

Fiche savoirs sur les fonctions

Définition :

- Une fonction est un bloc de code isolé, qui peut être appelé et donc réutilisé plusieurs fois dans le code de son programme.
- l'utilisation des fonctions permet d'avoir un code organisé en modules.

Utiliser une fonction pour éviter les répétitions de code

Voici l'exemple du programme **NombreCache** avec des contrôles de la saisie de l'essai (try catch) qui sont utilisés à plusieurs endroits dans le code de l'application et qui se ressemblent :

- le contrôle de la saisie avant la boucle (la première saisie)
- et le contrôle de la saisie dans la boucle:

[fonction.cs](#)

```
static void Main(string[] args) {  
    // déclaration  
    int valeur = 0, essai = 0, nbre = 1;  
    bool correct = false;  
    // saisie du nombre à chercher  
    while (!correct) {  
        try {  
            Console.WriteLine("Entrez le nombre à chercher = ");  
            valeur = int.Parse(Console.ReadLine());  
            correct = true;  
        }  
        catch {  
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");  
        }  
    }  
    Console.Clear();  
    // saisie du premier essai  
    correct = false;  
    while (!correct) {  
        try {  
            Console.WriteLine("Entrez un essai = ");  
            essai = int.Parse(Console.ReadLine());  
            correct = true;  
        }  
        catch {  
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");  
        }  
    }  
}
```

```
// boucle sur les essais
while (essai != valeur) {
    // test de l'essai par rapport à la valeur à chercher
    if (essai > valeur) {
        Console.WriteLine(" --> trop grand !");
    }
    else {
        Console.WriteLine(" --> trop petit !");
    }
    // saisie d'un nouvel essai
    correct = false;
    while (!correct) {
        try {
            Console.Write("Entrez un essai = ");
            essai = int.Parse(Console.ReadLine());
            correct = true;
        }
        catch {
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");
        }
    }
    // compteur d'essais
    nbre++;
}
// valeur trouvée
Console.WriteLine("Vous avez trouvé en "+nbre+" fois !");
Console.ReadLine();
}
```

Le premier objectif est d'éviter la répétition à l'identique de plusieurs lignes de code en créant une **fonction appelée saisie()** en dehors du **Main()**, juste au-dessus, pour isoler le bloc de code identique :

```
static void saisie()
{
    correct = false;
    while (!correct)
    {
        try
        {
            Console.Write("Entrez un essai = ");
            essai = int.Parse(Console.ReadLine());
            correct = true;
        }
        catch
        {
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");
        }
    }
}
```

```
}  
}
```



Problème :

- les variables **correct** et **essai** ne sont pas déclarées dans la fonction **saisie()**.
- il faut rajouter leur déclaration

La variable **correct** n'est utilisée que dans la fonction **saisie()** : on peut la déclarer directement dans la fonction **saisie()** :

```
static void saisie()  
{  
    bool correct = false;  
    while (!correct)  
    {  
        try  
        {  
            Console.Write("Entrez un essai = ");  
            essai = int.Parse(Console.ReadLine());  
            correct = true;  
        }  
        catch  
        {  
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");  
        }  
    }  
}
```

Par contre la variable **essai** doit pouvoir être réutilisée dans le **Main()**. Il est nécessaire que **essai** soit déclarée pour être visible et utilisable à la fois dans le **Main()** et dans la fonction **saisie()**. Il est préférable que **essai** soit déclarée comme une **variable globale** à l'ensemble du programme.

