

Module 3:

Utilisation des variables de type valeur

Vue d'ensemble

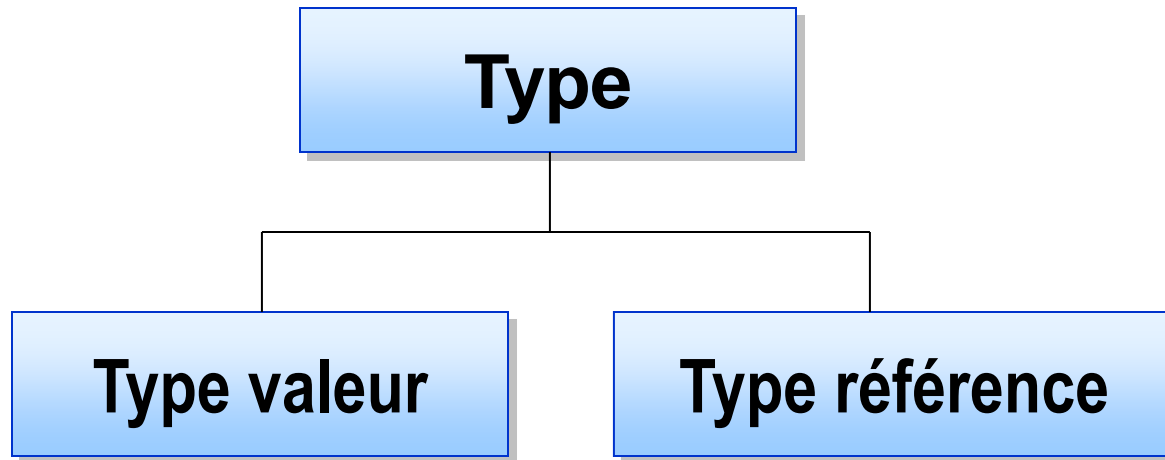
- Common Type system
- Nommer les variables
- Utilisation des types de données pré-définis
- Création de types de données définis par l'utilisateur
- Conversion des types de données

Common Type System

- Vue d'ensemble des CTS (Common Type System)
- Comparaison types valeur et référence
- Comparaison des types valeur du framework et ceux définis par l'utilisateur.
- Types simples

Vue d'ensemble des CTS

- CTS prend en charge les types à la fois de valeur et de référence



Comparaison Types valeur et référence

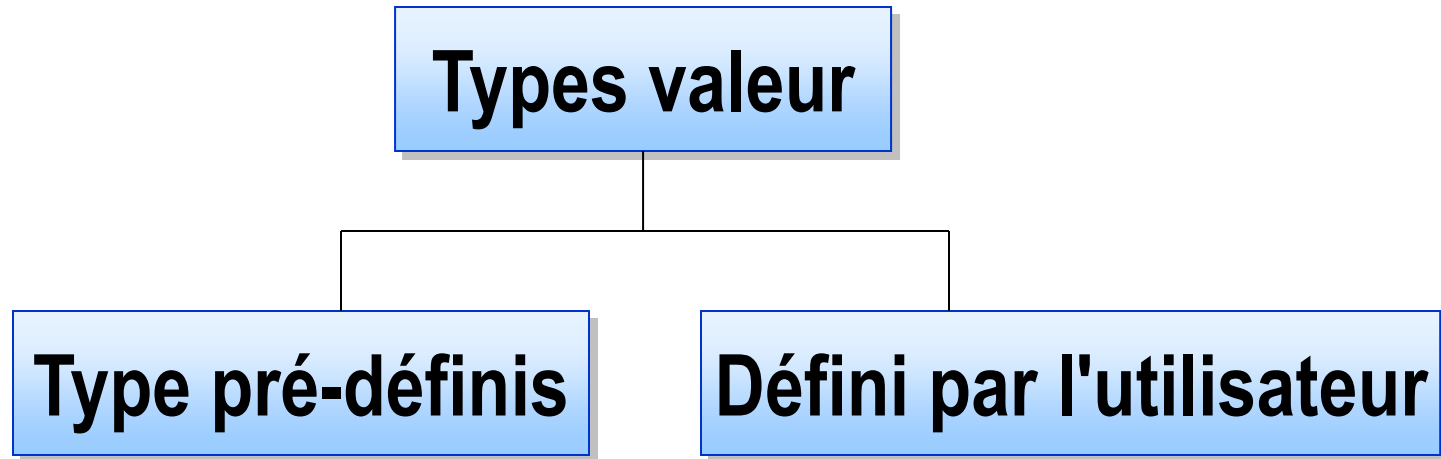
■ Les types de valeur:

- Contiennent directement leurs données
- Chacun a sa propre copie des données
- Les opérations sur l'un ne peuvent pas affecter un autre

■ Les types de référence:

- Contiennent des références vers leurs données (appelés objets)
- Deux variables de référence peuvent faire référence à un même objet
- Opérations sur l'un peuvent affecter un autre

Comparaison types intégrés et types valeur définis par l'utilisateur



- Des exemples de types de valeurs pré-définies:

- int
- float

- Des exemples de types de valeurs définies par l'utilisateur:

