERATURA ========= (terminar com 999)==");
```

```
// executar ciclo enquanto que o valor inserido seja menor que 999
while (leitura < 999)
{
    // atualizar variaveis de estado
    dias++;
    acumuladoLeituras += leitura;

    // verificar se existe novo maximo
    if (leitura > maximo)
        maximo = leitura;

    // nova leitura
    Console.Write("Insira a temperatura do dia {0}: ", (dias + 1).ToString());
    leitura = double.Parse(Console.ReadLine());
}

// mostrar resultados
Console.WriteLine("");
Console.WriteLine("");
Console.WriteLine("== RESULTADOS (Porto) ============= by Célio Carvalho ==");
Console.WriteLine("Numero de dias tratados: {0}", dias.ToString());

// mostrar maximo e media (atender ao problema da divisão por zero)
if (dias > 0)
{
    Console.WriteLine("Temperatura máxima: {0}", maximo);
    Console.WriteLine("Temperatura média: {0}", (acumuladoLeituras / dias).ToString());
}
else
{
    Console.WriteLine("Temperatura máxima: (sem valores)");
    Console.WriteLine("Temperatura média: (sem valores)");
}


// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | WHILE(…) {…} (EXERCÍCIO)

ENUNCIADO

A Associação de estudantes de Escola Secundária "Só Crânios" está a organizar um concurso de matemática com o objetivo de encontrar os 3 melhores alunos à disciplina.

A equipa organizadora do concurso necessita de uma aplicação que faça as inscrições de alunos, através da recolha do nome e da nota de 11º ano a matemática.

O número de alunos inscritos no concurso nunca poderá ser inferior a 5 e superior a 10. No entanto, as inscrições terminam se a média das notas dos alunos inscritos atingir os 18 valores, desde que cumprido o preceito do mínimo de 5 alunos inscritos.

Caso um candidato tenha nota inferior a 14 valores, a sua inscrição é automaticamente recusada.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | WHILE(…) {…} (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