Déclaration de la variable globale essai :

```
class Program  
{  
    static int essai;  
    ...  
}
```

Essai n'est plus déclarée dans le Main() :

```
// déclaration  
int valeur = 0, nbre = 1;  
essai = 0;
```

Programme du nombre caché avec la procédure saisie

```
static int essai;

static void Main(string[] args) {
    // déclaration
    int valeur = 0, nbre = 1;
    bool correct = false;
    // saisie du nombre à chercher
    while (!correct) {
        try {
            Console.WriteLine("Entrez le nombre à chercher = ");
            valeur = int.Parse(Console.ReadLine());
            correct = true;
        }
        catch {
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");
        }
    }
    Console.Clear();
    // saisie du premier essai
    saisie();
    // boucle sur les essais
    while (essai != valeur) {
        // test de l'essai par rapport à la valeur à chercher
        if (essai > valeur) {
            Console.WriteLine(" --> trop grand !");
        }
        else {
            Console.WriteLine(" --> trop petit !");
        }
        // saisie d'un nouvel essai
        saisie();
        // compteur d'essais
        nbre++;
    }
    // valeur trouvée
    Console.WriteLine("Vous avez trouvé en "+nbre+" fois !");
    Console.ReadLine();
}
```

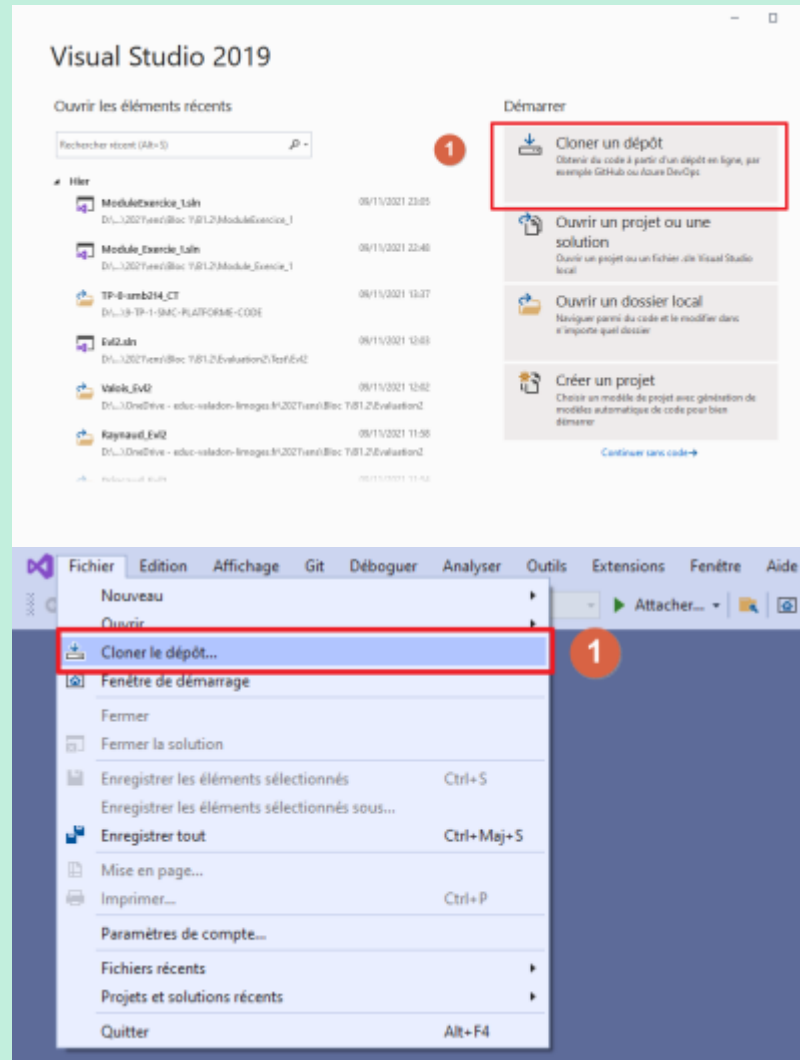
Exercice 1 :



- Créez un nouveau projet à partir de ce dépôt Git : https://github.com/ctecher/ModuleExercice_1.git
- Modifiez Le programme doit permettre de saisir plusieurs prix et au final d'afficher le total des prix.
- Avant chaque saisie, il faut poser la question "Avez-vous un prix à saisir ? (O/N) "

et saisir la réponse de préférence d'un seul caractère sans valider, en contrôlant que l'utilisateur a bien saisi O ou N et pas autre chose (sinon la question est reposée).

