Classe de Temps - VB.NET
Classe de gestion du temps en VB.NET

Dans le cadre d'un développement mettant en scène des temps en secondes, il est utile de pouvoir obtenir celui-ci en quelque chose de plus facile à manipuler et à afficher. Celle-ci a été utilisée dans le développement de l'application de visualisation des informations système d'une machine :

- **DotNetSysInfo**

Introduction

Dans le cadre du développement de l'application de visualisation de l'état d'une machine via page WEB :

- **DotNetSysInfo**

Il m'a été nécessaire de pouvoir disposer d'une classe permettant la manipulation et l'affichage de temps (en particulier pour l'affichage de l'Uptime de la machine en question).

Présentation

Ainsi donc après de nombreuses recherches je suis tombé sur une classe qui m'a servi de modèle qui a été écrite en Java dans le cadre d'un cours disponible sur Internet :

- **Classe de Temps Modèle en Java (FR)**
- **Cours entier disponible (FR)**

Ainsi à partir de ce modèle qui était très intéressant, je me suis lancé dans sa traduction en VB.NET et de ce fait, je vous mets à disposition le résultat de ce travail.

Classe VB.NET : TimeClass

Je vous met dans cette page la version sans commentaires afin de ne pas surcharger la page, vous avez un lien à la fin de l'article qui vous permettra de télécharger la version entièrement documenté (adapté à NDoc).

```
Public Class TimeClass

   Private _Secondes As Integer
   Private _Minutes As Integer
   Private _Heures As Integer
   Private _Jours As Integer

#Region "Liste Propriétés de la Classe"
   Public Property Secondes() As Integer
      Get
         Return _Secondes
      End Get
      Set(ByVal Value As Integer)
         _Secondes = Value
      End Set
   End Property
   Public Property Minutes() As Integer
      Get
         Return _Minutes
      End Get
      Set(ByVal Value As Integer)
         _Minutes = Value
      End Set
   End Property
   Public Property Heures () As Integer
      Get
         Return _Heures
      End Get
      Set(ByVal Value As Integer)
         _Heures = Value
      End Set
   End Property
   Public Property Jours() As Integer
      Get
         Return _Jours
      End Get
      Set(ByVal Value As Integer)
         _Jours = Value
      End Set
   End Property
#End Region

   ' ---------------------------------------------------------------------------
   Public Sub New()
  ' Constructeur par Défaut de la classe
      _Secondes = 0
      _Minutes = 0
      _Heures = 0
      _Jours = 0
   End Sub
   ' ---------------------------------------------------------------------------
   Public Sub New(ByVal NbJours As Integer, ByVal NbHeures As Integer, _
   ByVal NbMinutes As Integer, ByVal NbSecondes As Integer)
  ' Constructeur transmettant la valeur de chaque parametre de temps (jour, heure, minute et seconde)
      _Secondes = NbSecondes
      _Minutes = NbMinutes
      _Heures = NbHeures
      _Jours = NbJours
   End Sub

   ' ---------------------------------------------------------------------------
   Public Sub New(ByVal TotalNbSecondes As Long)
  ' Constructeur transmettant le nombre total de secondes
      Dim LeTemps As New TimeClass
      LeTemps.ConversionDetaillee(TotalNbSecondes)
      Secondes = LeTemps.Secondes
      Minutes = LeTemps.Minutes
      Heures = LeTemps.Heures
      Jours = LeTemps.Jours
   End Sub
```

```
    ' ---------------------------------------------------------------------
    Public Sub ConversionDetaillee(ByVal NBSec As Long)
   ' Procédure permettant de calculer chaque variable à partir du Nombre total de secondes
        Jours = NBSec \ 86400
        Heures = (NBSec - (Jours * 86400)) \ 3600
        Minutes = (NBSec - (Heures * 3600) - (Jours * 86400)) \ 60
        Secondes = NBSec - (Minutes * 60) - (Heures * 3600) - (Jours * 86400)
    End Sub

    ' ---------------------------------------------------------------------
    Public Function ConversionSeconde() As Long
   ' Fonction permettant de connaître le nombre total de seconde à partir des valeurs des variables
        Return CType(((((Jours * 24) + Heures) * 60 + Minutes) * 60) + Secondes, Long)
    End Function

    ' ---------------------------------------------------------------------
    Public Function Egal(ByVal ValeurAComparer As TimeClass) As Boolean
   ' Fonction de test d'égalité de 2 objets TimeClass
        Dim ValeurBase As Long = ConversionSeconde()
        Dim ValeurComp As Long = ValeurAComparer.ConversionSeconde
        If ValeurBase = ValeurComp Then
            Return True
        Else
            Return False
        End If
    End Function

   ' ---------------------------------------------------------------------
    Public Function DifferenceA(ByVal ValeurAComparer As TimeClass) As Integer
   ' Fonction qui renvoie la différence entre 2 objets TimeClass en secondes
        Dim ValeurBase As Long = ConversionSeconde()
        Dim ValeurComp As Long = ValeurAComparer.ConversionSeconde
        Return CType(ValeurBase - ValeurComp, Integer)
    End Function

    ' ---------------------------------------------------------------------
    Public Function AjouterTemps(ByVal ValAAjouter As TimeClass) As TimeClass
   ' Fonction qui ajoute un objet TimeClass à un autre par la somme des 2 totaux de secondes
        Dim ValeurBase As Long = ConversionSeconde()
        Dim ValeurComp As Long = ValAAjouter.ConversionSeconde
        Dim TemptResult As New TimeClass(ValeurBase + ValeurComp)
        Return TemptResult
    End Function

    ' ---------------------------------------------------------------------
    Public Function ReturnChaineTemps() As String
   ' Fonction qui renvoie la chaine permettant de définir textuellement la valeur d'un objet TimeClass
        Dim TempChaine As String = ""
        If Jours > 1 Then
            TempChaine &= Jours & " Jours "
        Else
            TempChaine &= Jours & " Jour "
        End If
        If Heures > 1 Then
            TempChaine &= Heures & " Heures "
        Else
            TempChaine &= Heures & " Heure "
        End If
        If Minutes > 1 Then
            TempChaine &= Minutes & " Minutes "
        Else
            TempChaine &= Minutes & " Minute "
        End If
        If Secondes > 1 Then
            TempChaine &= Secondes & " Secondes "
        Else
            TempChaine &= Secondes & " Seconde "
        End If
        Return TempChaine
    End Function

    ' ---------------------------------------------------------------------
End Class
```

Téléchargement

Ce script est la traduction enrichie de l'exemple fourni dans le cours dont le lien est plus haut. Il est traduit en VB.NET, mais peut l'être aussi très aisément en C# ou même J# (la base étant en Java).

Vous trouverez la version commentée selon la norme NDoc à l'adresse suivante :

- **Télécharger le fichier VB contenant la classe commentée.**

Conclusion

Cette source est très simple à comprendre et peut facilement adaptée et enrichie suivant vos besoins. Je vous invite d'ailleurs à consulter la liste des autres exemples disponibles dans le cours d'origine, car ceux-ci peuvent aussi être très utiles.

- **Cours complet avec les exemples en Java**

En vous souhaitant de bons projets de développement.

Romelard Fabrice (alias F___)