คำสั่งทางคณิตศาสตร์และตรรกะ

วีระยุทธ คุณรัตนสิริ

เนื้อหา

คำสั่งรับอินพุทจากผู้ใช้

ตัวดำเนินการทางคณิตศาสตร์

การเปลี่ยนชนิดข้อมูล

การเรียกใช้งานเมท็อด

เมท็อดในคลาส **Math**

คำสั่งรับอินพุท

ใช้เมท็อด **ReadLine** ซึ่งอยู่ในคลาส **Console**

- รับข้อมูลจากแป้นพิมพ์จนกว่าผู้ใช้จะกด *Enter*
- ส่งค่าทั้งหมดให้โปรแกรมในรูปสตริง

ตัวอย่าง:

```
string yourname;
yourname = System.Console.ReadLine();
```

ตัวอย่างโปรแกรม: ReadLine

โปรแกรมด้านล่างจะให้ผู้ใช้ป้อนชื่อและกล่าวคำทักทาย

```
using System;
class HelloYou
{
    static void Main()
    {
        string name;
        Console.Write("What is your name: ");
        name = Console.ReadLine();
        Console.WriteLine("Hello, {0}", name);
        Console.ReadLine();
    }
}
```

การเปลี่ยนสตริงให้เป็นตัวเลข

เป็นที่ทราบว่าเมท็อด **Console.ReadLine** คืนค่าเป็นสตริง

ตัวเลขที่อยู่ในรูปสตริงไม่สามารถนำไปใช้ในการคำนวณได้โดยตรง

```
string s = "12";
int x = s+1;  // Error!
```

ชนิดข้อมูลตัวเลขทุกชนิดมีเมท็อด **Parse** ซึ่งทำหน้าที่เปลี่ยนข้อมูลสตริงให้เป็นชนิดตัวเลขตามแบบข้อมูลนั้น ๆ

```
string s = Console.ReadLine();
int x = int.Parse(s);
double y = double.Parse(s);
```

ตัวอย่าง: *คำนวนพื้นที่สี่เหลี่ยม*

```
using System;
class AreaCalculation
{
    static void Main()
    {
        int width, height, area;
        string input;
        Console.Write("Enter width: ");
        input = Console.ReadLine();
        width = int.Parse(input);
        Console.Write("Enter height: ");
        input = Console.ReadLine();
        height = int.Parse(input);
        area = width * height;
        Console.WriteLine("Area = {0}x{1} = {2}", width, height, area);
    }
}
```

Obtaining "width"

Obtaining "height"

แบบฝึกหัด: *แก้สมการ*

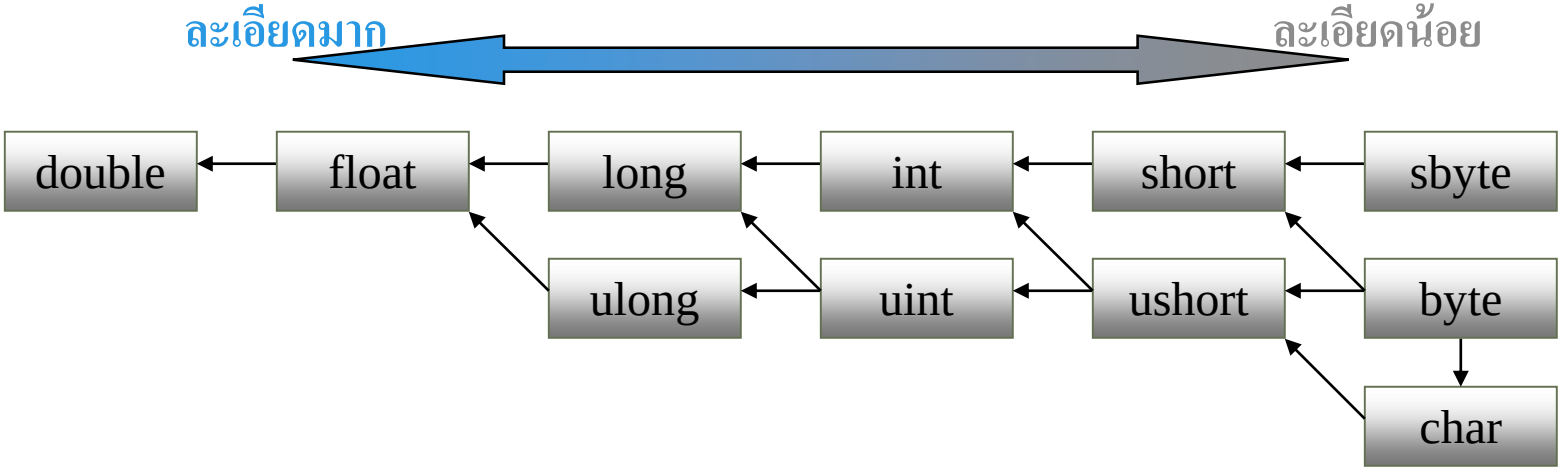
เขียนโปรแกรมเพื่ออ่านค่า $x+y$ และ $x-y$, จากนั้นให้ตอบว่าค่าของ x และ y คือเท่าใด