- Si l'utilisateur répond N, le total des prix est affiché et le programme s'arrête.
- Sachant qu'une saisie doit être prévue avant la boucle générale, et une à la fin de la boucle, le but est d'isoler ce bloc de code dans une fonction.



Utiliser une fonction qui renvoie un résultat (une valeur)

Il est possible de ne pas utiliser la variable globale en transformant la fonction pour qu'elle **renvoie** (**retourne**) au programme appelant, la valeur saisie.

Il faut supprimer la déclaration de la variable globale **essai** et remettre sa déclaration dans le **Main()**, comme c'était avant :

```
//static int essai; à supprimer
...
static void Main(string[] args)
```

```
{  
// déclaration  
int valeur = 0, essai = 0, nbre = 1;
```

Il est nécessaire de déclarer la variable **essai** dans la fonction **saisie()**.

Cette déclaration va se faire :

- en changeant le nom de la variable pour l'appeler **nombre** et en l'initialisant en début de fonction ;
- en modifiant la déclaration de la fonction : **void** est devenu **int** car la fonction va **retourner** une information de type **int** ;
- en rajoutant à la fin de la fonction la ligne **return nombre;** qui permet de **retourner** le contenu de la variable **nombre** :

```
static int saisie()  
{  
    int nombre = 0;  
    bool correct = false;  
    while (!correct)  
    {  
        try  
        {  
            Console.Write("Entrez un essai = ");  
            nombre = int.Parse(Console.ReadLine());  
            correct = true;  
        }  
        catch  
        {  
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");  
        }  
    }  
    return nombre;  
}
```

Pour récupérer le retour de la fonction **saisie()** dans le **Main()** il suffit d'affecter l'appel de la fonction, dans une variable.

Dans le **Main()**, l'appel de la fonction **saisie()** doit donc être affecté à la variable **essai** pour récupérer dans **essai** l'information saisie. Voici au final le **Main()** :

```
static void Main(string[] args) {  
    // déclaration  
    int valeur = 0, essai = 0, nbre = 1;  
    bool correct = false;  
    // saisie du nombre à chercher  
    while (!correct) {  
        try {  
            Console.Write("Entrez le nombre à chercher = ");  
            valeur = int.Parse(Console.ReadLine());  
            correct = true;  
        }
```

```
}
catch {
    Console.WriteLine("Erreur de saisie : saisissez un nombre entier");
}
}
Console.Clear();
// saisie du premier essai
essai = saisie();
// boucle sur les essais
while (essai != valeur) {
    // test de l'essai par rapport à la valeur à chercher
    if (essai > valeur) {
        Console.WriteLine(" --> trop grand !");
    }
    else {
        Console.WriteLine(" --> trop petit !");
    }
    // saisie d'un nouvel essai
    essai = saisie();
    // compteur d'essais
    nbre++;
}
// valeur trouvée
Console.WriteLine("Vous avez trouvé en "+nbre+" fois !");
Console.ReadLine();
}
```



Exercice 2 :

Reprenez le programme de l'exercice 1 et modifiez-le de cette manière afin de renvoyer un résultat.

Rendre la fonction paramétrable

Il reste maintenant au début du programme un bloc de code, très similaire à celui copié dans le module **saisie()** :

- c'est celui qui concerne la saisie de la valeur à chercher ;
- et la différence est au niveau du **message** qui doit s'afficher.

