Centre de Réadaptation de Mulhouse

Traduire son algorithme en C#

Vous trouvez ci-dessous des aides pour traduire votre pseudo code en code C#. Vous êtes libre d'enrichir votre document au fur et à mesure de vos découverte.

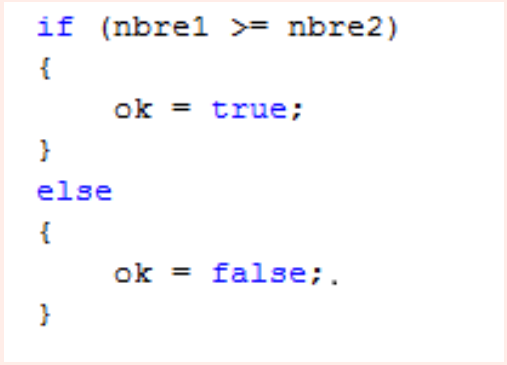
Du pseudo code à C#

Pseudo code	C#
entier nb1,nb2 chaine ch1 réel r double d booleen b	int nb1,nb2; string ch1; float r; double d; bool b;
chaine a lire a	string a; a = Console.ReadLine();
entier b; lire b	int b; b = int.Parse(Console.ReadLine());
ecrire c	Console.WriteLine(c);
//commentaire	//commentaire

Du pseudo code À C#: Instructions conditionnelles

Pseudo code	C#

si condition
alors
action(s) vraie(s)
sinon
action(s) fausse(s)
finsi
	if (condition)
	{
	action(s) vraie(s);
	}
	else
	{
	action(s) fausse(s);
	}

Ex : Si nbre1 supérieur ou égal nbre2 Alors ok <-vrai Sinon ok<-faux Fin si	

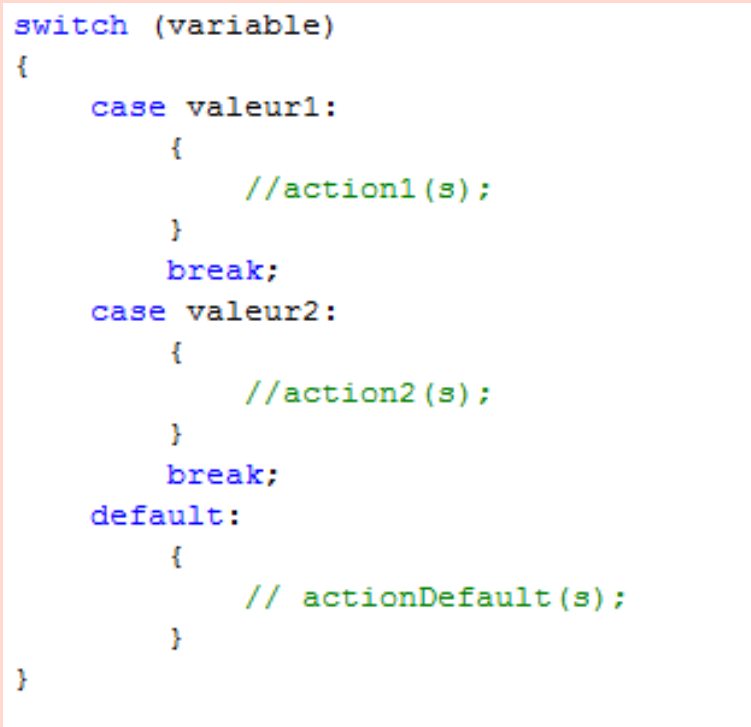
```
if (nbre1 >= nbre2)
{
    ok = true;
}
else
{
    ok = false;.
}
```

