الطرق (الدوال أو التوابع) Methods:

الطريقة هي جزء من البرنامج تقوم بأداء مهمة معينة وتعيد نتيجة ما، ويمكن أن تُمرر لها بارمترات من البرنامج المستدعي.

الهدف الأساسي من استخدام الطرق هو منع تكرار كتابة التعليمات البرمجية لأكثر من مرة، مما يؤدي إلى تقليل أسطر البرنامج، وبالتالي سهولة تتبع الأخطاء في حال حدوثها في البرنامج، هذا بالإضافة إلى أن الطرق تجعل البرنامج منظم بشكل أفضل.

يمكن استخدام الطريقة بعد التصريح عنها، عن طريق استدعاء هذه الطريقة في البرنامج أو في دالة أخرى، وذلك بذكر اسم الطريقة وتمرير قيم مناسبة للبارمترات الخاصة بهذه الطريقة.
عند استدعاء طريقة ما، ينتقل سير البرنامج إلى تلك الطريقة ويتم تنفيذ التعليمات الموجودة ضمنها، ثم تتم العودة إلى البرنامج الذي قام بالاستدعاء.

التصريح عن الطريقة (Method): يتم التصريح عن الطريقة أو الدالة بالشكل التالي:

```
accessModifier returnType methodName(Type Parameter1, Type Parameter2 ,…..)
{
    التصريح عن المتغيرات المحلية الخاصة بالطريقة
    .....
    تعليمات الطريقة
    .....
    return …….
}
```

accessModifier : محدد الوصول لتلك الطريقة.
returnType : نوع القيمة التي ستعيدها الطريقة، اذا كانت الطريقة تعيد قيم.
methodName : اسم الطريقة.
Type parameter1 : نوع واسم الوسيط الاول من وسطاء الطريقة.

Type parameter2 : نوع واسم الوسيط الثاني من وسطاء الطريقة.

.....

ملاحظات:

١. يجب أن يتطابق نوع القيمة التي ستتم إعادتها في تعليمة return مع النوع المحدد في القيمة المعادة returnType.

٢. في حال كانت الطريقة لا تعيد قيمة عندها تكون القيمة المعادة returnType هي **void**، وعندها لا نستخدم تعليمة return .

٣. يجب أن يتطابق عدد ونوع البارمترات المستخدمة عند التصريح عن الطريقة مع عدد ونوع البارمترات التي سيتم تمريرها إلى هذه الطريقة عند استدعائها.

المتغيرات المحلية (local) والمتغيرات العامة (global):

المتغيرات المحلية: هي المتغيرات التي يتم الإعلان عنها داخل الطريقة، وتسمى بهذا الاسم لأنها تكون غير معروفة خارج الطريقة المعرفة ضمنها، ومن الجدير بالذكر أن المتغير المحلي تنشأ له قيمة وكيان عند ابتداء تنفيذ الطريقة التي ينتمي إليها فقط، وتختفي عند الانتهاء من تنفيذ تلك الطريقة.

المتغيرات العامة: هي التي تكون قيمتها معروفة من أول البرنامج إلى آخره، ويمكن استعمالها في أي جزء منه، وتحتفظ بقيمتها أثناء تنفيذ البرنامج.

مثال: طريقة لجمع عددين صحيحين.

```
static int sum(int number1, int number2)
{
    int total = number1 + number2;
    return total;
}
```

وفي الـ main يتم استدعاء الطريقة بكتابة اسمها وتمرير قيم مناسبة للبارمترات.

```
int result = sum(5, 8);
Console.Write(result);
Console.ReadKey();
```

نلاحظ في هذا المثال أن المتغير total هو متغير محلي (local) ضمن الدالة sum وبالتالي لا نستطيع استخدامه خارج الطريقة.

مثال: اكتب طريقة تقوم بطباعة العبارة " Hello , in C# " على الشاشة :

```
static void print_hello()
 {
    Console.writeLine("Hello, in c#");
 }
Static void Main (string [ ] args)
  {
    Print_hello();
  }
```

نلاحظ من المثال السابق أن الأقواس ضرورية عند التصريح عن الطريقة وكذلك عند استدعائها حتى وان لم تكن هذه الطريقة تستخدم أية بارمترات.

تمرير البارمترات للدالة أو الطريقة:

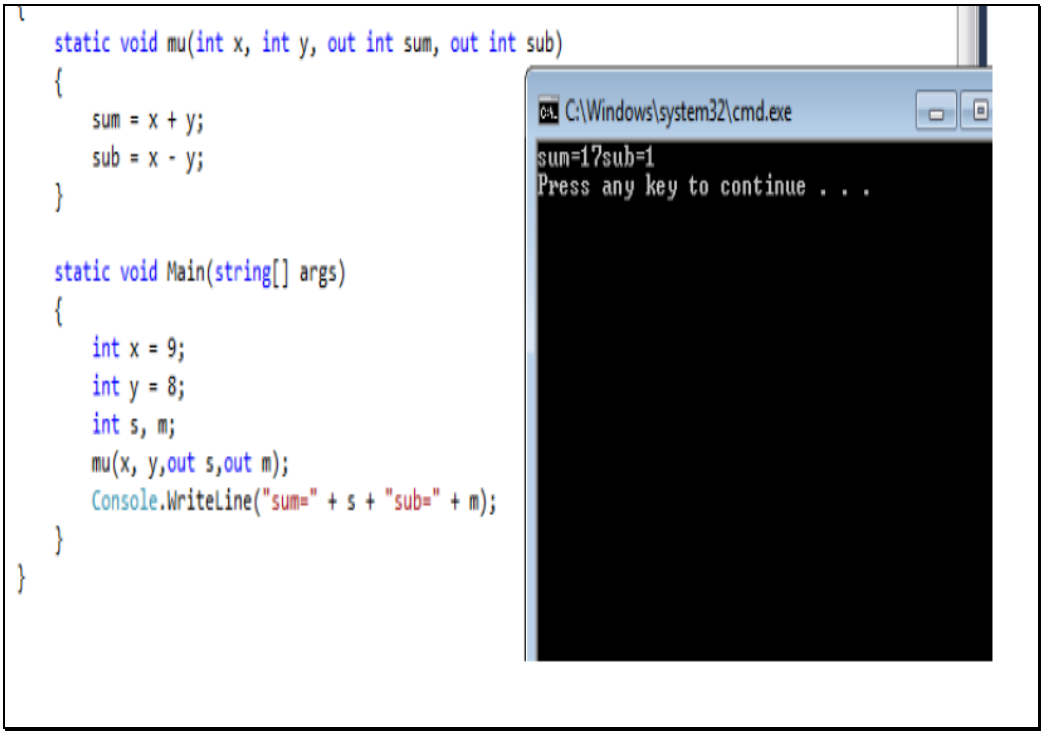
هناك عدة طرق لتمرير البارامترات إلى الطريقة، وهي:

