

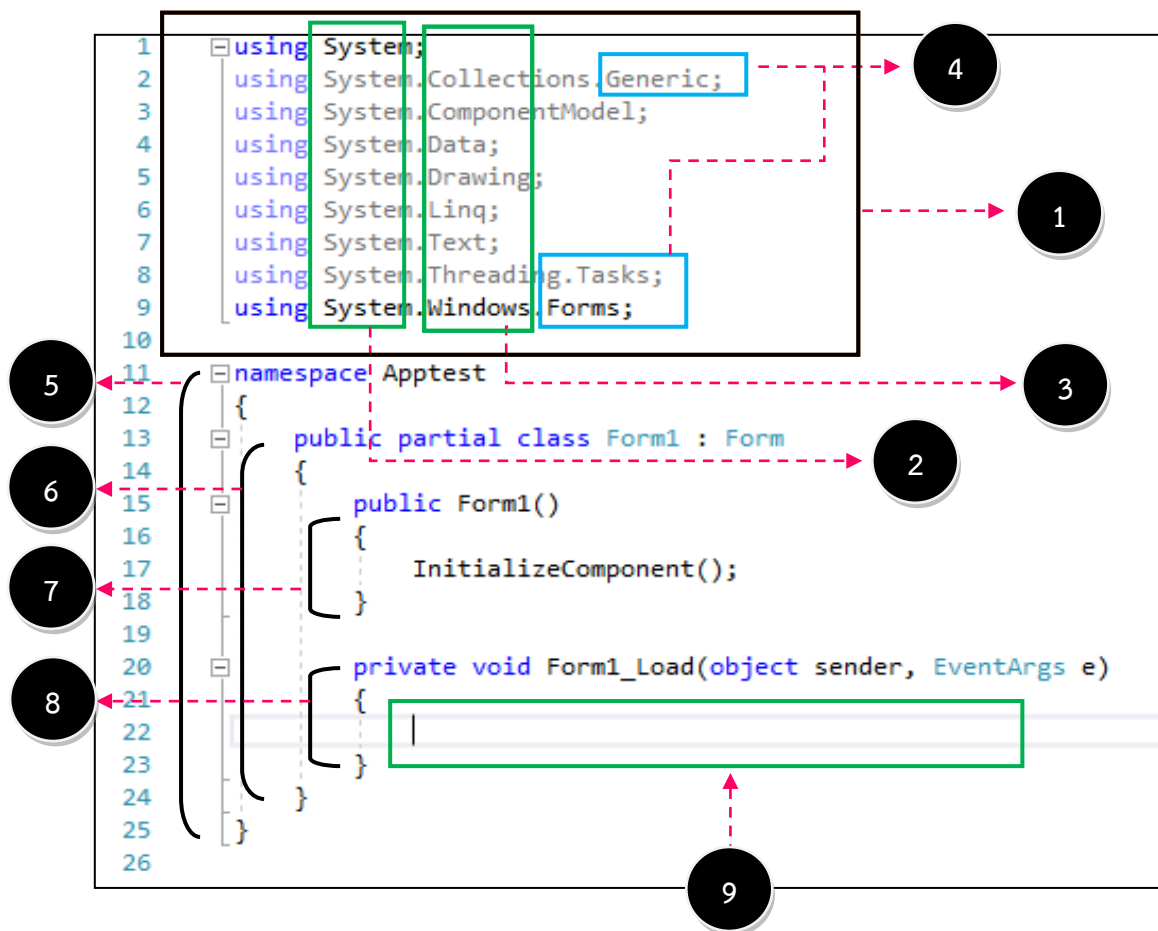


2.1 โครงสร้างของภาษา C#

ในบทนี้จะมาเรียนรู้เกี่ยวกับหลักการเขียนโปรแกรมภาษา C# ซึ่งจะเป็นพื้นฐานสำคัญที่จะทำให้เกิดเข้าใจในการเขียนโปรแกรมภาษา C# เบื้องต้นได้เป็นอย่างดี อีกทั้งยังเป็นพื้นฐานสำคัญสำหรับการต่อยอดในการเขียนโปรแกรมแบบ OOP และการใช้งาน .NET Framework อีกด้วย

