INFOPRESS

คู่มือ CODING ด้วย

VISUAL BASIC 2019

ฉบับผู้เริ่มต้น

- ครอบคลุมการพัฒนาแอพพลิเคชันด้วย .NET 5
- อ่านเข้าใจง่าย อธิบายอย่างเป็นขั้นตอน มีตัวอย่างประกอบทุกหัวข้อ
- มีตัวอย่างระบบงานพื้นฐานเพื่อใช้ในการเรียนรู้และพัฒนาสู่ระบบงานจริง
- เหมาะสำหรับนักเรียน นักศึกษา และผู้ที่เริ่มต้นพัฒนาแอพพลิเคชัน

ดาวน์โหลดฟรี ไฟล์ตัวอย่างภายในเล่ม https://serazu.com/

9786164871519

ศุภชัย สมพานิช

สารบัญ

บทที่ 1 เตรียมความพร้อมก่อนเริ่มต้นเขียนโปรแกรมด้วยภาษา Visual Basic 2019 1

ขอบเขตการนำเสนอของหนังสือเล่มนี้ 1

การดาวน์โหลดและติดตั้งโปรแกรม Visual Studio 2019 2

ทำความรู้จักกับประเภทโปรเจ็กต์ที่น่าสนใจในขั้นต้น 3

การอัพเดตโปรแกรม Visual Studio 2019 4

การติดตั้งฟอนต์ Cascadia Code สำหรับเขียนโค้ดโดยเฉพาะ 7

การสร้างโปรเจ็กต์ Windows Forms App (.NET Framework) 8

การออกแบบส่วนแสดงผลด้วยคอนโทรล (User Interface) 12

การปรับแต่งคอนโทรลขั้นต้นด้วยวิธีแก้ไขคุณสมบัติ (Property) 14

การกำหนดเลขบรรทัด 16

การทดสอบโปรเจ็กต์ 17

บทที่ 2 พื้นฐานการทำงานกับโปรเจ็กต์ประเภท Windows Forms App (.NET Framework) 19

ทำความรู้จักกับฟอร์ม (Form) 19

คุณสมบัติของ Form ที่สำคัญ 20

การเพิ่มฟอร์มใหม่เข้ามาในโปรเจ็กต์ 21

วิธีการสั่งให้เปิดฟอร์มใหม่โดยการเขียนโค้ด 22

หลักการเขียนโปรแกรมแบบ Event Driven Programming 24

พื้นฐานการใช้ฟอร์มแบบ MDI และสร้างระบบเมนู (Menu) 26

การสร้างเมนูให้กับฟอร์มหลัก 28

การลดขนาดโปรเจ็กต์ของ VB 2019 31

บทที่ 3 พื้นฐานการเขียนโปรแกรมด้วยภาษา Visual Basic 35

การประกาศตัวแปรขึ้นมาใช้เก็บข้อมูล (Variable) 35

การใช้งานข้อมูลชนิดตัวเลข 36

ความหมายของคำสั่ง Option Explicit On และ Option Strict On 37

□□□□□□.indd 1 10/9/2563 BE 15:18

กฎการตั้งชื่อตัวแปรในภาษา VB 38
คำสงวน (Reserved Words) ของภาษา VB 38
ขอบเขตการจัดเก็บข้อมูลแต่ละชนิดข้อมูลและขนาดหน่วยความจำที่ใช้ 39
การคั่นตัวเลขหลักพันด้วยเครื่องหมาย _ 43
การกำหนดชนิดข้อมูลโดยอัตโนมัติ (Type Inference) 43
การใช้งานเลขจำนวนเต็มชนิด BigInteger 46
ตัวดำเนินการด้านคณิตศาสตร์ 48
ตัวดำเนินการด้านอัพเดตค่า 48
การเขียนหมายเหตุ (Comment) ในโค้ด 49
ขอบเขตของตัวแปร (Variable Scope) 49
การใช้งานตัวแปรระดับ Local และระดับฟอร์มในเวลาเดียวกัน 51
การใช้งานตัวแปรระดับ Block 54
การสร้างตัวแปรเก็บหลายค่าในเวลาเดียวกันด้วย ValueTuple 55
ค่าคงที่ (Constant) 58

บทที่ 4 ทำงานกับข้อมูลชนิดข้อความ String และวันเวลา DateTime 61
พื้นฐานการทำงานกับข้อมูลชนิดข้อความ String 61
วิธีการตรวจสอบจำนวนตัวอักษรใน String 62
การต่อข้อความเข้าด้วยกัน 63
การแปลงตัวเลขที่อยู่ในฐานะข้อความ String ให้กลายเป็นข้อมูลชนิดตัวเลข 65
การแทนที่ข้อความด้วยเมธอด Replace() 66
การแบ่งข้อความด้วยฟังก์ชัน Split() 66
การใช้งานข้อมูลชนิดตัวอักษร Char 67
การแปลงชนิดข้อความ String เป็นชนิดข้อมูลอาร์เรย์ของ Char 67
การจัดรูปแบบของตัวเลขทศนิยม 68
การแทรกค่าของตัวแปรเข้าไปในข้อความ String ด้วยเครื่องหมาย $ 71
ทำงานกับข้อมูลชนิดวันที่และเวลา (DateTime) 72
การบวกเพิ่มวันที่กับข้อมูลประเภท DateTime 72

บทที่ 5 การตรวจสอบและควบคุม 77
พื้นฐานการตรวจสอบเงื่อนไขด้วยคำสั่ง If...Else 77
ตัวดำเนินการด้านเปรียบเทียบ 79

□□□□□□.indd 2 10/9/2563 BE 15:18

การใช้ตัวดำเนินการ “และ” (AndAlso) และตัวดำเนินการ “หรือ” (OrElse) 80
การตรวจสอบเงื่อนไขร่วมกับตัวดำเนินการด้านตรรกะ 80
การตรวจสอบเงื่อนไขด้วยคำสั่ง IIf() 83
การตรวจสอบเงื่อนไขด้วยคำสั่ง Select Case 84
การวนลูปด้วยคำสั่ง For...Next 86
การวนลูปด้วยคำสั่ง While...End While 88
การวนลูปแบบ Do...Loop While 89

