

# Cours de Programmation

## Travaux Pratiques 1

ITII PACA - A&M ParisTech 2010-2011



Ruding LOU  
ruding.lou@ensam.fr

### Découverte de l'environnement de développement Visual Studio &

### Programmation impérative en VB/Visual Studio

Tout au long des exercices proposés, prenez soin de lire et de comprendre les éventuelles erreurs de compilation ou d'exécution. Sauvez chaque exercice dans un projet (ou un module VB) différent, et familiarisez-vous avec les différentes fonctionnalités du logiciel.

#### TP 1.1

Ecrire le code déclarant

une variable nommée `monChiffre`, devant contenir un entier, initialisée à 10;

une variable nommée `monNom`, devant contenir une chaîne de caractères, initialisée à *toto*;

une variable nommée `verite`, devant contenir une valeur Booléenne, sans initialisation;

une variable nommée `tab`, devant contenir 10 chiffres réels, sans initialisation.

Créer une boucle qui répète à demander à l'utilisateur de saisir au clavier 10 chiffres et les stocke dans la variable `tab`.

Calculer la moyenne de ces dix chiffres stockés dans la variable `tab`.

Comparer cette moyenne par rapport à la variable `monChiffre`. Si la moyenne est plus grande on affecte *vrai* à la variable `verite` et on affiche **Bonjour toto, la moyenne xx est plus petite que 10**. Sinon on affecte *faux* à la variable `verite` et on affiche **Bonjour toto, la moyenne xx est plus grande que 10**.

#### TP 1.3

Implémenter une variation de l'exercice 3.3 en ajoutant une fonction. Ecrire un programme qui demande Cinq noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique. S'ils ne sont pas rangés alors afficher les noms dans l'ordre alphabétique.

#### TP 1.4

Implémenter les exercices 4.8 et 4.9 en VB.

#### TP 1.5

Implémenter l'exercice 5.6 en VB en ajoutant une fonctionnalité : on demande à l'utilisateur le nombre de chiffres de la suite.

Implémenter l'exercice 5.7.

#### TP 1.6

Implémenter un programme qui affiche le plus grand d'une suite d'entiers saisis au clavier. Les nombres saisis ne seront pas conservés en mémoire. La suite se termine avec la valeur 0. Vous utiliserez une structure de boucle.

## TP 1.7

Implémenter un programme qui réalise le jeu suivant :

L'ordinateur choisit un nombre aléatoire entre 1 et 100. L'utilisateur doit trouver ce nombre le plus rapidement possible. A chacune de ses réponses, l'ordinateur répond "Plus", "Moins" ou "Ok ! Vous avez mis X coups".

## TP 1.8

Pour les exercices précédents dans lesquelles on demande aux utilisateurs de saisir un chiffre, Que se passe-t-il si l'utilisateur entre une chaîne de caractères ? Comment éviter ces erreurs ? Réécrivez le programme afin qu'il ne puisse plus produire ces erreurs, (IsNumeric : tester si une variable porte une valeur numérique ; CInt : conversion d'une variable vers numérique)

## TP 1.9

Ecrire une procédure qui simule le comportement d'une calculatrice, c'est à dire qui exécute une boucle :

- (a) Lecture d'une ligne qui contient un entier, un opérateur et un entier (ex : 17 + 8). Les opérateurs sont +, -, \_, / et ^.
- (b) Calcul de la valeur de l'expression.
- (c) Impression du résultat à l'écran.

Outils de manipulation de chaîne de caractères:

String.Split, String.Length, String.Contains, IsNumeric, CInt,

L'exemple à tester:

```
Dim s As String = "Il est toto, il a 12 ans, Et toi?"
Dim sDecoupe As String()
Dim i As Integer
Dim age As Integer = 0

sDecoupe = s.Split(" ")
Console.WriteLine("Decoupe par espace")
For i = 0 To sDecoupe.Length() - 1 Step 1
    Console.WriteLine(sDecoupe(i))
    If IsNumeric(sDecoupe(i)) Then
        age = CInt(sDecoupe(i))
    End If
Next i

If age <> 0 Then
    Console.WriteLine("Detection d'age : " & age)
End If

If s.Contains(",") Then
    Console.WriteLine("Decoupe par virgule")
    sDecoupe = s.Split(",")
    For i = 0 To sDecoupe.Length() - 1 Step 1
        Console.WriteLine(sDecoupe(i))
    Next i
End If

If Not s.Contains(";") Then
    Console.WriteLine("phrase ne contient pas de ;")
End If
```

## TP 1.10

Réaliser une course de chevaux en console.

- On demande aux utilisateurs le nombre de chevaux pour la course.
- On définit une distance de parcours (ex. 70).
- On définit une vitesse maximale (ex. 3)
- On fait avancer itérativement tous les chevaux sur le parcours avec une distance entre 1 et vitesse maximale. La distance d'avancement à chaque itération est calculée aléatoirement.
- Les chevaux s'arrêtent sur le point final du parcours, on mémorise le nombre d'itérations pour y attendre pour chaque cheval.
- On calcule le classement de la course en sortant le nombre d'itérations.
- L'affichage d'avancement se fait avec des étoiles suivi par le numéro de cheval.

Outils possibles à utiliser : (Consulter la documentation en appuyant F1 pour savoir en détails)

- ☞ Calculer un nombre aléatoire `Randomize()` et `Rnd()`
- ☞ Faire attendre le system en millisecondes `System.Threading.Thread.Sleep(n)`
- ☞ Nettoyer la console `Console.Clear()`
- ☞ Sortir d'une boucle `Exit xxx` (ex. `For / Exit While ...`)
- ☞ Aller directement à une ligne de codes `GoTo()`

```
*****1
*****2
*****3
*****4
*****5
```

```
*****1
*****2
*****3
*****4
*****5
```

```
*****1
*****2
*****3
*****4
*****5
classement:
1 -> Cheval (1) temps: 35
2 -> Cheval (3) temps: 35
3 -> Cheval (2) temps: 37
4 -> Cheval (4) temps: 37
5 -> Cheval (0) temps: 40
Donner votre choix:
1. Recommencer le jeu avec même nombre de chevaux
2. Recommencer le jeu, changer le nombre de chevaux
Autre Chiffre. Terminer le jeu
```