2.1.1 โครงสร้างภาษา C#

เมื่อใช้งานในส่วนของ WindowFormApplication จะมีส่วนที่ช่วยเพิ่มความง่ายในการเขียนโปรแกรมคำสั่งภาษา C# เนื่องจากส่วนนี้จะสร้างคำสั่งที่เป็นโครงสร้างพื้นฐานขึ้นมาให้โดยอัตโนมัติ ซึ่งผู้ใช้งานจะเรียกส่วนนี้ขึ้นมาใช้งานได้เอง จากนั้นก็สามารถแก้ไข หรือเพิ่มเติมคำสั่งได้ตามความต้องการ โดยมีรายละเอียดของคำสั่งเหล่านี้ ดังนี้



รูปที่ 2.1.1 รูปแสดงโครงสร้างภาษา C#

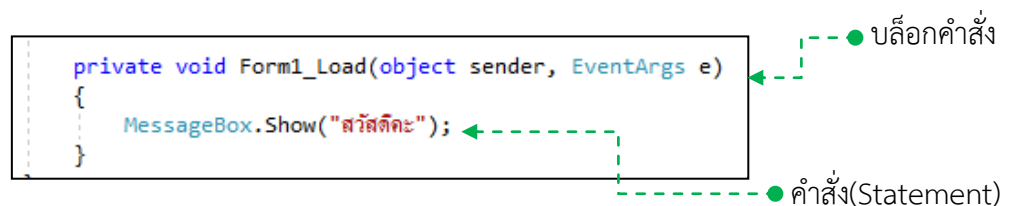
- หมายเลขที่ 1 เป็นส่วน Header ใช้สำหรับประกาศคำสั่งต่างๆ
- หมายเลขที่ 2 เป็นส่วนประกาศ Namespace ที่ชื่อว่า System
- หมายเลขที่ 3 เป็นส่วนประกาศ Namespace ที่ชื่อว่า Collections, ComponentModel, Data, Drawing, Linq, Text, Threading, Windows
- หมายเลขที่ 4 เป็นส่วนประกาศ Namespace ที่ชื่อว่า Generic, Tasks, Forms
- หมายเลขที่ 5 เป็นส่วนประกาศ Namespace ที่ชื่อว่า Apptest ซึ่งจะตรงกันกับชื่องานที่ตั้งขึ้นตอนที่สร้างโปรเจคใหม่
- หมายเลขที่ 6 เป็นส่วนประกาศ Class ที่ชื่อว่า Form1 ซึ่งหมายถึง Form1 ที่เราสร้างขึ้นมานั่นเอง
- หมายเลขที่ 7 เป็นส่วนประกาศ Constructor ของ Form1
- หมายเลขที่ 8 เป็นส่วนประกาศ Method ของ Form1 ที่จะสั่งให้วัตถุต่างๆ ใน Form1 ทำงานเมื่อเปิด Form1
- หมายเลขที่ 9 เป็นส่วนที่ใช้สำหรับพิมพ์ Statement หรือคำสั่งให้โปรแกรมทำงานตามที่ต้องการ

การกำหนดขอบเขตต่างๆ ของคำสั่ง จะใช้เครื่องหมายปีกกาเปิด { เพื่อบอกจุดเริ่มต้นของคำสั่ง และใช้เครื่องหมายปีกกาปิด } เพื่อบอกจุดสิ้นสุดของคำสั่งเสมอ จะสังเกตว่าส่วนของ namespace จะมีเครื่องหมายปีกกาเปิดและปิดคู่กันซึ่งจะเป็นส่วนที่คลุมคำสั่งทั้งหมดไว้อีกทั้งตำแหน่งของเครื่องหมายปีกกาก็จะตรงกันด้วย

หากผู้ใช้งานเผลอลบปีกกาอันใดอันหนึ่งไป ให้เติมเข้าไปให้ครบคู่ไม่อย่างนั้นแล้วโปรแกรมจะไม่สามารถทำงานได้ตามที่ต้องการ


2.1.2 รูปแบบการเขียนโปรแกรมภาษา C#

ในการเขียนโปรแกรมด้วยภาษา C# นั้นได้รับอิทธิพลมาจากภาษา C++ จึงมีรูปแบบคำสั่งที่คล้ายคลึงกัน และมีความคล้ายกับภาษา JAVA มากอีกภาษาหนึ่ง โดยที่คำสั่งนั้นจะถูกกำหนดให้อยู่ในบล็อกที่ครอบด้วยเครื่องหมายปีกกาเปิด { และเครื่องหมายปิด } นั่นเอง