บทที่ 6 การใช้งานคอนโทรลเบื้องต้น 93
ทำความรู้จักกับแถบคอนโทรล 93
พื้นฐานการสร้างโปรแกรมรับข้อความจากผู้ใช้งานด้วยคอนโทรล TextBox 95
คุณสมบัติของคอนโทรล TextBox ที่สำคัญ 98
การตรวจสอบข้อมูลว่างในคอนโทรล TextBox 99
การแสดงรูปภาพด้วยคอนโทรล PictureBox 101
การสร้างส่วนแสดงผลแบบ list ด้วย ComboBox และ ListBox 103
การเพิ่ม, ถอด และล้างรายการใน ListBox 108
การสร้างส่วนแสดงผลแบบรายการด้วยคอนโทรล ListView 112

บทที่ 7 โครงสร้างข้อมูลแบบอาร์เรย์ (Array) 117
พื้นฐานการใช้งานตัวแปรชุดแบบอาร์เรย์ (Array) 117
การอ่านค่าตัวแปรอาร์เรย์ด้วยลูปแบบ For Each...Next 119
การสร้างตัวแปรอาร์เรย์แบบ Type Inference 121
การกลับด้านข้อมูลในอาร์เรย์ 122
การสร้างอาร์เรย์ด้วยคลาส ArrayList 124
การถอดค่าซ้ำกันในอาร์เรย์ 126
การหาค่ามากสุดและน้อยสุดในอาร์เรย์ 128
การสร้างอาร์เรย์แบบหลายมิติ 132

บทที่ 8 การใช้งานฟังก์ชันและพารามิเตอร์ 135
รูปแบบฟังก์ชันในภาษา VB 135
การสร้างฟังก์ชันที่มีชื่อเหมือนกันด้วยการทำ Overload Method 139
ลักษณะของพารามิเตอร์ในฟังก์ชัน 143
พารามิเตอร์ชนิดรับเข้า ByVal (Input Parameter) 143

□□□□□□.indd 3 10/9/2563 BE 15:18

พารามิเตอร์ชนิดรับเข้าและส่งออกในเวลาเดียวกัน (ByRef) 144
การกำหนดลักษณะเพิ่มเติมของพารามิเตอร์ 145
พารามิเตอร์แบบทางเลือก (Optional Parameter) 146
การส่งค่าพารามิเตอร์แบบระบุชื่อ 148
การส่งพารามิเตอร์แบบ Params 149

บทที่ 9 พื้นฐานการเขียนโปรแกรมเชิงวัตถุ (OOP Programming - OOP) 153
การเขียนโปรแกรมเชิงวัตถุ (OOP) คืออะไร 153
พื้นฐานการสร้างคลาสขึ้นมาใช้งานเอง 154
ทำความรู้จักกับชนิดข้อมูลแบบ Anonymous Type 160
การจัดประเภทคลาสด้วยระบบเนมสเปซ (Namespaces) 164
วิธีการเรียกใช้เนมสเปซ 165
แบบที่ 1 อาศัยคำสั่ง Imports 165
แบบที่ 2 อาศัยการระบุชื่อเต็ม 166
แบบที่ 3 อาศัยการตั้งชื่อ Alias 167
แบบที่ 4 อาศัยคลาส Global 167
ทำความรู้จักกับ Constructor, การระบุสมาชิกด้วยคำสั่ง Me
และการกำหนดค่าเริ่มต้นด้วย Object Initializer 168

บทที่ 10 Inheritance & Polymorphism 175
กฎพื้นฐาน 3 ข้อของ OOP 175
พื้นฐานการสืบทอดคลาส (Inheritance) 176
การสร้างคลาสแบบ Abstract Class (MustInherit) 180
การใช้งาน Abstract Class ในฐานะ Dynamic Type 184
สถานะที่หลากหลาย (Polymorphism) 186

บทที่ 11 การใช้งานคอนโทรลต่างๆ ที่น่าสนใจ 191
แสดงข้อความด้วยคอนโทรล Label 191
คุณสมบัติของคอนโทรล Label ที่สำคัญ 192
การสร้างปุ่มกดด้วยคอนโทรล Button 192
คุณสมบัติของคอนโทรล Button ที่สำคัญ 193
การสร้างไฮเปอร์ลิงค์ด้วย LinkLabel 193
คุณสมบัติของคอนโทรล LinkLabel ที่สำคัญ 196

□□□□□□.indd 4 10/9/2563 BE 15:18

การแสดงความคืบหน้าในการทำงานด้วยคอนโทรล ProgressBar 197
การแสดงคำอธิบายด้วย ToolTip 198
การแสดงข้อมูลแบบลำดับชั้นด้วยคอนโทรล TreeView 199
คุณสมบัติของคอนโทรล TreeView ที่สำคัญ 203
การสร้างตัวจับเวลาด้วย Timer 204
การสร้างตัวเลือกแบบเลือกได้ 1 อย่างด้วยคอนโทรล RadioButton 205
การสร้างตัวเลือกแบบหลายตัวเลือกด้วยคอนโทรล CheckBox 207
การจัดกลุ่มด้วยคอนโทรล GroupBox 209
การใช้งานตัวบรรจุ Panel 209
การสร้างส่วนแสดงผลแบบแท็บด้วยคอนโทรล TabControl 210
การสร้างตัวเลือกวันที่ด้วย DateTimePicker 211
คุณสมบัติของคอนโทรล DateTimePicker ที่สำคัญ 213
การสร้างปฏิทินด้วยคอนโทรล MonthCalendar 214
คุณสมบัติของคอนโทรล MonthCalendar ที่สำคัญ 216
การสร้างรายการแบบมีตัวเลือก CheckBox ด้วยคอนโทรล CheckedListBox 217
การกำหนดรูปแบบข้อมูลที่รับเข้ามาด้วย MaskedTextBox 219
การสร้างตัวเลือกแบบตัวเลขด้วย NumericUpDowns 221
ทำงานกับไฟล์ข้อความด้วยคอนโทรล RichTextBox, OpenFileDialog
และ SaveFileDialog 223
คุณสมบัติของคอนโทรล RichTextBox ที่สำคัญ 223
คุณสมบัติของคอนโทรล OpenFileDialog ที่สำคัญ 223
คุณสมบัติของคอนโทรล SaveFileDialog ที่สำคัญ 224

บทที่ 12 การเขียนโปรแกรมแบบ Generic 229

ทำความรู้จักกับการเขียนโปรแกรมแบบ Generic ภายใน 30 นาที 229
ทำไม .NET จึงต้องมี Generic 234
พารามิเตอร์ T มีชนิดข้อมูลอะไรในขณะออกแบบ (Design-Time) 235
การสร้างคลาสแบบ Generic 236
การสร้างรายการ list แบบ Generic (Generic List) 238
การทำงานกับ Generic List ที่น่าสนใจ 240
ทำงานกับโครงสร้างข้อมูลแบบเข้าก่อน-ออกก่อน FIFO (First-In First-Out) 243
ทำงานกับโครงสร้างข้อมูลแบบเข้าหลัง-ออกก่อน LIFO (Last-In First-Out) 247

