

But de ce TP :

- Travailler avec des variables
- Effectuer des calculs sur des variables numériques
- Placer et redimensionner des contrôles
- Savoir utiliser graphiquement les propriétés des contrôles

Connaissances requises :

- Savoir déclarer des variables à une dimension
- Savoir utiliser des opérateurs arithmétiques simples
- Savoir utiliser l'interface de développement

A effectuer obligatoirement :

- Créer une application Windows Forms
- Nommer l'application : TP1_<votre_nom>
- Commenter les parties de code (ne pas tout commenter non plus)
- Utiliser les variables pour ne pas travailler sur le contenu des contrôles

Objet de votre application :

- Créer un convertisseur de prix HT/TTC sur la base d'un taux fixe de 19,6% (TVA Française)

Cahier des charges :

Phase 1

- Une fenêtre principale nommé Convertisseur HT/TTC avec les paramètres suivants :
 - o MaximizeBox = false
 - o MinimizeBox = false
 - o StartPosition = CenterScreen
 - o Opacity = 80%
- Utiliser 3 **linklabel** à la place des Button avec comme texte :
 - o Passer du prix HT au prix TTC
 - o Passer du prix TTC au prix HT
 - o Remettre à zéro
- Utiliser 2 textbox dont le nom sera :
 - o ht
 - o ttc
- Utiliser deux label avec comme texte (les placer devant la bonne textbox) :
 - o Prix HT :
 - o Prix TTC :
- Initialiser deux variables **publiques** : prixht et prix TTC dont le type sera adapté aux calculs monétaires et affecter les bonnes valeurs lors de l'exécution du calcul.
- Quand l'utilisateur cliquera sur « Passer du prix HT au prix TTC », la variable prixht subira l'opération : $[\text{prixht}] * (1 + 0,196)$ et le résultat sera affiché dans la boîte de saisie ttc

- Quand l'utilisateur cliquera sur « Passer du prix TTC au prix HT », la variable `prixttc` subira l'opération : $[\text{prixttc}]/(1+0,196)$ et le résultat sera affiché dans la boîte de saisie ht.
- Quand l'utilisateur cliquera sur « Remettre à zéro », toutes les variables seront réinitialisées, ainsi que les boîtes de saisie.

Quand ceci sera terminé, appeler pour vérification (n'oubliez-pas de sauvegarder régulièrement et de commenter votre code).

Phase 2

- Remplacer automatiquement le point par une virgule dans les boîtes de saisie.
- Détecter si la case d'origine est vide. Si elle est vide, lui donner un fond de couleur rouge.
- Afficher le résultat sur fond vert.

Annexes (Rappel de syntaxe)

Déclaration de variable :

Syntaxe : `dim nom as type`

nom ne doit pas être réservé et est sensible à la casse.

type :

Boolean	Contient une valeur Booléenne (logique): True ou False.
Byte	Contient les nombres entiers de 0 à 255 (sans signe)
Short	Entier sur 16 bits (-32768 à 32768)
Integer	Entier sur 32 bits (-2147483648 à 2147483647)
Long	Entier sur 64 bits (-9223372036854775808 à 9223372036854775807)
Single	Nombre réel en virgule flottante (-1,401298 *10 ⁻⁴⁵ à 1,401298 10 ⁴⁵)
Double	Nombre réel en virgule flottante double précision. (..puissance 324)
Decimal	Nombre réel en virgule fixe grande précision sur 16 octets.
Char	1 caractère alphanumérique
String	Chaîne de caractère de longueur variable (jusqu'a 2 milliard de caractères)
DateTime	Date plus heure
Object	Peut contenir tous les types de variables mais aussi des contrôles, des fenêtres...

Structure Ensemble de différentes variables définies par l'utilisateur.

(tableau : <http://plasserre.developpez.com/v1-6.htm>)

Déclarer une variable publique (ou globale)

Syntaxe : **Public** nom **As** Type

Les variables publiques sont des variables que vous pouvez utiliser pour n'importe quel contrôle sans à avoir besoin de les déclarer plusieurs fois → Elles ont une portée globale. Il faut déclarer ces variables une seule fois sous **Public/Private Class** <nom de la form>

Exemple :

```
Public Class Form1
    Public variable1 As Double
    Public variable2 As Integer
```

Structure conditionnelle :

- Linéaire : **If** condition à vérifier **then** instruction **else** instruction
- En colonne :

```
if condition à vérifier then
instruction1
instruction2
...
else
instruction1
instruction2
...
end if
```

Rechercher dans une chaîne de caractères :

Syntaxe : n=**InStr**(string1, string2)

Retourne un entier spécifiant la position de début de la première chaîne à l'intérieur d'une autre.

Exemple :

```
n=InStr(1, "aaaaRaa", "R")           'retourne 5
```

Remplacer dans une chaîne de caractères :

Syntaxe :

```
Resultat=ChaîneDép.Replace(ChaîneARemplacer, ChaîneQuiRemplace)
```

Exemple :

```
Dim s As String= "Visual_Basic_.NET"
s= s.Replace("_", " ") 'Donne s= "Visual Basic .NET"
```

Commenter du code

Syntaxe : **'**commentaire