Du pseudo code à C#: Instructions conditionnelles

Pseudo code	C#

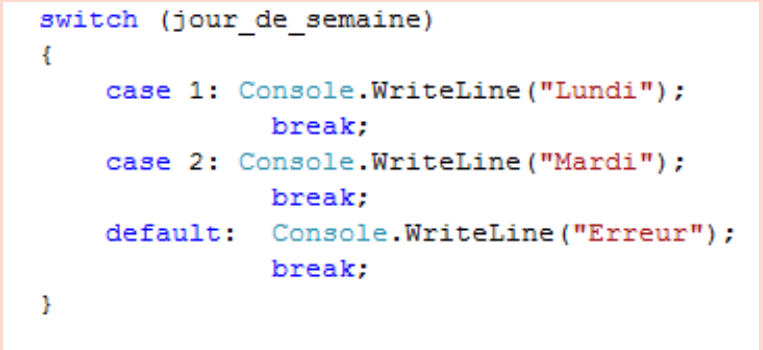
Ex : si A inférieur à 8
alors
ok <-faux
sinon
si A inférieur à 10
alors
ok<-faux
sinon
si A supérieur à 12
alors
ok<-faux
sinon
ok<-true
ecrire ok
finsi
finsi
finsi
	`if (A < 8)`
	`{`
	`ok = false;`
	`}`
	`else`
	`{`
	`if (A < 10)`
	`{`
	`ok = false;`
	`}`
	`else`
	`{`
	`if (A > 12)`
	`{`
	`ok = false;`
	`}`
	`else`
	`{`
	`ok = true;`
	`Console.WriteLine(ok.ToString());`
	`}`
	`}`
	`}`

Du pseudo code À C#: Instructions conditionnelles

Pseudo code	C#
selon variable choix valeur1 : //action1(s) choix valeur2 : //action2(s) .... choix valeurn : //actionn(s) défaut : //actionDéfauts(s) finselon	

```
switch (variable)
{
    case valeur1:
        {
            //action1(s);
        }
        break;
    case valeur2:
        {
            //action2(s);
        }
        break;
    default:
        {
            // actionDefault(s);
        }
}
```

Du pseudo code À C#: Instructions conditionnelles

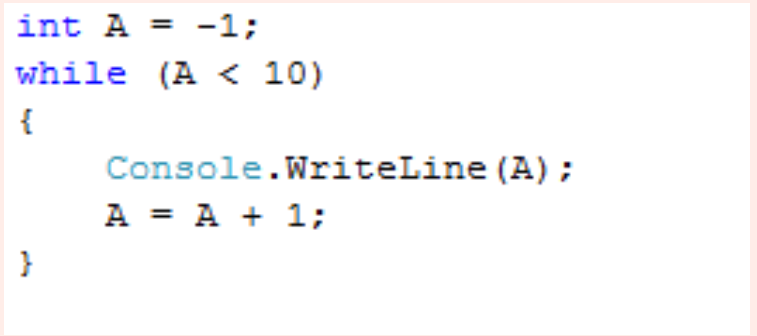
Pseudo code	C#
Ex : selon jour_de_semaine choix 1 : écrire « lundi » choix 2 : écrire « mardi » …… défaut : écrire « Erreur ! » finselon	

```
switch (jour_de_semaine)
{
    case 1: Console.WriteLine("Lundi");
            break;
    case 2: Console.WriteLine("Mardi");
            break;
    default:  Console.WriteLine("Erreur");
            break;
}
```

Du pseudo code À C#: Boucles conditionnelles

Pseudo code	C#

Tant que condition Faire
Action(s)
Fin tant que
	while (condition)
	{
	Action(s)
	}

Ex : Entier A <-1 Tant que A inférieure à 10 Faire Ecrire A A<-A+1 Fin tant que	

```
int A = -1;
while (A < 10)
{
    Console.WriteLine(A);
    A = A + 1;
}
```

Du pseudo code à C#: Boucles conditionnelles

Pseudo code	C#

pour *valeur*. **de** *valeurInitial* **à** *valeurFinal* **par** *pas*
faire
groupe d'opérations
finpour
	for (instructions_départ ; condition :instructions_fin_boucle)
	{
	groupe d'opérations ;
	}

pour i de 0 à 10 pas 1
faire
écrire i
finpour
	`for(int i=0;i<10;i++)`
	`{`
	`Console.Write(i);`
	`}`