- enum
- struct

Types simples

- Identifiés par des mots-clés réservés
 - int / / mot-clé réservés
 - Ou -
 - **System.Int32**

Règles et recommandations pour nommer les variables

- Règles

- Utilisez des lettres, le trait de soulignement et chiffres

Answer42
42Answer



- Recommandations

- Évitez d'utiliser les majuscules
- Évitez de commencer par un trait de soulignement
- Évitez d'utiliser des abréviations
- Utilisez PascalCasing dans les noms composés de plusieurs mots (classes, méthodes, propriétés ...)
- Utilisez notationCamel (variables ...)

autre
Autre



BADSTYLE
_poorstyle
BestStyle



Msg
Message



Mots clés C

- Mots-clés réservés aux identifiants

```
abstract, base, bool, default, if, finally
```

- Ne pas utiliser des mots clés comme noms de variables
 - Résultats d'une erreur de compilation
- Évitez d'utiliser des mots-clés en modifiant leur sensibilité à la casse

```
int INT; // mauvais style
```

Quiz: Pouvez-vous trouver les noms de variables interdits / déconseillés, pourquoi?

1

```
int 12count;
```

2

```
char $diskPrice;
```

3

```
char middleInitial;
```

4

```
float this;
```

5

```
int __identifier;
```

Utilisation des types de données pré-définies

- Déclaration de variables locales
- Affectation de valeurs aux variables
- Affectation composée
- Opérateurs principaux
- Incrémentation et décrémentation
- Priorité des opérateurs

Déclaration de variables locales

- Déclarée par le type de données et le nom de la variable:

```
int itemCount;
```

- Possible de déclarer plusieurs variables dans une déclaration:

```
int itemCount, employeeNumber;
```

- Ou -

```
int itemCount,  
    employeeNumber;
```

Attribution de valeurs aux variables

- Attribuer des valeurs à des variables qui sont déjà déclarés:

```
int employeeNumber;  
employeeNumber = 23;
```

- Initialiser une variable lors de sa déclaration:

```
int employeeNumber = 23;
```

- Vous pouvez également initialiser les valeurs de caractères:

```
char MiddleInitial = 'J';
```

Affectation composée

- L'ajout d'une valeur à une variable est très fréquent

```
itemCount = itemCount + 40;
```

- Il est possible de raccourcir cette expression de cette façon :

```
itemCount += 40;
```

- Ce raccourci fonctionne pour tous les opérateurs arithmétiques

```
itemCount -= 24;
```

Opérateurs communes

Opérateurs communes	Exemple
<ul style="list-style-type: none">• Opérateurs d'égalité• Les opérateurs relationnels• Les opérateurs conditionnels• Opérateur d'incrément• Opérateur de décrémentation• Les opérateurs arithmétiques• Les opérateurs d'affectation	<p>== !=</p> <p><> <=> = is</p> <p>&& ?:</p> <p>++</p> <p>--</p> <p>+ - * / %</p> <p>= *= /= %= += -= < <= > >= &= ^= =</p>

Augmentation et de diminution

- Augmenter ou diminuer le contenu d'une variable de 1 est très utilise.

```
itemCount += 1;  
itemCount -= 1;
```

- Il est possible de raccourcir cette expression comme ceci :

```
itemCount ++;  
itemCount --;
```

- Ce raccourci existe sous deux formes

```
++ItemCount;  
--ItemCount;
```


Priorité des opérateurs

- Priorité des opérateurs et associativité
 - Sauf pour les opérateurs d'affectation, tous les opérateurs binaires sont associatifs à gauche (*la partie gauche est prioritaire sur la partie droite*).
 - Les opérateurs d'affectation et les opérateurs conditionnels sont associatifs à droite (*la partie droite est prioritaire sur la partie gauche*).

Création de types de données définis par l'utilisateur

- Type énumération
- Structure Type

Types énumération

- Définition d'un type énumération

```
enum Color { Red, Green, Blue }
```

- L'utilisation d'un type énumération

```
Color colorPalette = Color.Red;
```

- Affichage d'une variable d'énumération

```
Console.WriteLine("{0}", colorPalette); // Affiche Red
```

Types Structure

- Définition d'un type de structure

```
public struct Employee
{
    public string firstName;
    public int age;
}
```

- L'utilisation d'un type de structure

```
Employee companyEmployee;
companyEmployee.firstName = "Joe";
companyEmployee.age = 23;
```

Conversion des types de données

- Conversion implicite de type de données
- Conversion explicite de type de données

Conversion implicite de type de données

- Pour convertir un int en long:

```
using System;
class Test
{
    static void Main( )
    {
        int intValue = 123;
        long longValue = intValue;
        Console.WriteLine("(long) {0} = {1}", intValue,
        ↪ longValue);
    }
}
```

- Les conversions implicites ne peuvent pas rater mais il est possible de perdre en précision

Explicit conversion de type de données

- Pour faire des conversions explicites, utiliser une expression de cast:

```
using System;
class Test
{
    static void Main( )
    {
        long longValue = Int64.MaxValue;
        int intValue = (int)longValue;
        Console.WriteLine("(int) {0} = {1}", longValue,
        ↪ intValue);
    }
}
```

Lab 3.1: Création et utilisation Types