□□□□□□.indd 5 10/9/2563 BE 15:18

บทที่ 13 พื้นฐานการดักจับและตรวจสอบข้อผิดพลาด (Debug & Exception) 251

พื้นฐานการดักจับข้อผิดพลาดด้วย Try...Catch 251

การดักจับข้อผิดพลาดกับโครงสร้างข้อมูล Array 254

บทที่ 14 พื้นฐานการใช้งานระบบฐานข้อมูล SQL Server 2019 Express Edition 257

การดาวน์โหลดและติดตั้งฐานข้อมูล SQL Server 2019 Express Edition 257

การทำงานกับระบบฐานข้อมูล SQL Server 262

การดาวน์โหลดและติดตั้ง SQL Server Management Studio (SSMS) 262

พื้นฐานการใช้งานฐานข้อมูล SQL Server 2019 Express Edition ด้วย SSMS 264

การยกเลิกป้องกันการแก้ไขโครงสร้างตาราง 269

การสร้างความสัมพันธ์ระหว่างตารางในฐานข้อมูล SQL Server 2019 270

การ Backup และ Restore ฐานข้อมูล SQL Server 2019 273

การเพิ่มฐานข้อมูล Northwind เข้าไปใน SQL Server 2019 Express Edition 277

การจัดการฐานข้อมูล SQL Server ด้วย Azure Data Studio (ADS) 279

พื้นฐานการเชื่อมต่อกับฐานข้อมูล SQL Server ด้วย ADS 281

การทำงานกับฐานข้อมูล SQL Server ในขั้นต้นด้วย ADS 283

บทที่ 15 พื้นฐานการทำงานกับฐานข้อมูล SQL Server ด้วย Entity Framework 285

พื้นฐานการติดต่อฐานข้อมูล SQL Server ด้วย Entity Framework 286

การเลือกดูข้อมูลบางฟิลด์โดยการทำ Anonymous Type 292

วิธีการกำหนดโครงสร้างข้อมูลขึ้นมาใช้งานด้วย Anonymous Type 296

การใช้งาน Anonymous Type ในฐานะแหล่งข้อมูลร่วมกับ Entity Framework 299

พื้นฐานการค้นหาข้อมูลด้วย Entity Framework 303

การแสดงข้อมูลร่วมกันตั้งแต่ 2 ตารางขึ้นไป 306

บทที่ 16 การจัดการข้อมูลในฐานข้อมูลด้วย Entity Framework 311

พื้นฐานการจัดการข้อมูลโดยอาศัย Entity Framework 311

การเพิ่มและลบข้อมูลแบบ 1 to Many (Master-Details) 322

การทำแบบฟอร์มป้อนข้อมูล (Data Entry) โดยอาศัย Entity Framework 328

□□□□□□.indd 6 10/9/2563 BE 15:18

บทที่ 17 พื้นฐานการพัฒนา Web Application ด้วย ASP.NET MVC 345
โครงสร้างพื้นฐานของหน้าเว็บเพจด้วยภาษา HTML5 345
วิธีการเข้ารหัสหน้าเว็บเพจ ให้สามารถแสดงผลภาษาไทย 346
พื้นฐานการสร้างโปรเจ็กต์แบบ ASP.NET MVC 347
การทดสอบรันโปรเจ็กต์ ASP.NET MVC 350
ทำความรู้จักกับโครงสร้างโปรเจ็กต์ของ ASP.NET MVC 353
ส่วนแสดงผลของโปรเจ็กต์ ASP.NET MVC 355

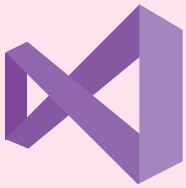
บทที่ 18 Controller และ Model 359
การสร้างคอนโทรลเลอร์ใหม่ในโปรเจ็กต์ ASP.NET MVC 359
การสร้าง Hyperlink ใน ASP.NET MVC 364
การส่งค่าให้กับเมธอดแบบมีพารามิเตอร์ 367
ทำงานกับส่วนของข้อมูล (Model) 369

บทที่ 19 การใช้งาน ASP.NET MVC กับระบบฐานข้อมูล SQL Server 375
การใช้งาน EF ร่วมกับฐานข้อมูล SQL Server ใน ASP.NET MVC 375
การแสดงข้อมูลจากฐานข้อมูล SQL Server ด้วย EF 379
การค้นหาข้อมูล 385

บทที่ 20 การจัดการข้อมูล (CRUD) ข้อมูลในฐานข้อมูล SQL Server ด้วย ASP.NET MVC 389
การสร้างโปรเจ็กต์ CRUD ใน ASP.NET MVC 389
การทำงานของโปรเจ็กต์ CRUD ของ ASP.NET MVC 408

บทที่ 21 การสร้างบริการ Back End ด้วย ASP.NET Web API 423
ASP.NET Web API คืออะไร 423