Voici le bloc de code concerné :

```
static void Main(string[] args)
{
    ...
    bool correct = false;
    // saisie du nombre à chercher
```

```
while (!correct) {
    try {
        Console.Write("Entrez le nombre à chercher = "); // message qui est
différent
        valeur = int.Parse(Console.ReadLine());
        correct = true;
    }
    ...}
```

Il est possible :

- d'ajouter des **paramètres** à une fonction pour lui envoyer une information ;
- en **ajoutant le paramètre** dans l'entête du module ;
- et ce paramètre est utilisable dans le module comme une **variable locale** :

```
static int saisie(string message) // -> paramètre avec son type qui peut
recevoir une valeur
{
    int nombre = 0;
    bool correct = false;
    while (!correct)
    {
        try
        {
            Console.Write(message + " = "); // -> utilisation du paramètre comme
une variable locale
            nombre = int.Parse(Console.ReadLine());
            correct = true;
        }
        catch
        {
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");
        }
    }
    return nombre;
}
```

Maintenant, nous pouvons dans le **Main()**, modifier les 3 appels à la fonction **saisie()** pour envoyer une valeur au paramètre.

Ajout du paramètre lors de l'appel de la fonction saisie

```
static void Main(string[] args)
{
    // déclaration
    int valeur = 0, essai = 0, nbre = 1;
    // plus de déclaration de la variable correct à faire //bool correct =
false;
    // saisie du nombre à chercher
```



```
    valeur = saisie("Entrez le nombre à chercher"); // -> 1er appel de la
fonction saisie() avec un message en paramètre
    Console.Clear();
    // saisie du premier essai
    essai = saisie("Entrez un essai"); // -> 2ème appel de la fonction
saisie() avec un message en paramètre
    // boucle sur les essais
    while (essai != valeur)
    {
        // test de l'essai par rapport à la valeur à chercher
        if (essai > valeur) {
            Console.WriteLine(" --> trop grand !");
        }
        else {
            Console.WriteLine(" --> trop petit !");
        }
        // saisie d'un nouvel essai
        essai = saisie("Entrez un essai"); // -> 3ème appel de la fonction
saisie() avec un message en paramètre
        // compteur d'essais
        nbre++;
    }
    // valeur trouvée
    Console.WriteLine("Vous avez trouvé en "+nbre+" fois !");
    Console.ReadLine();
}
```

- Les messages d'information **Entrez le nombre à chercher** et **Entrez un essai** sont maintenant transférés dans le paramètre **message** selon les besoins.

La fonction est paramétrée et le code de la fonction **Main()** est nettement plus court.



Dépôt GitHub : https://github.com/ctecher/FonctionC_cor.git

Correction

Correction

```
using System;
class Program
{
    static int saisie(string message)
    {
        bool correct = false;
        int chiffre = 0;
        while (!correct)
        {
```

```
        try
        {
            Console.Write(message);
            chiffre = int.Parse(Console.ReadLine());
            correct = true;
        }
        catch
        {
            Console.WriteLine("Erreur de saisie : saisissez un nombre entier");
        }
    }
    return chiffre;
}

public static void Main(string[] args)
{
    // déclaration
    int valeur = 0, essai = 0, nbre = 1;

    // saisie du nombre à chercher
    valeur = saisie("Entrez le nombre à chercher = ");

    Console.Clear();
    // saisie du premier essai
    essai = saisie("Entrez un essai = ");

    // boucle sur les essais
    while (essai != valeur)
    {
        // test de l'essai par rapport à la valeur à chercher
        if (essai > valeur)
        {
            Console.WriteLine(" --> trop grand !");
        }
        else
        {
            Console.WriteLine(" --> trop petit !");
        }
        // saisie d'un nouvel essai
        essai = saisie("Entrez un essai = ");
        // compteur d'essais
        nbre++;
    }
    // valeur trouvée
    Console.WriteLine("Vous avez trouvé en " + nbre + " fois !");
    Console.ReadLine();
}
```

Aller plus loin avec plusieurs paramètres

Il est possible de mettre plusieurs paramètres à une fonction.



Exemple :

```
static int saisie(string message, int borneMin, int borneMax)
```

From:

<https://siocours.lycees.nouvelle-aquitaine.pro/> - **Les cours du BTS SIO**

Permanent link:

<https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/bloc1/fichesavoirsfonction1>

Last update: **2023/11/08 12:51**