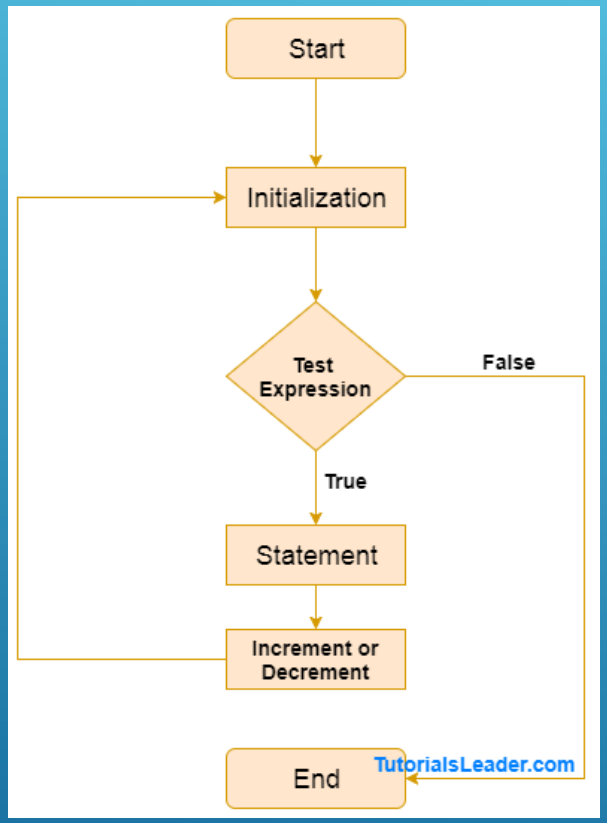
```
// declarar variaveis necessárias
string nome, lista = "";
double nota, soma = 0;
int numeroInscritos = 0;
```

```
// (ou) while (numeroInscritos < 5 || (numeroInscritos < 10 && (soma / numeroInscritos) < 18))
while (numeroInscritos != 10 && (numeroInscritos < 5 || (soma / numeroInscritos) < 18))
{
    // limpar a consola
    Console.Clear();

    // solicitar e recolher informação ao utilizador
    Console.WriteLine("= NOVO CANDIDATO ======= (inscritos: {0}), (média atual: {1})",
        numeroInscritos.ToString(),
        numeroInscritos > 0 ? (soma / numeroInscritos).ToString() : "0"
        );
    Console.WriteLine("");
    Console.Write("  > NOME: ");
    nome = Console.ReadLine();
    Console.Write("  > NOTA DO 11º ANO: ");
    nota = double.Parse(Console.ReadLine());

    // avaliar cumprimento das regras
    if (nota < 14)
    {
        // se nota infeior a 14 rejeita candidatura
        Console.WriteLine("");
        Console.WriteLine("\t\tINSCRIÇÃO RECUSADA (nota inferior a 14 valores)");
        Console.ReadKey();
        continue;   // salta para proxima iteração
    }

    // incrementar o contador do numero de inscritos
    numeroInscritos++;

    // atualizar valor
    soma += nota;

    // guardar informação acerca da inscricao
    lista += string.Format("{0}) {1}\t\t{2} valores\n", numeroInscritos.ToString("00"), nome, nota);

    // informar que a inscrição foi aceite
    Console.WriteLine("\t\tINSCRIÇÃO ACEITE");
    Console.ReadKey();
}
```

```
// mostrar resultados
Console.WriteLine("TOTAL DE INSCRITOS: {0}", numeroInscritos.ToString());
Console.WriteLine("MÉDIA DOS INSCRITOS: {0}", (soma / numeroInscritos).ToString());
Console.WriteLine("------------------- RELATÓRIO DE INSCRITOS -----------------");
Console.WriteLine(lista);

// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO | FOR(…;…;…){…}