□□□□□□.indd 7 10/9/2563 BE 15:18



Visual Basic 2019

□□□□□□.indd 8 10/9/2563 BE 15:18

คุณสมบัติของ Form ที่สำคัญ

แสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
Name	ทำหน้าที่ตั้งชื่อให้กับฟอร์ม ใช้สำหรับอ้างอิงตอนเขียนโค้ด
Text	กำหนดข้อความที่จะแสดงในแถบ Title Bar
BackColor	กำหนดสีพื้นหลังให้กับฟอร์ม
ForeColor	กำหนดสีตัวอักษรที่อยู่ในฟอร์ม
FormBorderStyle	กำหนดรูปแบบเส้นขอบของฟอร์ม โดยที่ • **None** หมายถึง ไม่มีแถบไตเติลบาร์ • **FixedSingle** หมายถึง ปรับขนาดฟอร์มไม่ได้ แต่มีปุ่ม Minimize, Maximize และปุ่ม Close • **Fixed3D** หมายถึง กำหนดให้มีเส้นขอบแบบ FixedSingle แต่มีแนวลึกด้วย • **FixedDialog** หมายถึง กำหนดให้เป็นแบบ FixedSingle แต่ใช้ในรูปแบบไดอะล็อกโต้ตอบกับผู้ใช้งาน • **Sizable** หมายถึง แบบฟอร์มปรับขนาดได้ • **FixedToolWindow** หมายถึง แบบฟอร์มปรับขนาดไม่ได้ และไม่มีปุ่ม Minimize และปุ่ม Maximize • **SizableToolWindow** หมายถึง แบบฟอร์มปรับขนาดได้ และไม่มีปุ่ม Minimize และปุ่ม Maximize
MaximizeBox	กำหนดให้แบบฟอร์มแสดงปุ่มขยายขนาดฟอร์ม โดยที่ • **True** หมายถึง ให้มีปุ่ม Maximize มุมขวา-บน • **False** หมายถึง ซ่อนปุ่ม Maximize
MinimizeBox	กำหนดให้แบบฟอร์มแสดงปุ่มย่อฟอร์ม โดยที่ • **True** หมายถึง ให้มีปุ่ม Minimize มุมขวา-บน • **False** หมายถึง ซ่อนปุ่ม Minimize
IsMdiContainer	กำหนดให้ฟอร์มทำหน้าที่บรรจุฟอร์มอื่นๆ แบบหลายหน้าต่างได้หรือไม่ โดยที่ • **True** หมายถึง กำหนดให้ฟอร์มเป็นตัวบรรจุแบบ MDI • **False** หมายถึง กำหนดให้เป็นแบบฟอร์มปกติ
WindowState	กำหนดขนาดของฟอร์ม โดยที่ • **Normal** หมายถึง กำหนดให้ฟอร์มมีขนาดตามที่เราออกแบบ • **Minimized** หมายถึง กำหนดให้ฟอร์มมีขนาดย่อลงในแถบ Taskbar • **Maximized** หมายถึง กำหนดให้ฟอร์มมีขนาดเต็มหน้าจอส่วนแสดงผล

□□□□□ 2.indd 20 10/9/2563 BE 16:43

การแปลงตัวเลขที่อยู่ในฐานะข้อความ String ให้กลายเป็นข้อมูลชนิดตัวเลข

จากที่ผู้เขียนกล่าวไว้ในตอนต้นที่ว่า ตัวเลขที่อยู่ในฐานะข้อความไม่สามารถนำค่าไปคำนวณทางคณิตศาสตร์ หากต้องการนำไปคำนวณต้องแปลงตัวเลขดังกล่าวให้กลายเป็นข้อมูลชนิดตัวเลขเสียก่อน

ผู้เขียนมีข้อความ “586” เก็บอยู่ในตัวแปร str ต่อมาผู้เขียนต้องการแปลงตัวแปรตัวนี้ให้กลายเป็นข้อมูลชนิดตัวเลขจำนวนเต็ม Integer โดยอาศัยเมธอด **Parse()** ผลการแปลงที่ได้เก็บไว้ในตัวแปร num1

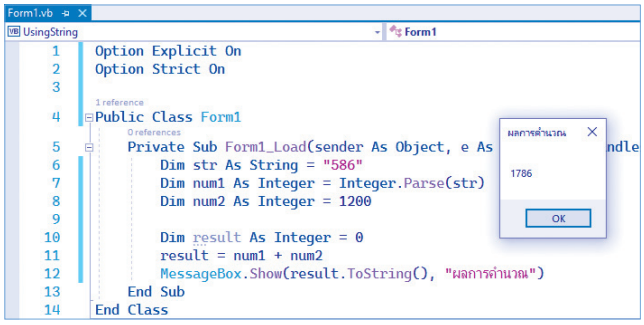
ผู้เขียนทดสอบนำไปใช้บวกค่ากับตัวแปร num2 พบว่าสามารถคำนวณทางคณิตศาสตร์ได้แล้ว

Form1.vb

```
Dim str As String = “586”
Dim num1 As Integer = Integer.Parse(str)
Dim num2 As Integer = 1200

Dim result As Integer = 0
result = num1 + num2
MessageBox.Show(result.ToString(), “ผลการคำนวณ”)
```

ทดสอบการทำงานได้ผลลัพธ์ดังนี้



รูปที่ 4-5 ผลการแปลงข้อมูลด้วยเมธอด Parse()

เมธอด Parse() ติดมากับข้อมูลชนิดพื้นฐานที่คุณเคยใช้งาน คุณสามารถใช้กับชนิดข้อมูลพื้นฐานอื่นๆ ได้อีกด้วย เช่น Double.Parse() หมายถึง ต้องการแปลงเป็นข้อมูลชนิดตัวเลขทศนิยม Double เป็นต้น

□□□□□ 4.indd 65 10/9/2563 BE 16:48

การแทนที่ข้อความด้วยเมธอด Replace()

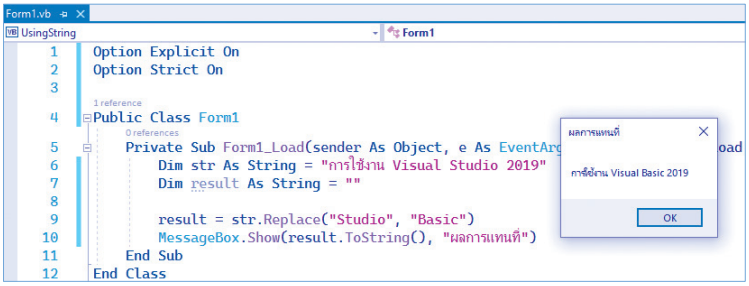
ผู้เขียนมีข้อความ “การใช้งาน Visual Studio 2019” เก็บอยู่ในตัวแปร str ต่อมามีการเรียกใช้เมธอด **Replace()** เข้ามาทำหน้าที่แทนที่ข้อความเดิม จากคำว่า “Studio” เปลี่ยนเป็นคำว่า “Basic” ดังโค้ดต่อไปนี้

Form1.vb

```
Dim str As String = “การใช้งาน Visual Studio 2019”
Dim result As String = “”

result = str.Replace(“Studio”, “Basic”)
MessageBox.Show(result.ToString(), “ผลการแทนที่”)
```

ทดสอบการทำงานได้ผลลัพธ์ดังนี้



รูปที่ 4-6 ผลการทำงานของเมธอด Replace()

การแบ่งข้อความด้วยฟังก์ชัน Split()

หลักการแบ่งข้อความในข้อมูลชนิด String ก็คือ คุณต้องบอกว่า “ใช้อะไรเป็นเงื่อนไขในการแบ่งข้อความ” ในกรณีนี้ผู้เขียนมีข้อความ “การใช้งาน Visual Studio 2019” เก็บอยู่ในตัวแปร str