Du pseudo code à C#: Boucles conditionnelles

Pseudo code	C#

répéter
  Action(s)
jusqu'à (*condition*)
	do
	{
	  Action(s)
	}while (condition)

Ex :
entier A<-1
faire
  ecrire A
  A<-A+1
tantque A inférieure à 10
	`int A = -1;`
	`do`
	`{`
	`  Console.WriteLine(A);`
	`  A = A + 1;`
	`}`
	`while (A < 10);`

Du pseudo code À C#: Instruction d'affectation et Opérateurs logiques

Pseudo code	C#

A<-valeur;
 	
A<-5
	`A = valeur;`
	
	`A = 5;`

bool A,B
 	
A ou B
A et B
A et non B
 	
si A ou b sont vraies
alors
finsi
	`bool A, B;`
	
	`A || B;`
	`A && B;`
	`A && !B;`
	
	`if (A || B)`
	`{`
	`}`

Du pseudo code à C#: Opérateurs binaires

Pseudo code	C#

entier A,B
A ou binaire B
A et binaire B
	`int A, B;`
	
	`A | B;`
	`A & B;`

entier a,b,c
a<-2
b<-10
C<-a ou binaire b
	`int a, b, c;`
	`a = 2;`
	`b = 10;`
	`c = a | b;`
	`//le resultat est c=10`

+,-,diviser,multiplier,modulo	+,-,/,*,%

Du pseudo code à C#: Opérateur ternaire

Pseudo code	C#
Si condition Alors action(s) vraie(s) Sinon action(s) fausse(s) Fin si	(condition) ? { action(s) vraie(s); } : { action(s) fausse(s); }
Ex : Si nbre1 supérieur ou égal nbre2 Alors valeur <-15 Sinon valeur<-16 Fin si	`valeur = (nbre1 >= nbre2) ? true : false;`
+,-,diviser,multiplier,modulo	+,-,/,*,%

Du pseudo code à C#. Opération de casting et pieges!

Pseudo code	C#
reel r =(reel)a	float c = (float)d;

Attention : La division d'un entier par un entier est un entier en C#	Ex :
	int a = 3, b = 5;
	float c;
	c = ((float)a / b);

Du pseudo code à C#: Déclaration de tableaux statiques à 1 dimension

Pseudo code	C#
tableau nom_tableau(taille) de type	type[] nom_tableau=new type[taille] ;

Ex :
tableau nbStagF(5) d' entier
//instanciation
nbStagF(0)<-0
nbStagF(1)<-0
nbStagF(2)<-0
nbStagF(3)<-0
nbStagF(4)<-0
	//déclaration et instanciation
	//1ere possibilite
	int[] nbStagF = new int[5];
	
	nbStagF[0] = 0;
	nbStagF[1] = 0;
	nbStagF[2] = 0;
	nbStagF[3] = 0;
	nbStagF[4] = 0;
	
	//déclaration et instanciation
	//2ere possibilite (CONTRACTE)
	int[] nbStagF = new int[] { 0,0,0,0,0};

entier B = nbStagF(4)	int B = nbStagF[4];
entier lg = | nbStagF |	int lg = nbStagF.Length;

Du pseudo code à C#: Déclaration de tableaux statiques à 2 dimensions

Pseudo code	C#
tableau nom_tableau(n,m) de type	Type[,] Tableau=new Type[n,m];

Ex :
tableau tab2Dim(5,2) d' entier
//instanciation
tab2Dim (0,0)<-0
tab2Dim (1,0)<-0
tab2Dim (2,0)<-0
tab2Dim (3,0)<-0
tab2Dim (4,0)<-0
tab2Dim (0,0)<-0
tab2Dim (1,0)<-0
tab2Dim (2,0)<-0
tab2Dim (3,0)<-0
tab2Dim (4,0)<-0
	`//déclaration et instanciation`
	`//1ere possibilite`
	`int[,] tab2Dim = new int[5,2];`
	