```
static void Main()
{
    for (int i = 0; i < 10; i++)
    {
        Console.WriteLine("The value of i is : {0}", i);
    }
}
```

```
The value of i is : 0
The value of i is : 1
The value of i is : 2
The value of i is : 3
The value of i is : 4
The value of i is : 5
The value of i is : 6
The value of i is : 7
The value of i is : 8
The value of i is : 9
```

http://www.tutorialsleader.com/csharp/c-sharp-for-loop.htm

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | FOR(…;…;…){…} (EXERCÍCIO)

ENUNCIADO

Desenvolva um programa capaz de calcular o fatorial de um dado numero (ver formula seguinte).

$$\texttt{factorial}(n) = \begin{cases} n = 0 & \rightarrow & 1 \\ n \geq 1 & \rightarrow & n * \texttt{factorial}(n-1) \end{cases}$$

Exemplo: factorial(5)=5*4*3*2*1=120

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | FOR(…;…;…){…} (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// declarar variaveis
byte numero;
uint fatorial;

// limpar ecra
Console.Clear();

// solicitar numero para calculo
Console.WriteLine("Insira o numero que pretende calcular fatorial n! [0:12]");
numero = byte.Parse(Console.ReadLine());

// inicializar o fatorial
fatorial = 1;

// calculo do fatorial
for (uint i = 1; i <= numero; i++)
{
    fatorial *= i;
}

// apresentar resultado
Console.WriteLine("O fatorial de {0}!: {1}", numero.ToString(), fatorial.ToString());
Console.WriteLine("by Célio Carvalho.");

// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | FOR(…;…;…){…} (EXERCÍCIO)

ENUNCIADO

Desenvolva um programa capaz de gerar e apresentar ao utilizador 10 números inteiros aleatórios entre 0 e 100.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INST. REPETIÇÃO | FOR(…;…;…){…} (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// declarar variaveis
int numero;
Random random = new Random();

// preparar ecra
Console.Clear();
Console.WriteLine("=== GERADOR DE NUMEROS ALEATÓRIOS ===== \nby Célio Carvalho");
Console.WriteLine();

// gerar numeros aleatórios e mostrar ao utilizador
for (int i = 0; i < 10; i++)
{
    // gerar numero aleatorio
    numero = random.Next(0, 100);

    // mostrar numero gerado
    Console.WriteLine("{0}º gerado: {1}", (i+1).ToString("00"), numero.ToString());
}

// mostrar fim de geração
Console.WriteLine();
Console.WriteLine("== Fim");

// pausa
Console.ReadKey();
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO

ENUNCIADO

Há a necessidade de criar uma aplicação que, perante a introdução de dois limites inteiros, calcule a quantidade e o somatório de todos os valores ímpares nesse intervalo. Note que os limites do intervalo podem ser inseridos no menor para o maior ou do maior para o menor.

- Desenhe e implemente um algoritmo que resolva o problema usando cada um dos ciclos que aprendeu.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO

ENUNCIADO

Um número é primo se apenas for divisível por, exatamente, 4 números (2 positivos e 2 negativos). Por exemplo, o número 11 é primo porque apenas é divisível por {11, 1, -1, -11}. Já o número 21 não é primo porque é divisível por {21, 7, 3, 1, -1, -3, -7, -21}.

- Desenhe e implemente um algoritmo capaz de validar se um determinado numero inteiro inserido pelo utilizador é ou não primo.
- Evolua o algoritmo anterior de forma a indicar os divisores responsáveis pelo facto de um determinado numero não ser primo.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO

ENUNCIADO

A secretaria de uma escola necessita de uma aplicação que permita calcular a média, nota máxima e mínima de uma turma à disciplina de Geometria. A aplicação deve receber o número de alunos existentes e solicitar as notas apenas para essa quantidade de estudantes.