ต่อมามีการใช้งานฟังก์ชัน Split() เข้ามาทำหน้าที่แบ่งข้อความที่เก็บอยู่ในตัวแปร str โดยใช้ช่องว่างเป็นเงื่อนไขในการแบ่งข้อความ ดังโค้ดต่อไปนี้

Form1.vb

```
Dim str As String = “การใช้งาน Visual Studio 2019”
Dim result As String() = str.Split(“ ”c)

MessageBox.Show(result(0) +
                Environment.NewLine + result(1) +
                Environment.NewLine + result(2), “ผลการแบ่งข้อความ”)
```


□□□□□ 4.indd 66 10/9/2563 BE 16:49

Form1.vb

```
    Private Sub lsbLanguage_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
lsbLanguage.SelectedIndexChanged
        Dim UserSelect As String = lsbLanguage.SelectedItem.ToString()
        MessageBox.Show(UserSelect, "รายการที่ถูกเลือกใน ListBox")
    End Sub
```

3. ท้ายที่สุด ส่วนวิธีการตรวจสอบรายการที่ถูกเลือกในคอนโทรล cboLanguage สามารถตรวจสอบผ่านทางคุณสมบัติ **SelectedItem**

Form1.vb

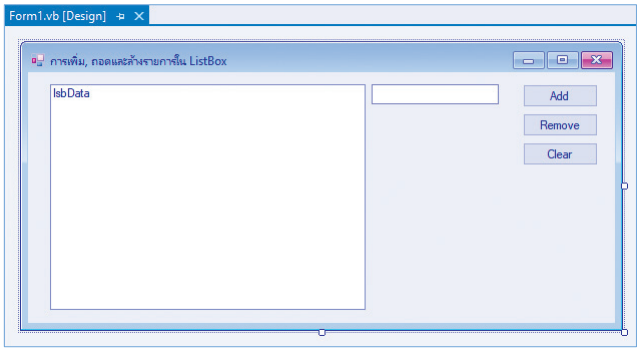
```
    Private Sub cboLanguage_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
cboLanguage.SelectedIndexChanged
        Dim UserSelect As String = cboLanguage.SelectedItem.ToString()
        MessageBox.Show(UserSelect, "รายการที่ถูกเลือกใน ComboBox")
    End Sub
```

การเพิ่ม, ถอด และล้างรายการใน ListBox

การทำงานกับรายการที่ปรากฏใน ListBox ประกอบด้วย การเพิ่ม, ถอด และล้างรายการนั่นเอง ให้คุณดูตัวอย่างต่อไปนี้

ตัวอย่างที่ 6-5 การเพิ่ม, ถอด และล้างรายการใน ListBox มีขั้นตอนดังนี้

1. ให้คุณออกแบบ Form1 ดังรูปที่ 6-19

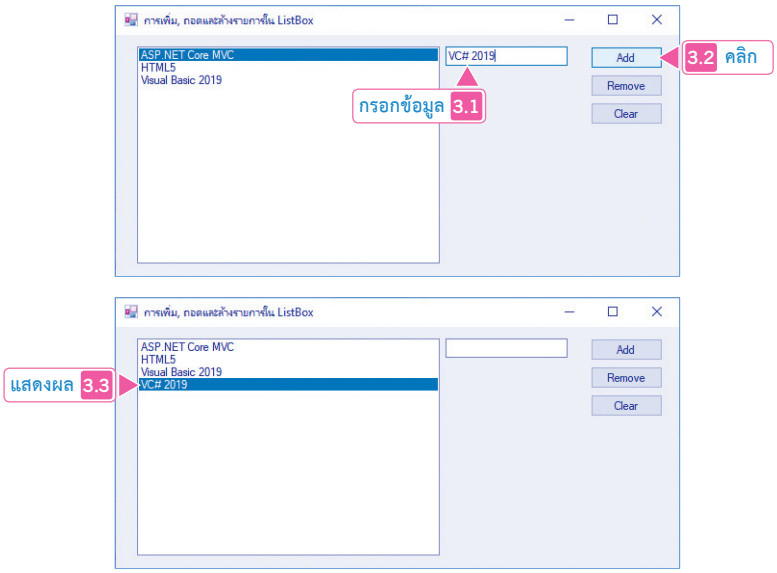


รูปที่ 6-19 แสดง Form1 ในขณะออกแบบ

จากรูปที่ 6-19 ให้คุณกำหนดคุณสมบัติต่างๆ ดังนี้

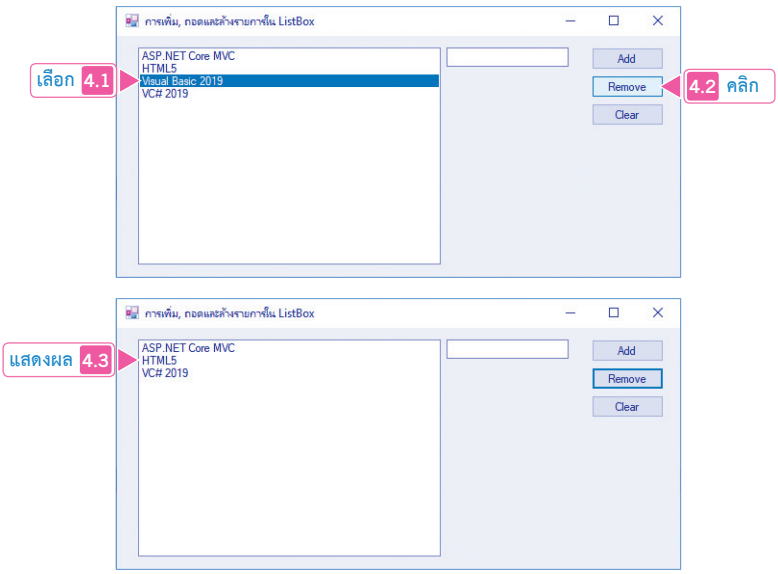
□□□□□ 6.indd 108 10/9/2563 BE 16:50

3. ท้ายที่สุดให้คุณรันโปรเจ็กต์การเพิ่มรายการใน lsbData ทำได้โดยการป้อนข้อมูลที่ต้องการ แล้วคลิกที่ปุ่ม Add เพื่อเพิ่มรายการ ดังรูปที่ 6-20



รูปที่ 6-20 กรณีเพิ่มรายการ

4. การถอดรายการ ให้คุณเลือกรายการที่ต้องการถอดก่อน จากนั้นคลิกที่ปุ่ม Remove เพื่อถอดรายการที่ถูกเลือก ดังรูปที่ 6-21