١. التمرير بالقيمة (Passing By Value): في هذه الطريقة أي تعديلات تطرأ على قيمة البارامتر ضمن الطريقة لن تؤثر على القيمة الأساسية لهذا المتغير في البرنامج المستدعي، لأنه في هذه الحالة يتم عمل نسخة من البيانات وإرسالها إلى الطريقة، لتقوم الطريقة باستخدام هذه النسخة، وبالتالي أي تعديلات تطرأ ستؤثر على هذه النسخة ولن تتأثر القيمة الأصلية بها.

٢. التمرير بالمرجع (Passing By Refernce): في هذه الطريقة أي تعديلات تطرأ على قيمة البارامتر ضمن الطريقة ستُغير القيمة الأساسية لهذا المتغير في البرنامج المستدعي، للدلالة على أن هذا البارامتر سيُمرر بالمرجع لابد من استخدام الكلمة ref قبل البارامتر وذلك عند التصريح عن الطريقة وكذلك عند استدعائها.

٣. استخدام بارامترات الخرج (Out Parameters): بارامترات الخرج مشابهة إلى حد بعيد لبارامترات المرجع لكنها تختلف باستخدام الكلمة out قبل اسم البارامتر عند التصريح وعند الاستدعاء، وكذلك تختلف بأن بارامترات الخرج لا تحتاج إلى تهيئتها بقيم ابتدائية قبل استدعاء الطريقة.

في المثال التالي تم التصريح عن الطريقة mu التي تقوم بجمع عددين وتعيد ناتج الجمع في بارامتر الخرج sum وناتج الطرح في بارامتر الخرج sub



نلاحظ أنه في الطريقة الرئيسية للبرنامج (Main) تم تمرير البارمترات s, m للطريقة mu بدون أن يتم تهيئتهما بقيم ابتدائية.

مثال:

```
Static   void  set_value ( int  x  )
  {
     X = 100 ;
  }
Static void Main (string [ ] args)
  {
    int   my_value = 5 ;
```

```
    Set_value (  my_value ) ;
Console.writeLine ("The value is {0}", my_value);
  }
```

تم في هذا المثال تمرير البارامتر للطريقة بواسطة القيمة، وبالتالي فإن ناتج التنفيذ هو :

The value is 5 .

بينما لو قمنا بتمرير البارامتر بواسطة المرجع:

```
Static   void  set_value ( ref int  x )
 {
     X = 100 ;
 }
Static void Main (string [ ] args)
  {
    int   my_value = 5 ;
    Set_value ( ref my_value) ;
Console.writeLine ("The value is {0}", my_value);
  }
```

فإن ناتج التنفيذ هو :

The value is 100

التراكيب (السجلات) Structures:

السجل هو نوع بيانات يتم تعريفه من قبل المبرمج، وهو عبارة عن بنية معطيات تضم مجموعة متحولات من أنواع مختلفة، ويمكن أن يتضمن عدداً من الطرق Methods، متحولات وطرق السجل تسمى (أعضاء السجل).

التصريح عن التركيب (السجل): يجب أن يتم التصريح عن التركيب أو السجل خارج الدالة الرئيسية main، وله الشكل العام التالي:

```
struct  structure_name
 {
   access_modifier   dataType    var1 ;
   access_modifier   dataType    var2 ;
  ........
  ..........
 ........
 }
```

حيث:

structure_name: اسم التركيب أو السجل.

Access_modifier : محدد الوصول لهذا المتحول (العضو) من خارج السجل.

dataType var1 : نوع واسم العضو (المتحول) الاول من أعضاء السجل.

dataType var2 : نوع واسم المتحول الثاني من أعضاء السجل.

.....

مثال:

المطلوب بناء بنية معطيات تسمح بتخزين اسم المستخدم وعنوانه وعمره، ثم طباعة هذه المعلومات على الشاشة، عند تسجيل دخول مستخدم جديد.

```
namespace Person_struct
{
    0 references
    class Program
    {
        1 reference
        struct person
        {
            public string name;
            public string address;
            public int age;

        } 2
        0 references
        static void Main(string[] args)
        {
            person p;
            Console.Write("Please, Enter Your Name: ");
            p.name = Console.ReadLine();

            Console.Write("Please, Enter Your Address: ");
            p.address = Console.ReadLine();

            Console.Write("Please, Enter Your Age: ");
            p.age = int.Parse(Console.ReadLine());

            Console.WriteLine("************************************");
            Console.WriteLine("Hello {0}, ", p.name);
            Console.WriteLine("Your address is {0}, and you are {1} years old. ", p.address, p.age );

            Console.ReadKey();
        }
    } }
```