

Chapitre 5 : Passage de paramètres à une fonction

Nous nous intéressons ici au mode de passage des paramètres d'une fonction. Considérons la fonction statique suivante :

```
private static void ChangeInt(int a)
{
    a = 30;
    Console.WriteLine("Paramètre formel a=" + a);
}
```

Dans la définition de la fonction, *a* est appelé un paramètre formel. Il n'est là que pour les besoins de la définition de la fonction *changeInt*. Il aurait tout aussi bien pu s'appeler *b*.

Considérons maintenant une utilisation de cette fonction :

```
static void Main(string[] args)
{
    int age = 20;
    ChangeInt(age);
    Console.WriteLine("Paramètre effectif age=" + age);
}
```

Ici dans l'instruction *ChangeInt(age)*, *age* est le paramètre effectif qui va transmettre sa valeur au paramètre formel *a*.

Nous nous intéressons à la façon dont un paramètre formel récupère la valeur d'un paramètre effectif.

5.1 Passage par valeur

L'exemple suivant nous montre que les paramètres d'une fonction sont par défaut passés par valeur, c'est à dire que la valeur du paramètre effectif est copiée dans le paramètre formel correspondant. On a deux entités distinctes. Si la fonction modifie le paramètre formel, le paramètre effectif n'est lui en rien modifié.

```
using System;
namespace Chapitre5
{
    class Program
    {
        private static void ChangeInt(int a)
        {
            a = 30;
            Console.WriteLine("Paramètre formel a=" + a);
        }
        static void Main(string[] args)
        {
            int age = 20;
            ChangeInt(age);
            Console.WriteLine("Paramètre effectif age=" + age);
        }
    }
}
```

Les résultats obtenus sont les suivants :

1. Paramètre formel $a=30$
2. Paramètre effectif $age=20$

La valeur 20 du paramètre effectif age a été recopiée dans le paramètre formel a . Celui-ci a été ensuite modifié.

Le paramètre effectif est lui resté inchangé. Ce mode de passage convient aux paramètres d'entrée d'une fonction.

5.2 Passage par référence

Dans un passage par référence, le paramètre effectif et le paramètre formel sont une seule et même entité. Si la fonction modifie le paramètre formel, le paramètre effectif est lui aussi modifié. En C#, ils doivent être tous deux précédés du mot clé **ref** :

Voici un exemple :

```
using System;
namespace Chapitre5
{
    class Program
    {
        private static void ChangeInt2(int a2)
        {
            A2 = 30;
            Console.WriteLine("Paramètre formel a2=" + a2);
        }
        static void Main(string[] args)
        {
            int age2 = 20;
            ChangeInt2(ref age2);
            Console.WriteLine("Paramètre effectif age2=" + age2);
        }
    }
}
```

et les résultats d'exécution :

1. Paramètre formel $a2=30$
2. Paramètre effectif $age2=30$

Le paramètre effectif a suivi la modification du paramètre formel. Ce mode de passage convient aux paramètres de sortie d'une fonction.

5.3 Passage par référence avec le mot clé out

Considérons l'exemple précédent dans lequel la variable $age2$ ne serait pas initialisée avant l'appel à la fonction $changeInt2$:

```
using System;
namespace Chapitre5
{
    class Program
    {
        private static void ChangeInt2(ref int a2)
        {
```

```

        a2 = 30;
        Console.WriteLine("Paramètre formel a2=" + a2);
    }
    static void Main(string[] args)
    {
        int age2;
        ChangeInt2(ref age2);
        Console.WriteLine("Paramètre effectif age2=" + age2);
    }
}

```

Lorsqu'on compile ce programme, on a une erreur :
Use of unassigned local variable 'age2'

On peut contourner l'obstacle en affectant une valeur initiale à *age2*. On peut aussi remplacer le mot clé **ref** par le mot clé **out**.

On exprime alors que le paramètre est uniquement un paramètre de sortie et n'a donc pas besoin de valeur initiale :

```

using System;
namespace Chapitre5
{
    class Program
    {
        private static void ChangeInt3(out int a3)
        {
            a3 = 30;
            Console.WriteLine("Paramètre formel a3=" + a3);
        }
        static void Main(string[] args)
        {
            int age3;
            ChangeInt3(out age3);
            Console.WriteLine("Paramètre effectif age3=" + age3);
        }
    }
}

```

Les résultats de l'exécution sont les suivants :

1. Paramètre formel a3=30
2. Paramètre effectif age3=30