รูปที่ 6-21 กรณีถอดรายการ

□□□□□ 6.indd 110 10/9/2563 BE 16:51

Note

เห็นได้ว่า การทำ Overload Method แก้ปัญหาได้เพียงอย่างเดียว นั่นคือ คุณสามารถตั้งชื่อฟังก์ชันหรือเมธอดซ้ำกันได้ โดยที่ฟังก์ชันหรือเมธอดดังกล่าวต้องมี Signature ต่างกัน แต่ยังไม่ได้แก้ปัญหาที่ต้องเขียนโค้ดซ้ำ การแก้ปัญหาในลักษณะนี้ ต้องเขียนโปรแกรมแบบ Generic จะกล่าวถึงอีกครั้งในภายหลัง

ลักษณะของพารามิเตอร์ในฟังก์ชัน

องค์ประกอบส่วนหนึ่งของฟังก์ชันที่มีความสำคัญเป็นอย่างยิ่ง นั่นคือ พารามิเตอร์ (Parameter) หรืออาจจะเรียกอีกอย่างหนึ่งว่า **อาร์กิวเมนต์ (Argument)** ก็ได้ คุณจะได้ศึกษารายละเอียดปลีกย่อยที่จะทำให้คุณรู้จักกับพารามิเตอร์มากยิ่งขึ้น

พารามิเตอร์ต่างๆ ที่คุณนำมาใช้ในฟังก์ชัน บางครั้งจะเรียกว่า อาร์กิวเมนต์ (Argument) ก็ได้ เป็นเนื้อหาอีกส่วนที่น่าสนใจเป็นอย่างยิ่ง เพราะโดยปกติแล้วเราใช้พารามิเตอร์กันเสมอ เมื่อมีการสร้างเมธอดหรือฟังก์ชันขึ้นมาใช้งาน ดังนั้น พารามิเตอร์อยู่ใกล้ชิดกับเราพอสมควร เราจะดูว่ามีอะไรน่าสนใจเกี่ยวกับพารามิเตอร์บ้าง

พารามิเตอร์ ถ้าแบ่งตามพฤติกรรมของตัวมันเอง สามารถแบ่งได้ 2 รูปแบบ คือ

1. พารามิเตอร์ชนิดรับเข้า (Input) เป็นรูปแบบที่เราพบเห็นมากที่สุด
2. พารามิเตอร์ชนิดรับเข้าและส่งออกในเวลาเดียวกัน

พารามิเตอร์ชนิดรับเข้า ByVal (Input Parameter)

ให้คุณดูตัวอย่างโค้ดต่อไปนี้ประกอบคำอธิบาย

VB 2019

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
    Dim result As Integer
    result = AddData(5, 10)
End Sub

Private Function AddData(ByVal x As Integer, ByVal y As Integer) As Integer
    Return x + y
End Function
```

โค้ดข้างต้น ผู้เขียนสร้างฟังก์ชัน AddData() ขึ้นมา ทำหน้าที่บวกตัวเลข 2 ตัว ผู้ใช้งานต้องส่งค่าให้กับพารามิเตอร์ x และ y เข้ามาก่อน มีชนิดข้อมูลเป็นเลขจำนวนเต็ม Integer ผลบวกที่ได้คืนค่ากลับเป็นข้อมูลชนิด Integer เช่นกัน

□□□□□ 8.indd 143 10/9/2563 BE 16:51

คุณสมบัติของคอนโทรล LinkLabel ที่สำคัญ

คุณสมบัติอื่นๆ ที่น่าสนใจ มีดังนี้

คุณสมบัติ	หน้าที่
Text	ข้อความที่แสดงอยู่ในตัวคอนโทรล LinkLabel
AutoSize	กำหนดขนาดของคอนโทรล LinkLabel มี 2 ค่า คือ • True หมายถึง ให้มีขนาดตามข้อความที่แสดงอยู่ • False หมายถึง กำหนดขนาดได้อย่างอิสระ
BackColor	กำหนดสีพื้นหลังให้กับคอนโทรล LinkLabel
BorderStyle	กำหนดให้แสดงเส้นขอบหรือไม่ มี 3 แบบ คือ • None หมายถึง ไม่ต้องแสดงเส้นขอบ • FixedSingle หมายถึง ให้แสดงเส้นขอบ • Fixed3D หมายถึง ให้แสดงเส้นขอบแบบมีส่วนลึกด้วย
DisabledLinkColor	กำหนดสีในกรณีที่ลิงค์ถูกยกเลิก
Font	กำหนดรูปแบบฟอนต์ที่จะใช้แสดงข้อความ
ForeColor	กำหนดสีข้อความ
TextAlign	กำหนดตำแหน่งของข้อความ มี 9 ตำแหน่ง กล่าวคือ แยกออกเป็น 3 แถว คือ บน, กลาง, ล่าง โดยที่ในแต่ละแถวยังแบ่งออกเป็นอีก 3 ตำแหน่ง คือ ซ้าย, กลาง และขวา
LinkBehavior	กำหนดรูปแบบของไฮเปอร์ลิงค์ มี 4 รูปแบบ คือ • SystemDefault หมายถึง ใช้รูปแบบปกติ (ขีดเส้นใต้ + ตัวอักษรเป็นสีน้ำเงิน) • AlwaysUnderline หมายถึง ให้แสดงด้วยข้อความแบบมีเส้นขีดเส้นใต้กำกับไว้ • HoverUnderline หมายถึง เมื่อผู้ใช้งานเอาเมาส์ไปลอยอยู่เหนือไฮเปอร์ลิงค์แล้วแสดงเส้นขีดเส้นใต้ • NeverUuderline หมายถึง ไม่ต้องแสดงเป็นตัวอักษรแบบมีเส้นขีดเส้นใต้
LinkColor	สีของข้อความที่เป็นไฮเปอร์ลิงค์ สีเริ่มต้น คือ สีน้ำเงิน
LinkVisited	กำหนดให้แสดงลิงค์แบบเคยคลิกแล้วหรือไม่ (สีของไฮเปอร์ลิงค์เป็นสีที่แตกต่างจากสีปกติ) • True หมายถึง ให้ใช้ • False หมายถึง ไม่ให้ใช้
VisitedLinkColor	สีของข้อความไฮเปอร์ลิงค์ที่เคยคลิกมาแล้วอย่างน้อย 1 ครั้ง สีเริ่มต้น คือ สีม่วง
Visible	กำหนดการปรากฏของคอนโทรล LinkLabel กล่าวคือ • True หมายถึง ให้แสดงคอนโทรล LinkLabel ตามปกติ • False หมายถึง ให้ซ่อนตัวคอนโทรล LinkLabel

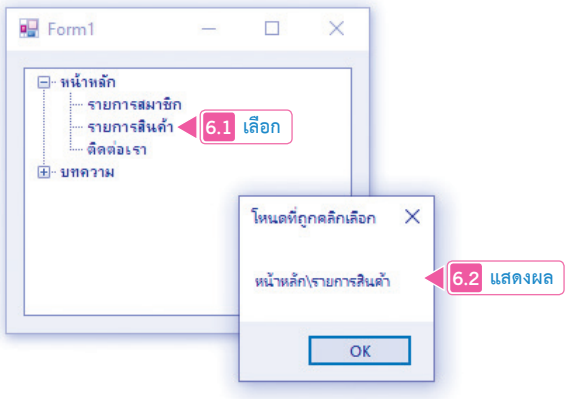
□□□□□ 11.indd 196 10/9/2563 BE 16:52

5. การทำงานที่สำคัญที่สุดของ TreeView ก็คือ วิธีการตรวจสอบว่าโหนดใดถูกคลิกเลือกนั่นเอง ให้คุณเขียนโค้ดดังต่อไปนี้

โค้ด VB 2019 ที่ 11-4 การแสดงข้อมูลแบบลำดับชั้นด้วยคอนโทรล TreeView (Form1.vb)