รูปที่ 2.1.2 รูปแสดงบล็อกคำสั่งของภาษา C#

ภายในบล็อกนั้นจะประกอบไปด้วยคำสั่งซึ่งแต่ละคำสั่งแยกจากกัน(หรือสั้นสุดลง) ด้วยเครื่องหมาย ; ซึ่งคำสั่งหนึ่งไม่จำเป็นต้องเขียนจบภายในบรรทัดเดียวกันก็ได้ เพราะโปรแกรมจะใช้เครื่องหมาย ; เพื่อเป็นตัวบอกสั้นสุดคำสั่งนั่นเอง

ยกตัวอย่างการเขียนคำสั่งให้กับปุ่มปิดโปรแกรม ในการเขียนคำสั่งจะผูกติดกับวัตถุอื่นๆ แยกเป็นอิสระออกจากวัตถุอื่น ซึ่งง่ายต่อการแก้ไข เมื่อผู้ใช้งานนำ control Button (ปุ่ม) มาวางบน Form ปรับแต่งคุณสมบัติต่างๆ ของปุ่มเรียบร้อยแล้ว ผู้ใช้ก็เรียกส่วนการเขียนคำสั่งขึ้นมา แล้วเขียนคำสั่งปิดโปรแกรมลงไปในปุ่ม 

```
private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}
```



รูปที่ 2.1.3 รูปแสดงบล็อกคำสั่งของปุ่มปิด

2.1.3 รู้จัก Event

ในการทำงานของออบเจกต์ที่อยู่บนแอปพลิเคชันที่ทำงานบน Window นั้นผู้ใช้งานย่อมพบกับเหตุการณ์ต่างๆ ที่เกิดจากทั้งการโต้ตอบกับผู้ใช้ งาน, การทำงานร่วมกับโปรแกรมอื่น, หรือทำงานร่วมกับ Windows เอง เราเรียกเหตุการณ์ที่เกิดขึ้นกับออบเจกต์ต่างๆ นี้ว่า อีเวนต์ ในการจัดการกับอีเวนต์นั้นไม่จำเป็นต้องทำทุกตัว แต่จะเลือกจัดการเฉพาะตัวที่สนใจเท่านั้น ซึ่งโปรแกรมที่เขียนขึ้นมาเพื่อจัดการกับอีเวนต์แต่ละตัวนั้นเรียกว่า Event Handler

ในโปรแกรม Visual C# 2017 จะมีลักษณะการทำงานที่เรียกว่า Event Drivent คือมีการทำงานที่ตอบสนองต่ออีเวนต์ต่างๆ และมีลักษณะเป็น Visual คือมองเห็นได้ นั้นหมายถึงว่า ปุ่ม(Button) ที่วางบนฟอร์ม (Form) จะมีอีเวนต์คือ คลิกที่ปุ่ม แล้วคำสั่งต่างๆ ใน Method ที่ชื่อว่า button1_click จะทำงาน เป็นต้น ดังตัวอย่าง

```
private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}
```

รูปที่ 2.1.4 รูปแสดงคำสั่งใน Method button1_click

จากบล็อกคำสั่งในรูปที่ 2.1.4 อธิบายได้ว่า

ออบเจกต์ที่ใช้ (Control)	คือ button1
อีเวนต์หรือเหตุการณ์ของออบเจกต์นี้ (Event)	คือ Click
เมธอด(Method)	ชื่อว่า button1_click
คำสั่ง (Statement)	คือ this.close(); (สั่งให้ปิดโปรแกรม)

ตารางที่ 2.1.1 ตารางแสดงตัวอย่างอีเวนต์ต่างๆ ของ control

control	อีเวนต์	หน้าที่ของอีเวนต์
Form	Active	เมื่อฟอร์มทำงาน
	Click	เมื่อคลิกเมาส์ลงบนฟอร์ม
	Load	เมื่อฟอร์มถูกเรียกขึ้นมาใช้งาน
	Unload	เมื่อยกเลิกการใช้งานฟอร์ม
Textbox	Keypress	พิมพ์ข้อมูลเข้าไปใน Textbox
	TextChanged	เมื่อ Textbox มีการเปลี่ยนตัวอักษร
Button	Click	เมื่อคลิกเมาส์ลงบนปุ่ม
ComboBox	DropDown	เมื่อคลิกลูกศรที่ ComboBox
	Enter	เมื่อ ComboBox ถูกเข้าถึงครั้งแรก
ListBox	KeyDown	เมื่อ ListBox ถูกคลิกเป็นพิมพ์ใดๆ
	MouseDoubleClick	เมื่อดับเบิลคลิกเมาส์ลงบนลิสต์บ็อก