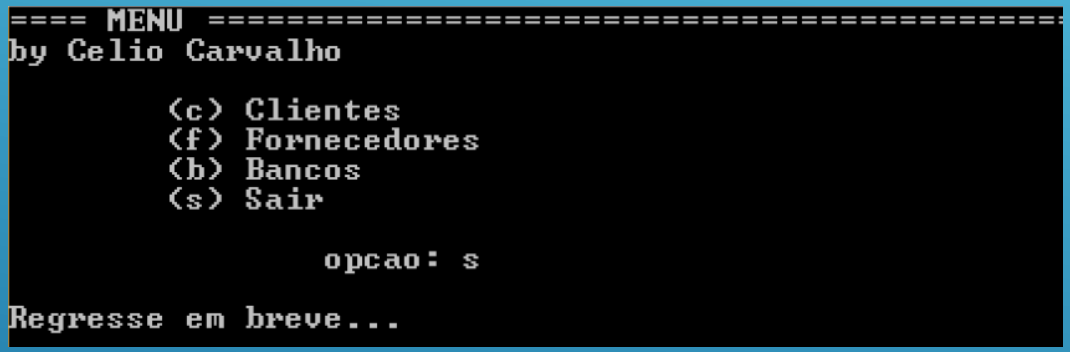
- Desenhe um algoritmo que resolva o problema e implemente-o de seguida.
- Evolua o algoritmo anterior e respetiva implementação, para que o número de alunos não tenha que ser comunicado num momento prévio ao processamento. Use como condição de paragem a inserção de uma nota acima dos 20 valores.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO

ENUNCIADO

São várias as formas como uma aplicação pode indicar ao utilizador as funcionalidades que oferece. Na imagem seguinte é apresentado um pequeno menu que permite ao utilizador acesso a funcionalidades de terceiros.



- Implemente uma solução que permita a navegação nas opções. A aplicação deve apenas terminar quando pressionada a letra "s" e, quando selecionada qualquer uma das restantes opções, deve apenas indicar uma linha especificando a opção escolhida.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

INSTRUÇÕES DE REPETIÇÃO

ENUNCIADO

Aproveitando o facto de um programa conseguir gerar números aleatórios, é possível criar um jogo para adivinhar um desses números. Não esqueça as dicas ao utilizador para saber se o número é maior ou menor.

- Desenhe o algoritmo do jogo e implemente-o.
- Evolua o algoritmo anterior no sentido de ter os seguintes 3 níveis de dificuldade:

NIVEL	INTERVALO	TENTATIVAS
1) Iniciante	[1:10]	3
2) Medio	[1:30]	10
3) Experiente	[1:50]	15

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | CONTEXTUALIZAÇÃO

A programação com recurso a sub-rotinas (pequena unidade de código) permite encapsular processamento e, desta forma, os programas ficam mais simples de desenvolver e de evoluir. Quanto mais independentes forem as sub-rotinas (módulos), mais atentamente o programador se pode concentrar em cada pequeno problema individual ignorando temporariamente os restantes.

Quando num determinado ponto de código se faz uma chamada a uma sub-rotina, o controlo do programa é transferido para essa sub-rotina, ou seja, as próximas instruções a serem executadas são as constantes nessa sub-rotina. Uma vez terminada a sua execução, o controlo do programa é devolvido ao bloco que havia solicitado a execução da rotina agora terminada e continua na instrução imediatamente a seguir.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | TIPOS

Na programação estruturada são normalmente referidas dois tipos de sub-rotinas:

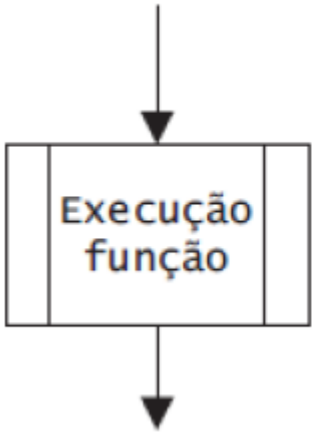
- Funções – são sub-rotinas que, depois de executadas, conseguem retornar um valor ao bloco de instruções que a chamou em primeiro lugar. Funcionam como uma caixa preta, que se desconhece o conteúdo, mas que tem a capacidade de receber valores (designados por parâmetros ou argumentos) e devolver no final um resultado. As funções em linguagem estruturada e nas linguagens de programação tem um funcionamento similar às funções matemáticas como o seno ou raiz quadrada que, perante a comunicação de um valor (argumento), retorna um outro (resultado)

- Procedimentos – são sub-rotinas muito parecidas com as funções, mas que não retornam qualquer valor quando terminam a execução, melhor dito, retornam um valor "nada". Apesar desta característica, conseguem receber argumentos como as funções.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | FUNÇÕES

Em C# tanto as funções como os procedimentos chamam-se apenas métodos, que são identificados através da sua assinatura. A assinatura é composta pelo nome do método (função ou procedimento) e pela sequência de tipos de dados aceites como argumentos (zero ou mais). É desta forma que se torna possível o *overload* de métodos, ou seja, a reescrita do mesmo método com várias implementações.