```
        Private Sub TreeView1_AfterSelect(sender As Object, e As TreeViewEventArgs) Handles
TreeView1.AfterSelect
            If (e.Action = TreeViewAction.ByMouse) Then
                Dim userSelected As String = Nothing
                userSelected = e.Node.FullPath
                MessageBox.Show(userSelected, "โหนดที่ถูกคลิกเลือก")
            End If
        End Sub
```

6. ทดสอบการทำงานได้ผลลัพธ์ดังนี้



รูปที่ 11-16 ผลการรันตัวอย่างที่ 11-4

จากรูปที่ 11-16 เมื่อผู้ใช้งานคลิกเลือกโหนดใดก็ตาม ก็จะมีการแสดงข้อความแจ้งบอกพาธที่ถูกเลือกไว้ด้วย

อธิบายการทำงานของโค้ด

1. เราจะใช้เหตุการณ์ **AfterSelect()** เข้ามาทำหน้าที่ตรวจสอบโหนดที่ถูกคลิกเลือก ผ่านทางเมธอด **Action()** ของพารามิเตอร์ e ในกรณีนี้เราสนใจเฉพาะโหนดที่ถูกคลิกด้วยเมาส์เท่านั้น

Form1.vb

```
        Private Sub TreeView1_AfterSelect(sender As Object, e As TreeViewEventArgs) Handles
TreeView1.AfterSelect
            If (e.Action = TreeViewAction.ByMouse) Then
```


□□□□□ 11.indd 202 10/9/2563 BE 16:54

2. ผู้เขียนสร้างตัวแปรที่ชื่อว่า userSelected ขึ้นมา ทำหน้าที่เก็บพาธของโหนดที่ถูกเลือก (**userSelected = e.Node.FullPath**) และแสดงพาธที่ได้ให้ผู้ใช้งานทราบ

Form1.vb

