Function ใน C#

Suphot Sawattiwong
tohpus@gmail.com

การรับส่งค่าข้อมูล

```
int i = 1;
float a = 20.0f;
Console.WriteLine("Hello World");
i =Console.ReadLine();
Console.ReadLine();
```

ส่วนที่เป็นข้อมูล ในนี้ได้แก่

- i, กับ a เป็นข้อมูลที่เป็นตัวแปร
- ส่วน “Hello World” เป็นข้อมูลที่เป็น literal

Literal

- เป็นข้อมูล ดังนั้นมันจึงต้องใช้เนื้อที่ในหน่วยความจำเหมือนกัน เพียงแต่ข้อมูลแบบนี้จะมีค่าตายตัวในหน่วยความจำ ซึ่งหน่วยความจำในตำแหน่งที่เก็บ literal จะไม่มีถูกจองไว้เหมือนกับที่ตัวแปรจอง หลังจากใช้งานเสร็จแล้ว หากหน่วยความจำตำแหน่งนั้นต้องถูกใช้งานด้วยข้อมูลอื่นๆ ก็จะเขียนทับไปเลย เนื่องจากมันเป็นข้อมูลที่ใช้ครั้งเดียวทิ้ง

การรับส่งค่าข้อมูล

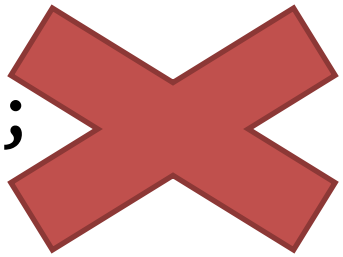
```
int x = 1;
int y = x;
Console.WriteLine(x);
```

x รับค่าจาก integer literal ที่มีค่า 1

x ส่งค่าให้แก่ตัวแปร y

x ส่งค่าให้ parameter ของ WriteLine(...);

```
7=x;
```



x ส่งค่าให้ integer literal ที่มีค่า 7 ไม่ได้

สรุปเรื่องการรับส่งค่าข้อมูล

- ข้อมูลทุกประเภทสามารถส่งค่าได้
- ข้อมูลที่เป็นตัวแปรเท่านั้นที่รับค่าได้
- ข้อมูลต้องมีประเภท(type) และ ค่า(value) เสมอ

Function

- มีลักษณะเหมือน “โปรแกรมย่อย” ที่ช่วยให้การเขียนโปรแกรมมีความเป็นระเบียบมากขึ้น แยกแยะออกเป็นสัดส่วน ลดจำนวน source code ที่ซ้ำกันออกไป ซึ่งเป็นพื้นฐานก่อนการเขียนโปรแกรมเชิง object
- Function เราสร้าง function เพื่อให้มันมีหน้าที่ทำงานใดงานหนึ่งเท่านั้นให้กับโปรแกรมหลัก
- “หน้าที่” ต้องมีผู้กระทำหน้าที่(function) กับ ผู้สั่งงาน(โปรแกรมที่เรียกใช้ function)
- ดังนั้นการสร้าง function แบ่งเป็น 2 ส่วน คือ การสร้าง function และ การเรียกใช้

ตัวอย่าง

```
using System;

namespace AreaRectCal
{
    class Program
    {
        static void Main(string[] args)
        {
            int width = 20;
            int height = 30;
            int area = width * height;
            Console.WriteLine(area);
            Console.ReadLine();
        }
    }
}
```

ตัวอย่างแบบที่การเรียกใช้ function

```
using System;

namespace AreaRectCal
{
    class Program
    {
        static void Main(string[] args)
        {
            int width = 20;
            int height = 30;
            int area = getRectArea(width, height);
            Console.WriteLine(area);
            Console.ReadLine();
        }
    }
}
```

ประโยชน์ของ Function

- ช่วยให้โปรแกรมดูง่ายขึ้น
- ช่วยให้เขียนโปรแกรมได้ง่ายและรวดเร็ว
- ช่วยลดงานที่ทำซ้ำบ่อยๆ

การสร้าง Function

- การสร้าง function ประกอบด้วย
  - การตั้งชื่อ function
  - การกำหนดประเภทตัวแปรในการส่งออกจาก function
  - การรับข้อมูลผ่าน Parameter

```
static <ชนิดตัวแปรส่งออก> <ชื่อfunction> (<ชนิดตัวแปร> <ชื่อ param1>, …)
{
    // ส่วนของโปรแกรม

}
```

ตัวอย่าง

```
static int getRectArea(int w, int h)
{
    // ส่วนของโปรแกรม
        return w * h;
}
```

การส่งค่าออก ใช้คำสั่ง return ตามด้วยตัวแปร

การเรียกใช้ Function

- การเรียกใช้ function ทำได้โดยการพิมพ์ชื่อ Function แล้วตามด้วย Parameter
- หากมีการส่งค่าออกมา ควรมีตัวแปรมาลองรับด้วยเช่น

```
int area = getRectArea(width, height);
double pi =getPi();
showRectArea(width, height);
showLine();
```

Function Main()

- Main เป็นรูปแบบ Function ชนิดหนึ่ง แต่ข้อแตกต่างระว่าง Main() กับ Function อื่นๆ คือ Main() เป็น function ที่ทำงานอัตโนมัติ ไม่เหมือน function ทั่วไปที่ต้องเรียกใช้งาน

รูปแบบของ Function

- Function มีหลายรูปแบบเช่น
  - Function ที่รับค่าและส่งค่าออก
  - Function ที่ส่งค่าออกอย่างเดียว
  - Function ที่รับค่าอย่างเดียว
  - Function ที่ไม่มีการรับค่า และส่งค่าออก

Procedure

Function ที่รับค่าและส่งค่าออก

```
static int getRectArea(int w, int h)
{
    // ส่วนของโปรแกรม
        return w * h;
}
```

Function ที่ส่งค่าออกอย่างเดียว

```
static double getPi()
{
    // ส่วนของโปรแกรม
        return 22/7.0;
}
```

Function ที่รับค่าอย่างเดียว

```
static void showRectArea(int w, int h)
{
    // ส่วนของโปรแกรม
        Console.WriteLine("Area is {0}", (w*h));
}
```

Function ที่ไม่รับค่าและส่งค่า

```
static void showLine()
{
    // ส่วนของโปรแกรม
        Console.WriteLine("----------------");
}
```

ตัวอย่าง

```
using System;

namespace FunctionExample1
{
   class Program
  {
      static int sum(int x, int y)
     {
         return x + y;
     }
      static void Main(string[] args)
     {
         int a, b;
         a = 6;
         b = 7;
         Console.WriteLine(sum(a, b));      // 13
         Console.WriteLine(sum(sum(a,9),b));   //22
         Console.ReadLine();
     }
  }
}
```

ตัวอย่าง 2

```
using System;

namespace FunctionExample2
{
   class Program
  {
     static int arrayMaxValue(int[] a)
    {
        int max = a[0];
        for (int i = 1; i < a.Length; i++)
       {
           if (a[i] > max) max = a[i];
        }
        return max;
    }
     static void Main(string[] args)
    {
        int[] x = new int[5] { 6, 2, 7, 8, 10 };
        Console.WriteLine("max ="+arrayMaxValue(x));
        Console.ReadLine();
    }
  }
}
```