```
Função nomeFuncao(listaParametros)
inicio
...
nome ← <expressão>
...
fim-função
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | FUNÇÕES (EXERCÍCIO)

ENUNCIADO

Crie a função `media(n1,n2,n3)` que tenha a capacidade de calcular a media de 3 valores indicados. Consuma essa função num programa que, perante a inserção dos 3 valores pelo utilizador, indique a respetiva média.

```
// variaveis
int numero1, numero2, numero3;
double media;

// solicitar valores ao utilizador
Console.WriteLine("Calculo de media (função media)  ======\n");
Console.WriteLine("by Célio Carvalho");
Console.WriteLine();
Console.Write("Numero1: ");
numero1 = int.Parse(Console.ReadLine());
Console.Write("Numero2: ");
numero2 = int.Parse(Console.ReadLine());
Console.Write("Numero3: ");
numero3 = int.Parse(Console.ReadLine());

// efetuar o calculo
media = CalculaMedia(numero1, numero2, numero3);

// mostrar resultado
Console.WriteLine("Media: {0}", media.ToString());

// pausa
Console.ReadKey();
```

PROPOSTA DE SOLUÇÃO

```
static double CalculaMedia(int n1, int n2, int n3)
{
    // declaração variáveis (locais)
    double media;

    // calculo da media
    media = (n1 + n2 + n3) / 3.0;

    // devolver o valor da media
    return media;

}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | FUNÇÕES (EXERCÍCIO)

ENUNCIADO

É necessário criar uma aplicação que calcule a potência matemática através da inserção da base e do expoente. Como o cálculo da potência pode ser necessário de efetuar em diferentes pontos do programa, use uma sub-rotina na solução.

```
// variaveis
int _base, expoente;
uint resultado;

// solicitar valores ao utilizador
Console.WriteLine("Calculo da potencia  ======\n");
Console.WriteLine("by Célio Carvalho");
Console.WriteLine();
Console.Write("Base: ");
_base = int.Parse(Console.ReadLine());
Console.Write("Expoente: ");
expoente = int.Parse(Console.ReadLine());

// efetuar o calculo
resultado = CalculaPotencia(_base, expoente);

// mostrar resultado
Console.WriteLine("Resultado: {0}", resultado.ToString());

// pausa
Console.ReadKey();
```

PROPOSTA DE SOLUÇÃO

```
static uint CalculaPotencia(int _base, int expoente)
{
    // variaveis locais
    uint resultado = 1;

    // sucessão de multiplicações com acumulação
    for (int i = 0; i < expoente; i++)
    {
        resultado = (uint)(resultado * _base);
    }

    // devolver o resultado
    return resultado;
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | PROCEDIMENTOS

Um procedimento é em tudo parecido com a função com a grande diferença que não retorna qualquer valor à instrução de chamada (instrução no bloco de código que efetuou a chamada ao procedimento).

Como já dito na secção anterior (funções), em C# os procedimentos chamam-se também métodos. A forma de identificar é também composta pelo nome e a sequência de tipos de dados deduzidos dos argumentos recebidos.

Pode ver-se o procedimento como um caso especial de função, em que é retornado sempre um tipo muito especial - void que significa: nada, ausência de valor, nenhum valor ou vazio.


```
Procedimento nomeProcedimento(listaParametros)
...
blocos-de-instruções
...
Fim-procedimento
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | PROCEDIMENTOS (EXERCÍCIO)

ENUNCIADO

Crie uma aplicação que implemente um procedimento, que receba um inteiro e o escreva na consola com os algarismos invertidos.

```
// variaveis
int numero;

// solicitar valores ao utilizador
Console.WriteLine("Inverter numero (procedimento InverteNumero)\n");
Console.WriteLine("by Célio Carvalho");
Console.WriteLine();
Console.Write("Numero a inverter: ");
numero = int.Parse(Console.ReadLine());

// escrever o numero invertido na consola
MostraNumeroInvertido(numero);

// pausa
Console.ReadKey();
```

PROPOSTA DE SOLUÇÃO

```
static void MostraNumeroInvertido(int numero)
{
    // variaveis locais
    int algarismo;
    string numeroInvertido = "";

    // iterar enquanto o numero recebido for maior que zero
    while (numero > 0)
    {
        // o algarismo à direita do numero é achado através da divisão inteira por 10
        algarismo = numero % 10;

        // concatenar numeroInvertido
        numeroInvertido += algarismo.ToString();

        // truncar o algarimo à direita
        numero = (numero - algarismo) / 10;
    }

    // mostra no ecra
    Console.WriteLine("Numero invertido: {0}", numeroInvertido);
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | PROCEDIMENTOS (EXERCÍCIO)

ENUNCIADO

Crie uma aplicação que receba do utilizador um intervalo de inteiros e que escreva na consola se cada inteiro nesse intervalo é ou não primo. Crie uma função `primo(numero)` para avaliar se um dado numero é ou não primo e o procedimento `mostra(numero, primo)` para escrever na consola cada resultado processado.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | PROCEDIMENTOS (EXERCÍCIO)

