

เมทอด

Method

วีระยุทธ คุณรัตนสิริ

เนื้อหา

ภาพรวมของเมททีอด

การเรียกเมททีอด

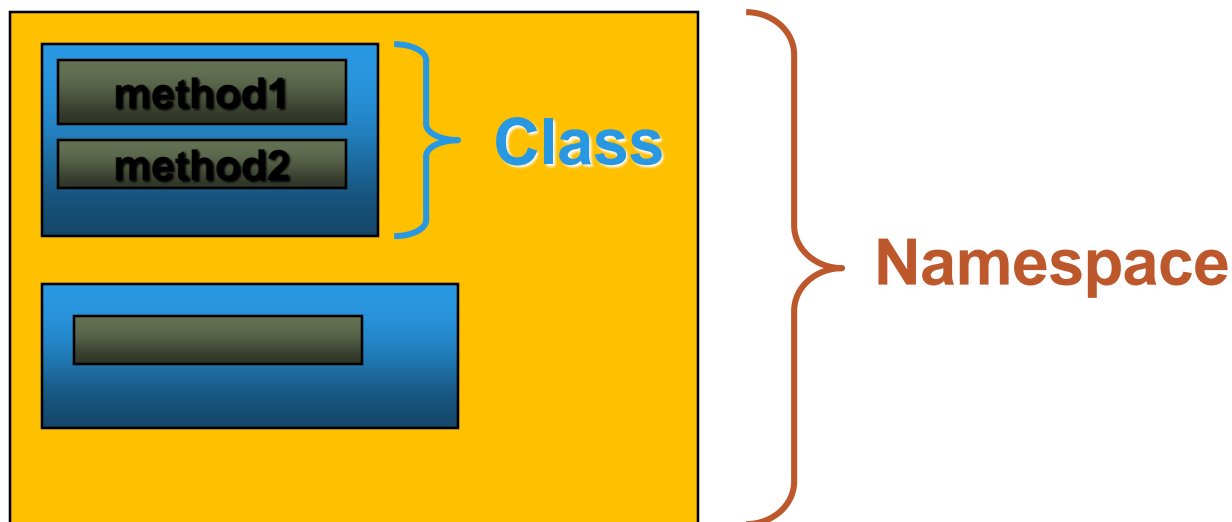
การประกาศเมททีอด

โครงสร้างโปรแกรมภาษา C#

โปรแกรมภาษา C# ประกอบด้วยหลายเนมสเปซ (namespace)

เนมสเปซหนึ่งประกอบด้วยหลายคลาส

คลาสหนึ่งประกอบด้วยหลายเมทอด



โครงสร้างโปรแกรม

```
namespace Hello {  
    class HelloWorldClass {  
        static void Main () {  
            System.Console.WriteLine("Hello World!");  
            System.Console.ReadLine();  
        }  
    }  
}
```

ที่ผ่านมามีโปรแกรมเราจะมีเพียงเมธอดเดียวเท่านั้นคือ Main

ประโยชน์ของการใช้เมท็อด

- ช่วยมองงานที่ซับซ้อนเป็นงานย่อย ๆ ที่เล็กลง
- ช่วยลดการเขียนโค้ดที่ซ้ำซ้อน
- ช่วยซ่อนรายละเอียดของโปรแกรมไว้ที่ส่วนอื่น
- ทำให้โปรแกรมดูมีระเบียบและง่ายต่อการเข้าใจ
- เพิ่มความสะดวกในการนำโค้ดที่เขียนไว้แล้วไปใช้ในโปรแกรมอื่น ๆ

ภาพรวมของเมธอด

โปรแกรมหนึ่งสามารถประกอบด้วยเมธอดมากกว่าหนึ่ง

```
namespace Hello {  
    class HelloWorldClass {  
        static void Main () {  
            :  
        }  
  
        static void Method1 () {  
            :  
        }  
  
        static int Method2 (int x, double y) {  
            :  
        }  
    }  
}
```

ประเภทของเมทอด

ประเภทของเมทอด

เมทอดที่มีอยู่แล้ว

ส่งคืนค่า

ไม่คืนค่า

เมทอดที่สร้างขึ้นเอง

ส่งคืนค่า

ไม่คืนค่า

ฟังก์ชันและโปรแกรมย่อย

ฟังก์ชัน (Function) — เมทีอดที่มีการส่งคืนค่า

โปรแกรมย่อย (Subroutine หรือ Procedure) — เมทีอดที่ไม่มีการส่งคืนค่า

การเรียกเมทอด

การเรียกเมทอดซึ่งไม่มีการส่งคืนค่า

```
Console.WriteLine("Hello, World!");  
Console.Write("Enter N:");
```

การเรียกเมทอดซึ่งมีการส่งคืนค่า

```
string s = Console.ReadLine();  
double d = double.Parse(Console.ReadLine());  
double p = Math.Pow(3, 5);
```