- เริ่มต้นจากการคิดขั้นตอนวิธี และเขียน *pseudo-code*
- นำ pseudo-code มาเขียนเป็นโปรแกรม

การเปลี่ยนชนิดข้อมูลตัวเลข

การกำหนดค่าให้ตัวแปรที่เก็บข้อมูลได้ละเอียดมากกว่าทำได้ทันที

- เนื่องจากข้อมูลไม่มีการสูญหาย



การเปลี่ยนชนิดข้อมูลตัวเลข

C# บังคับให้ระบุชนิดข้อมูลให้ชัดเจนสำหรับการกำหนดค่าให้ตัวแปรที่ละเอียดน้อยกว่า

```
int i = 50;
double d = i;  // OK
```

```
double d = 5000.78;
long x = d;          // ERROR!!!
long x = (long) d; // OK
```

ระบุชัดเจนว่าต้องการเปลี่ยนค่า double เป็น long

ตัวดำเนินการแบบ "modify-and-assign"

เราสามารถผสมตัวดำเนินการทางคณิตศาสตร์บางตัวเข้ากับ "=" เพื่อใช้เป็นตัวดำเนินการแบบ modify-and-assign

รูปแบบคำสั่ง	ความหมาย
var += *expression*	เพิ่มค่าใน *var* เป็นจำนวนเท่ากับ *expression*
var -= *expression*	ลดค่าใน *var* เป็นจำนวนเท่ากับ *expression*
var *= *expression*	คูณค่า *var* ด้วย *expression* เก็บผลลัพธ์ใน *var*
var /= *expression*	หารค่า *var* ด้วย *expression* เก็บผลลัพธ์ใน *var*

```
sum += x;      // ให้ผลเหมือนกับ sum = sum + x
prod *= 2.5;   // ให้ผลเหมือนกับ prod = prod * 2.5
y -= 3+a;      // ให้ผลเหมือนกับ y = y - (3+a)
```

ตัวดำเนินการสำหรับเพิ่ม/ลดค่า

ตัวดำเนินการ ++ และ — ใช้สำหรับเพิ่มและลดค่าให้ตัวแปรทีละ 1

คำสั่ง	ผลลัพธ์
x++	เพิ่มค่า x ขึ้น 1
x--	ลดค่า x ลง 1

ตัวอย่าง:

```
int n = 0;

n++;     // ให้ผลเหมือนกับ n = n+1, or n += 1
n++;     // n มีค่าเท่ากับ 2
n--;     // n มีค่าเท่ากับ 1
```

การเรียกใช้งานเมท็อด

คลาสหลายคลาสมีเมท็อดชนิดสแตติก (static) ที่สามารถเรียกใช้ผ่านคลาสโดยตรงได้ทันที

- เรียกเมท็อด *methodName* ผ่านคลาส *className* ที่อยู่ในเนมสเปส *namespaceName*

namespaceName.className.methodName(optional-arguments)

- หากเนมสเปส *namespaceName* ถูกระบุผ่านคำสั่ง **`using`** ไว้ก่อนหน้านี้ การเรียกเมท็อดไม่จำเป็นต้องระบุเนมสเปส

การเรียกใช้งานเมท็อด

เมท็อดที่คุ้นเคยกันได้แก่ System.Console.WriteLine() และ System.Console.ReadLine()

```
using namespaceName;
  :
className.methodName(optional-arguments)
```

คลาส *Math*

คลาส Math ในเนมสเปส System ได้เตรียมเมท็อดและค่าคงที่ไว้หลายตัวสำหรับการคำนวณทางคณิตศาสตร์ที่ซับซ้อนขึ้น

ตัวอย่าง (บางส่วน):

เมท็อด/ค่าคงที่	ค่าที่คืนกลับมา	ตัวอย่างการใช้งาน	ผลลัพธ์
PI	ค่า π	`Math.PI`	**3.1415927**
Max(x,y)	ค่าที่มากกว่าระหว่าง x และ y	`Math.Max(1,2)`	**2**
Abs(x)	ค่าสัมบูรณ์ของ x	`Math.Abs(-1.3)`	**1.3**
Sqrt(x)	รากที่สองของ x	`Math.Sqrt(4.0)`	**2.0**
Round(x)	ค่า x ที่ปัดเป็นจำนวนเต็ม	`Math.Round(0.8)`	**1**
Pow(x,y)	ค่า x ยกกำลัง y	`Math.Pow(3,2)`	**9.0**
Log(x)	ล็อกฐานธรรมชาติของ x	`Math.Log(10)`	**2.302585**
Ceiling(x)	ค่า x ที่ปัดเศษขึ้น	`Math.Ceiling(4.1)`	**5**
Cos(x)	โคซายน์ของ x เรเดียน	`Math.Cos(Math.PI)`	**-1**