PROPOSTA DE SOLUÇÃO

```
// variaveis
int limite1, limite2;
bool primo;

// pedir intervalo ao utilizador
Console.WriteLine("=== AVALIAÇÃO DE NUMEROS PRIMOS");
Console.WriteLine("by Célio Carvalho");
Console.Write("Limite inferior: ");
limite1 = int.Parse(Console.ReadLine());
Console.Write("Limite superior: ");
limite2 = int.Parse(Console.ReadLine());
Console.WriteLine("\n");

// inicia a mostragem de resultados
Console.WriteLine("NUMERO\tSIM\tNAO");

// percorrer todos os inteiros no intervalo
for (int i = limite1; i <= limite2; i++)
{
    primo = Primo(i);
    Mostra(i, primo);
}

// pausa
Console.ReadKey();
```

```
static bool Primo(int numero)
{
    // variáveis
    bool primo = true;

    // o 0 e 1 não são primos
    if (numero >= -1 && numero <= 1)
        return false;

    for (int i = 2; i < numero && primo == true; i++)
    {
        if (numero % i == 0)
            primo = false;

    }

    return primo;
}
```

```
static void Mostra(int numero, bool primo)
{
    // o que é enviado para o ecra devende se p número é ou não primo
    if (primo)
        Console.WriteLine(numero.ToString("00000#\t  X"));
    else
        Console.WriteLine(numero.ToString("00000#\t\t  X"));
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | PROCEDIMENTOS (EXERCÍCIO)

ENUNCIADO

Crie uma aplicação capaz de receber um número (inserido pelo utilizador) e de indicar na consola, qual o maior algarismo desse inteiro (exemplo: 367283, o maior algarismo é 8).

- Desenhe e implemente um algoritmo que resolva o problema, implementando a função (método em C#) `MaiorAlgarismo(numero)` que determina o maior dos algarismos do numero inteiro recebido. Salientar que deverá ser a aplicação principal a escrever na consola e não a função.

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | PROCEDIMENTOS (EXERCÍCIO)

ENUNCIADO

O cálculo do valor de um desconto é uma funcionalidade usada em vários programas e até várias vezes pela mesma aplicação.

- Crie um algoritmo para a função (método em C#) `CalculaDesconto(valor,taxa)` que devolva o valor do desconto de um determinado valor perante uma determinada taxa.

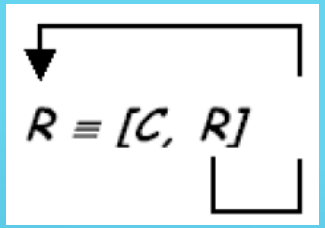
- Sugira também um algoritmo capaz de calcular o valor liquido a pagar `CalculaValorLiquido(valor,taxaDesconto)` que devolve o valor a pagar abatido do desconto obtido pela taxa de desconto. Faça reuso de código através do consumo da função criada na alínea anterior.

- Crie o procedimento `MostraTotais(valor,taxaDesconto,taxaIva)` que, através do reuso dos métodos anteriores, escreva no ecrã o resumo de uma venda conforme imagem abaixo. Considere implementar outras sub-rotinas que considere necessárias.

```
== CALCULAR DESCONTO ======
valor: 236,21
taxa de desconto: 2,87
taxa de iva: 23

==== TOTAIS DA VENDA ======
Valor inicial  :    236,21
Valor do desc. :     6,78
TOTAL com desc.:    229,43
Valor do Iva   :     52,77
TOTAL COM IVA  :    282,20
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | RECURSIVIDADE



- Um algoritmo que para resolver um problema divide-o em subprogramas mais simples, cujas soluções requerem a aplicação dele mesmo, é chamado recursivo.