การประกาศเมทอด

ไวยากรณ์

```
static <return-type> <method-name>(<parameter list>)  
{  
    <statement-1>;  
    <statement-2>;  
    :  
}
```

<return-type> สามารถเป็น

- ประเภทข้อมูล C# : **int**, **double**, **string**, ...
- ประเภทข้อมูลซึ่งกำหนดโดยผู้ใช้: struct/class name, ...
- ไม่มีการส่งคืนค่า: **void**

} **ต้องมีคำสั่ง return**

<parameter-list>

- ประกอบด้วย **<parameter-type>** และ **<parameter-name>** เช่น **int x**, **string s**
- หรือ อาจเว้นว่างไว้ หากไม่มี

ตัวอย่างการประกาศเมธอด

เมธอด **Show10Stars** แสดงอักขระ '*' 10 ตัวบนหน้าจอ

```
static void Show10Stars () {  
}
```

เมธอด **Show10Stars** แสดงอักขระ '*' ตามจำนวนที่ได้กำหนด

```
static void ShowStars (int n) {  
}
```

เมธอด **ComputeArea** คำนวณขนาดพื้นที่ของรูปสามเหลี่ยม

```
static double ComputeArea (double w, double h) {  
}
```

คำสั่ง return

ต้องใช้สำหรับเมทอดที่ต้องการส่งค่าคือ

ไวยากรณ์:

```
return <expression>;
```

<expression> ต้องสอดคล้องกับประเภทข้อมูลที่ส่งคืนซึ่งได้ประกาศไว้ เช่น

```
static double Average(double x, double y) {  
    return (x+y)/2.0;  
}
```

ค่าที่ส่งคืน

...ต้องสอดคล้องกับ
return type ที่กำหนด

พารามิเตอร์ของเมทอด

เมทอดสามารถประกาศให้รับได้ทั้งหนึ่ง หรือหลายพารามิเตอร์

เมื่อเรียกเมทอด

```
class MethodEx {  
    static void Main() {  
        int x = Method1();  
        Method2(x, "Hello", 3.5);  
    }  
  
    static int Method1() {  
        :  
    }  
  
    static void Method2(int a, string x, double b) {  
        :  
    }  
}
```

พารามิเตอร์ของเมทอด

```
class MethodEx {
    static void Main() {
        int x = Method1();
        Method2(x, "Hello", 3.5);
    }

    static int Method1() {
        :
    }

    static void Method2(int a, string x, double b) {
        :
    }
}
```

แบบฝึกหัด

พิจารณาโค้ด

หากทำการเรียกเมธอดใน
Main() การเรียกเมธอดใดถูกต้อง?

- ✓ 1. `x = Hello();`
- ✗ 2. `Hello("everyone");`
- ✓ 3. `Hello();`
- ✗ 4. `Hi(s, y);`
- ✗ 5. `y = Hi(3, t);`
- ✓ 6. `Hi(x, "there");`
- ✓ 7. `Sawadee(x, x);`

```
class Quiz {
    static void Main() {
        int x, y; string s, t;
        :
        :
    }

    static int Hello()
    {
        :
    }

    static void Hi(int a, string b)
    {
        :
    }

    static void Sawadee(int a, int b)
    {
        :
    }
}
```

ตัวอย่างเมทอด #1

เขียนเมทอด `RectArea`

- รับพารามิเตอร์: `int w, int h`
- คำนวนและส่งคืนค่าพื้นที่ของสี่เหลี่ยมผืนผ้าที่มีความกว้างเท่ากับ `w` และความสูงเท่ากับ `h`

```
static int RectArea(int w, int h) {  
    int area;  
    area = w * h;  
    return area;  
}
```


ตัวอย่างเมท็อด #2

เขียนเมท็อด `PrintRect`

- รับพารามิเตอร์: `int w, int h`
- พิมพ์รูปสี่เหลี่ยมผืนผ้าขนาดความกว้าง `w` และความสูง `h` ด้วยอักขระ '*' บนหน้าจอ

```
static void PrintRect(int w, int h) {  
    int i, j;  
    for (i = 0; i < h; i++) {  
        for (j = 0; j < w; j++) {  
            Console.Write('*');  
        }  
        Console.WriteLine();  
    }  
}
```

ลึกลับมอง

เขียนเมทอด **DrawLine**

- รับพารามิเตอร์หนึ่งตัว: `int len`
- ไม่มีการส่งคืนค่าใด ๆ
- วาดเส้นตรงด้วย `len` '*'

```
using System;  
class Quiz {  
    static void Main() {  
        DrawLine(10);  
    }  
}
```

ใส่เมทอด **DrawLine** ตรงนี้...

โปรแกรมข้างต้นต้องแสดงผลลัพธ์เป็น

```
*****
```