	`tab2Dim[0,0] = 0;`
	`tab2Dim[1,0] = 1;`
	`tab2Dim[2,0] = 2;`
	`tab2Dim[3,0] = 3;`
	`tab2Dim[4,0] = 4;`
	`tab2Dim[0,1] = 5;`
	`tab2Dim[1,1] = 6;`
	`tab2Dim[2,1] = 7;`
	`tab2Dim[3,1] = 8;`
	`tab2Dim[4,1] = 9;`
	
	`//déclaration et instanciation`
	`//2ere possibilite (CONTRACTE)`
	`int[,] tab2Dim = new int[,] {`
	`{0,5},`
	`{1,6},`
	`{2,7},`
	`{3,8},`
	`{4,9}};`

entier caseNum1 <- tab2Dim (0,0)	int caseNum1 = tab2Dim[0,0];
entier nbLigne = | tab2Dim | 0 entier nbColonne = | tab2Dim |1	int nbLigne = tab2Dim.GetLength(0); int nbColonne = tab2Dim.GetLength(1);

Du pseudo code à C#: Déclarations de procédure : (Une procédure ne retourne rien)

Pseudo code

PROCEDURE *nom_procédure* (**VAL** *type nom_arg1*, …, **VAR** *type nom_argn*,…)
  Déclarations des variables locales
  Actions
FIN PROCEDURE

C#

```
void  nom_procedure( [type] arg1, ref [type] arg1,..)
{
  //Déclarations des variables locales
  //Actions
}
Ex:
void Modifier(int nb1, ref int nb2)
{
 nb1 = 15;
 nb2 = 20;
}
```

//VAL par valeur
Le passage par valeur ne modifie pas la valeur de la variable passé en paramètre

//VAR par référence
Le passage par référence modifie la valeur de la variable passé en paramètre

Du pseudo code à C#: Déclarations de procédure : (Une procédure ne retourne rien)

Pseudo code

PROCEDURE *nom_procédure* (**VAL** *type nom_arg1*, …, **VAR** *type nom_argn*,…)
  Déclarations des variables locales
  Actions
FIN PROCEDURE

C#

```
void  Nom_procedure( [type] arg1, ref [type] arg1,..)
{
  //Déclarations des variables locales
  //Actions
}
Ex:
void Modifier(int nb1, ref int nb2)
{
 nb1 = 15;
 nb2 = 20;
}
```

//VAL par valeur
Le passage par valeur ne modifie pas la valeur de la variable passé en paramètre

//VAR par référence
Le passage par référence modifie la valeur de la variable passé en paramètre

Du pseudo code à C#: Déclaration d'une fonction: (Une fonction retourne toujours une et une seule valeur)

Pseudo code

type **FONCTION** *nom_fonction* (**VAL** *type nom_arg1*, …, **VAR** *type nom_argn*,…)
//Déclarations des variables locales
//Actions
RETOURNE valeur
FIN FONCTION

C#

```
[type_retour] Nom_fonction ([type] arg1, ref [type] arg2 …)
{
    [type_retour] variableARetourner ;
    //Déclarations des variables locales
    //Actions
    return variableARetourner;
}
```

//VAL par valeur
Le passage par valeur ne modifie pas la valeur de la variable passé en paramètre

//VAR par référence
Le passage par référence modifie la valeur de la variable passé en paramètre

Du pseudo code à C#: Déclaration d'une fonction: (Une fonction retourne toujours une et une seule valeur)

Exemple: Pseudo code

```
Réel  Fonction Somme ( VAL réel nb1, VAR réel nb2)
        réel somme
        somme<-nb1+nb2
        retourne somme
fin fonction
```

Exemple: C#

```
float Somme(float nb1,ref float nb2)
{
float somme;
 somme = nb1 + nb2;
 return somme;  //valeur de retour
}
```

Du pseudo code À C#: function

Pseudo code	C#
a **mod** b	a%b
chaine prénom **|prénom|**	string prénom prénom.Length
prénom$_{2<--4}$	prénom.Substring(2,3)
x^y	Math.Pow(x,y)
‘tutu’	@"tutu"

A VOUS DE CONTINUER!

Pseudo code	C#
…	…
s	