- Um exemplo clássico é calculo do fatorial F(n) ou n!
  - n! = 1 se n = 0
  - n! = 1 se n = 1
  - n! = n . (n-1)! se n > 1

0! = 1
1! = 1
2! = 2 × 1
3! = 3 × 2 × 1
4! = 4 × 3 × 2 × 1
5! = 5 × 4 × 3 × 2 × 1
6! = 6 × 5 × 4 × 3 × 2 × 1
7! = 7 × 6 × 5 × 4 × 3 × 2 × 1
8! = 8 × 7 × 6 × 5 × 4 × 3 × 2 × 1
9! = 9 × 8 × 7 × 6 × 5 × 4 × 3 × 2 × 1
10! = 10 × 9 × 8 × 7 × 6 × 5 × 4 × 3 × 2 × 1

- Este é um exemplo de recursividade, porque o fatorial de um número pode ser calculado através do fatorial do seu antecessor.

- Exemplos:
  - 4! = 4 . (4-1)!
  - 3! = 3 . (3-1)!

- 6! = 6 x 5 x 4 x 3 x 2 x 1 = 720

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | RECURSIVIDADE

- Um algoritmo recursivo pode ter um ou mais casos de base e um ou mais casos gerais. E para que o algoritmo termine, as chamadas recursivas devem convergir em direção ao caso de base, senão o algoritmo não terminará jamais. Convergir significa ter uma parte menor do problema para ser resolvido.

```
F(4) = 4.F(4-1)
            F(3) = 3.F(3-1)
                        F(2) = 2.F(2-1)
                                    F(1) = 1.F(1-1)
                                                F(0) = 1 ---------- Caso Base
                                    F(1) = 1.1
                        F(2) = 2.1
            F(3) = 3.2
F(4) = 4.6
```

- Vantagens:
  - Simplifica a solução de alguns problemas;
  - O código, nestes casos, fica mais pequeno e fácil de ler;

- Desvantagens:
  - Os erros de implementação são típicos, e pode conduzir a erros de *stack overflow* (condição de paragem mal definida);

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

SUB-ROTINAS | RECURSIVIDADE

- O código seguinte apresenta a implementação recursiva do método **`Fatorial()`**.

```
Console.WriteLine($"Factorial(6) = { Factorial(6) }");
Console.WriteLine("End");


static int Factorial(int n)
{
    if (n == 0 || n == 1)
        return 1;

    return n * Factorial(n - 1);
}
```


Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

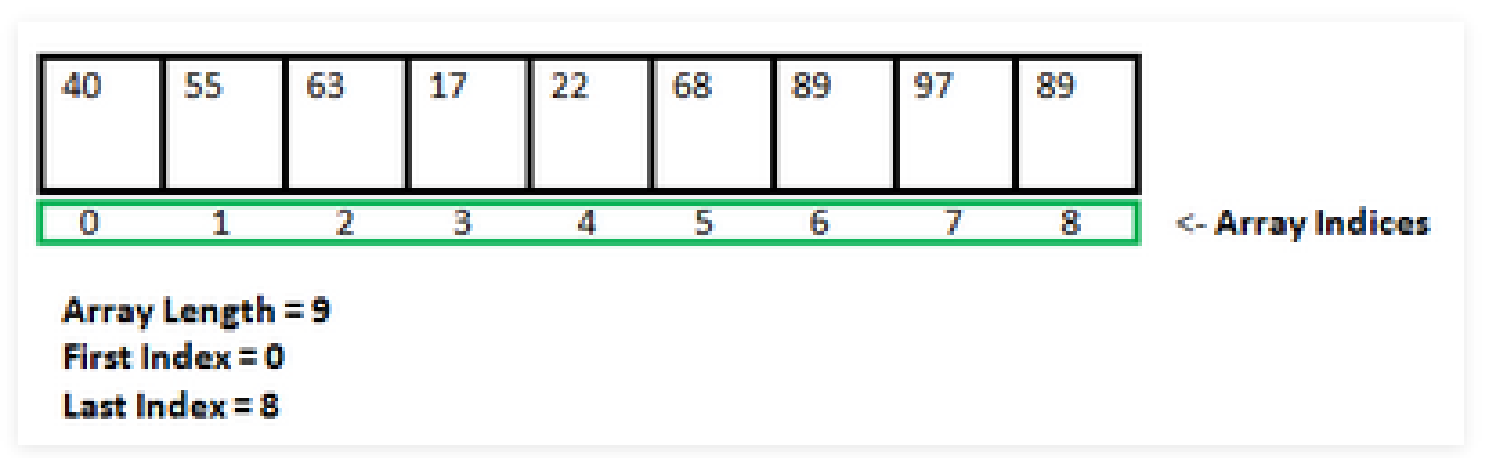
ESTRUTURAS ESTÁTICAS
VS.
ESTRUTURAS DINÂMICAS

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

ESTRUTURAS ESTÁTICAS VS. ESTRUTURAS DINÂMICAS

ESTRUTURAS ESTÁTICAS

- O tamanho é fixo (estático), i.e., o espaço de memória reservado é fixo (não é possível aumentar nem diminuir após a declaração da variável);
- Todos os elementos do conjunto são do mesmo tipo;
- Exemplo de estrutura estática: `array`.



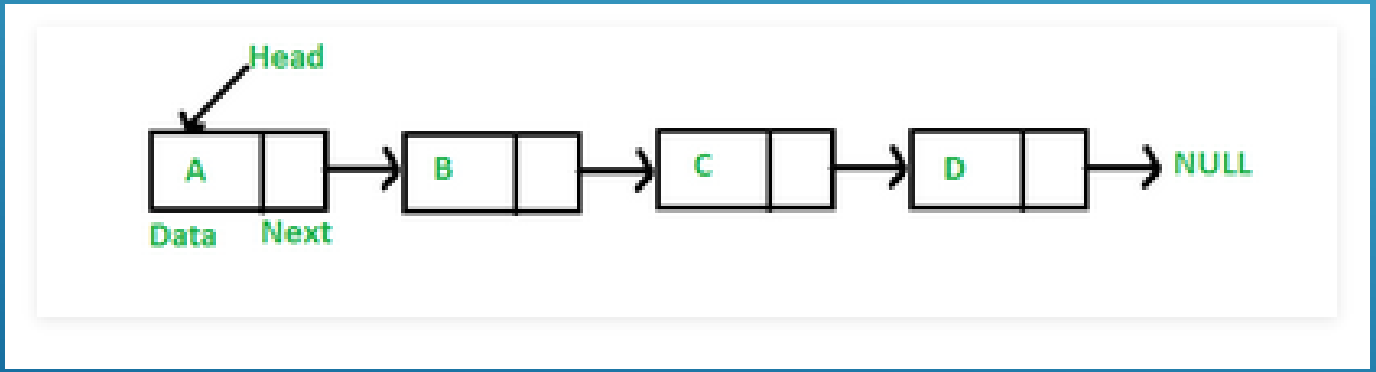
https://www.geeksforgeeks.org/static-data-structure-vs-dynamic-data-structure/

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga

ESTRUTURAS ESTÁTICAS VS. ESTRUTURAS DINÂMICAS

ESTRUTURAS DINÂMICAS

- O tamanho não é fixo (é dinâmico). A memória vai sendo reservada, à medida que cada elemento é inserido no conjunto, e libertada à medida que cada elemento é retirado do conjunto;
- Os elementos do conjunto não têm necessariamente de ser do mesmo tipo, i.e., podem conter estrutura diferente;
- Exemplo de estrutura dinâmica: Lista ligada.



https://www.geeksforgeeks.org/static-data-structure-vs-dynamic-data-structure/

Célio Carvalho
cdcarvalho@ipca.pt
UpSkill - PowerPlatform, IPCA Braga