Annexe – Cours Visual Basic

Types et opérateurs basiques

I) Type de base

1) Types

Visual Basic	Visual Basic.net	Taille mémoire	Plage de valeur
`Boolean`	`Boolean`	2 octets	`True` ou `False`
`Byte`	`Byte`	1 octet	0 à 255 (non signé)
`Char`	`Char`	2 octets	Caractère unicode
`Date`	`DateTime`	8 octets	Date/heure de 0:00:00 le 1/1/1 à 23:59:59 le 31/12/9999
`Short`	`Int16`	2 octets	-32 768 à 32 767
`Integer`	`Int32`	4 octets	-2 147 483 648 à 2 147 483 647
`Long`	`Int64`	8 octets	-9 223 372 036 854 775 808 à 9 223 372 036 854 775 807
`Single`	`Single`	4 octets	-3,402 823E + 38 à -1,401 298E – 45 pour les valeurs négatives; 1,401 298E – 45 à 3,402 823E + 38 pour les valeurs positives
`Double`	`Double`	8 octets	-1,797 693 134 862 31E + 308 à -4,940 656 458 412 47E – 324 pour les valeurs négatives; 4,940 656 458 412 47E – 324 à 1,797 693 134 862 31E + 308 pour les valeurs positives
`Decimal`	`Decimal`	16 octets	Valeur numérique utilisé pour un montant de 0 à +/- 79 228 162 514 264 337 593 543 950 335 sans décimale; 0 à +/- 7,922 816 251 426 433 759 354 395 033 5 avec 28 décimales; le plus petit nombre différent de zéro étant +/- 1E - 28
`String`	`String`	Variable	Chaîne de caractères de longueur variable (0 à environ 2 milliards de caractères unicodes)
`Object`	`Object`	Variable	Type global

Rappels :
Exemple de déclaration d'un type :

```
Dim unBoolean As Boolean = true
```

Pour déclarer un tableau :

```
'.Net
Dim semaine() As String = {"Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", _
"Samedi", "Dimanche"}
'Visual Basic
Dim semaine(6) As String
```

```
semaine(0) = "Lundi" '(…)
```

II)Opérateurs

1) Opérateurs Visual Basic et Visual Basic.net

Opérateurs par ordre décroissant de priorité			
Catégorie	Description	Opérateur	Exemple/application
Arithmétique	Puissance	^	2^3 = 8
	Moins unaire	-	Inversion du signe
	Multiplication/Division	*, /	2*3 = 6, 4/2 = 2
	Division entière	\	10\3 = 3
	Modulo	Mod	10 Mod 3 = 1, 9 Mod 3 = 0
	Addition/Soustraction	+, -	2+1 = 3, 2-1 = 1
	Concaténation*	&	1 & 2 = 12
Comparaison**	Egalité	=	1 = 2/2
	Différence	<>	1 <> 10
	Inférieur à/Supérieur à	<, >	1 < 2, 2 > 1
	Inférieur ou égal à/Supérieur ou égal à	<=, >=	1 <= 2, 2 >= 1
	Pattern (voir ci-dessous)	Like	Dim patternEx As String = "B?t" "Bat" Like patternEx "Bot" Like patternEx Les deux renverront True
	Référence	Is	Dim ob1 = monObjet Dim ob2 = monObjet Dim ob3 = new Object() ob1 is ob2 ' == true ob2 is ob3 ' == false
	Type	TypeOf ... Is	Dim ob1 = new Object() If TypeOf ob1 is Object 'True
Logique	Négation	Not	Dim a As Boolean = True Dim b As Boolean = Not a '=False
	Et logique	And, AndAlso	a=true And b=true And _ c=true

	Ou logique	`Or`, `OrElse`	`a=true Or b=true Or _ c=true`
	Ou exclusif	`Xor`	Xor | 0 | 1 0 | 0 | 1 1 | 1 | 0

*L'opérateur de concaténation est valable pour les `String` et les nombres.
**Les opérateurs de comparaison ont tous la même priorité
**Spécificité de la plateforme .net, il est désormais possible d'utiliser `AndAlso` et `OrElse`, ils permettent de bloquer l'exécution si une condition précédente n'est pas vérifiée (`AndAlso`) ou si elle est vérifiée (`OrElse`)
Ils permettent ainsi un gain d'exécution notamment dans les branchement Si...Alors utilisant plusieurs comparaisons simultanées.

Comparaison de pattern

L'opérateur de comparaison de pattern permet de tester si une chaine de caractères (`String`) correspond à un pattern prédéfinit.

Liste des caractères spécifiques :

Caractères	Signification
`?`	Un caractère quelconque
`#`	Un caractère numérique (0 à 9 inclus)
`[]`	Une liste de caractère autorisée entre crochet
`[!]`	Une liste de caractère à exclure entre crochet
`*`	Zéro ou plusieurs caractères
`-`	L'opérateur tiret permet de spécifier une plage, exemple `a-z` correspond aux caractères de `a` à `z`

D'une manière générale, il s'agit ici simplement d'un test d'expression régulière.

Voici quelques exemples :
`abc*` → toutes les chaines de caractères commençant par abc.
`abc?` → toutes les chaines de caractères commençant par abc et ayant un seul caractère après
`abc[a-d]` → `abca` ou `abcb` ou `abcc` ou `abcd`
`abc[!a-d]` → toutes les chaines de caractères ayant 4 caractères, dont les 3 premiers commencent par `abc` et le dernier n'étant pas dans la plage `a-d`.

2) Opérateurs Visual Basic.net

Visual Basic.net se rapproche de C/C++ en incluant les opérateurs d'affectations, voici la liste :

Exemple	Expression équivalente	Résultat pour x=4
`x += 1`	`x = x + 1`	x=5
`x -= 1`	`x = x - 1`	x=3
`x *= 2`	`x = x * 2`	x=8
`x /= 2`	`x = x / 2`	x=2
`x \= 3`	`x = x \ 3`	x=1
`x ^= 2`	`x = x ^ 2`	x=16