```
        Dim userSelected As String = Nothing
        userSelected = e.Node.FullPath
        MessageBox.Show(userSelected, "โหนดที่ถูกคลิกเลือก")
    End If
End Sub
```

คุณสมบัติของคอนโทรล TreeView ที่สำคัญ

คุณสมบัติอื่นๆ ที่น่าสนใจ มีดังนี้

คุณสมบัติ	หน้าที่
CheckBoxs	ให้แสดงแต่ละโหนดในรูปแบบตัวเลือก CheckBox
HotTracking	ให้ดักจับเมาส์ลอยอยู่เหนือแต่ละโหนด เพื่อแสดงว่าโหนดใดกำลังได้รับโฟกัสอยู่ โดยที่ • True หมายถึง ให้ดักจับ • False หมายถึง ไม่ต้องดักจับ
Indent	กำหนดระยะระหว่างโหนดแม่กับโหนดลูก
ItemHeight	กำหนดความสูงของแต่ละโหนด
LabelEdit	ยินยอมให้ผู้ใช้งานแก้ไขข้อความแต่ละโหนดหรือไม่ โดยที่ • True หมายถึง ยินยอม • False หมายถึง ไม่ยินยอม
PathSeparator	กำหนดตัวอักษรคั่นระหว่างโหนดแม่กับโหนดลูก ตัวอักษรเริ่มต้น คือ \
Scrollable	กำหนดให้แสดงแถบเลื่อนภายในคอนโทรล TreeView หรือไม่ โดยที่ • True หมายถึง ให้แสดง • False หมายถึง ไม่ต้องแสดง
ShowLines	กำหนดให้แสดงเส้นระดับชั้นระหว่างโหนดหรือไม่ โดยที่ • True หมายถึง ให้แสดง • False หมายถึง ไม่ต้องแสดง
ShowPlusMinus	กำหนดให้แสดงเครื่องหมาย + - หน้าโหนดแม่หรือไม่ โดยที่ • True หมายถึง ให้แสดง • False หมายถึง ไม่ต้องแสดง

□□□□□ 11.indd 203 10/9/2563 BE 16:54

ทำงานกับไฟล์ข้อความด้วยคอนโทรล RichTextBox, OpenFileDialog และ SaveFileDialog

การทำงานกับไฟล์ข้อความขั้นต้น (*.txt หรือ *.rtf) อยู่ในความรับผิดชอบของคอนโทรลอย่างน้อย 3 ตัว คือ

- **RichTextBox** ทำหน้าที่แสดงเนื้อหาที่ได้จากไฟล์ข้อความ
- **OpenFileDialog** ทำหน้าที่สร้างไดอะล็อกเปิดไฟล์
- **SaveFileDialog** ทำหน้าที่สร้างไดอะล็อกบันทึกไฟล์

ก่อนที่จะเข้าสู่ตัวอย่างใช้งาน ผู้เขียนขอแนะนำคุณสมบัติขั้นต้นของคอนโทรลทั้ง 3 ตัวก่อน

คุณสมบัติของคอนโทรล RichTextBox ที่สำคัญ

แสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
MaxLenght	กำหนดจำนวนตัวอักษรที่จะแสดงในตัวคอนโทรล RichTextBox ไม่เกิน 2,147,483,647 ตัวอักษร
Multiline	กำหนดให้แสดงข้อมูลได้หลายบรรทัด ใช่หรือไม่ • True หมายถึง ให้แสดงได้หลายบรรทัด • False หมายถึง ให้แสดงบรรทัดเดียว
ScrollBars	กำหนดให้แสดงแถบเลื่อนข้อมูล

คุณสมบัติของคอนโทรล OpenFileDialog ที่สำคัญ

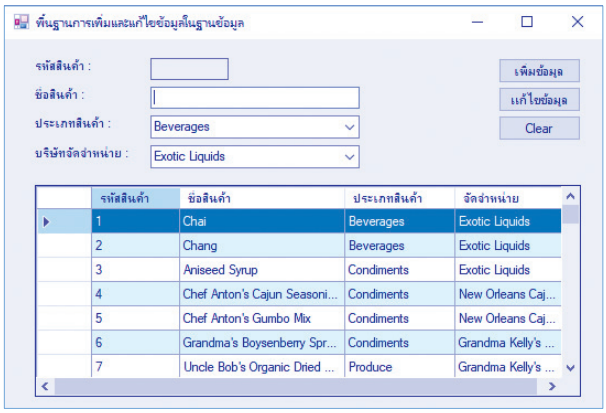
แสดงดังตารางต่อไปนี้

คุณสมบัติ	หน้าที่
CheckFileExists	ทำหน้าที่ตรวจสอบว่าไฟล์ที่กำลังเปิดมีอยู่หรือไม่
CheckPathExists	ทำหน้าที่ตรวจสอบว่ามีพาธที่กำลังเลือกอยู่หรือไม่
Filter	กำหนดนามสกุลไฟล์ที่มองเห็น
InitialDirectory	กำหนดพาธเริ่มต้น เช่น C:\
Multiselect	กำหนดให้เปิดได้หลายไฟล์ ใช่หรือไม่ • True หมายถึง เปิดได้หลายไฟล์ • False หมายถึง เปิดได้ไฟล์เดียว

□□□□□ 11.indd 223 10/9/2563 BE 16:54

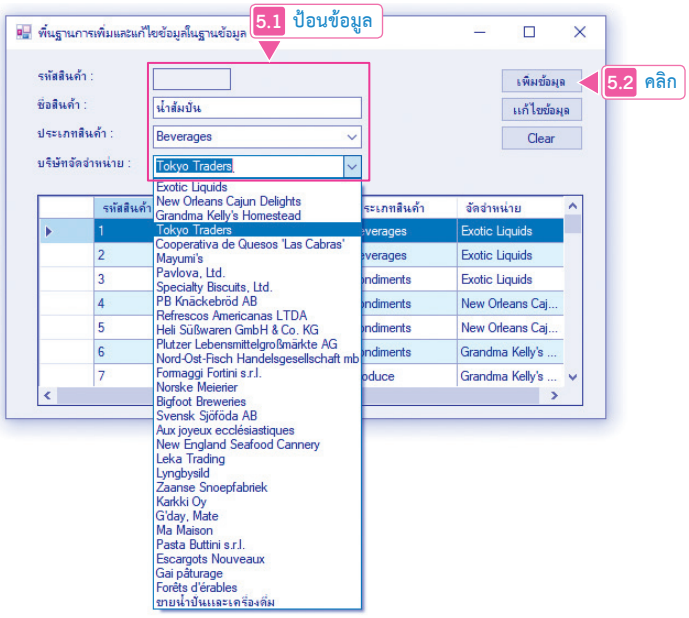
```
    End Sub
    Private Sub cmdClear_Click(sender As Object, e As EventArgs) Handles cmdClear.Click
        ClearAllData()
        txtProductName.Focus()
    End Sub
End Class
```

4. ทดสอบการทำงานได้ผลลัพธ์ดังนี้

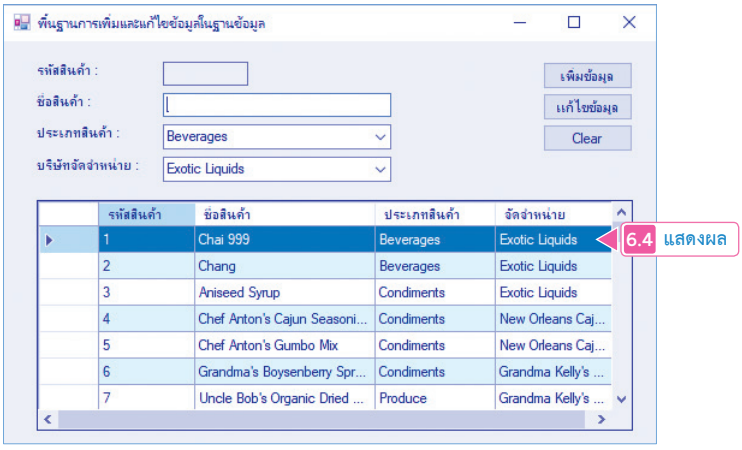


รูปที่ 16-3 ผลการรันตัวอย่างที่ 16-1

5. **การเพิ่มข้อมูลสินค้าใหม่** ขอให้คุณป้อนข้อมูลให้ครบ จากนั้นคลิกที่ปุ่ม เพิ่มข้อมูล เพื่อเพิ่มข้อมูลสินค้า ดังรูปที่ 16-4



□□□□□ 16.indd 317 10/9/2563 BE 16:57



รูปที่ 16-5 แสดงการแก้ไขข้อมูลสินค้า

อธิบายการทำงานของโค้ด

1. ในเหตุการณ์ Form_Load() คิวรีรายชื่อประเภทสินค้าออกมาจากคุณสมบัติ Categories และคิวรีรายชื่อบริษัทจัดจำหน่ายออกมาจากคุณสมบัติ Suppliers แสดงเป็นรายการให้ผู้ใช้เลือกในคอนโทรล ComboBox ทั้ง 2 ตัว

Form1.vb

```
Dim cs = From c In db.Categories
If cs.Count() > 0 Then
    With cboCategory
        .BeginUpdate()
        .DisplayMember = "CategoryName"
        .ValueMember = "CategoryID"
        .DataSource = cs.ToList()
        .EndUpdate()
    End With
End If

Dim ss = From s In db.Suppliers
If ss.Count() > 0 Then
    With cboSupplier
        .BeginUpdate()
        .DisplayMember = "CompanyName"
        .ValueMember = "SupplierID"
        .DataSource = ss.ToList()
        .EndUpdate()
    End With
End If
```

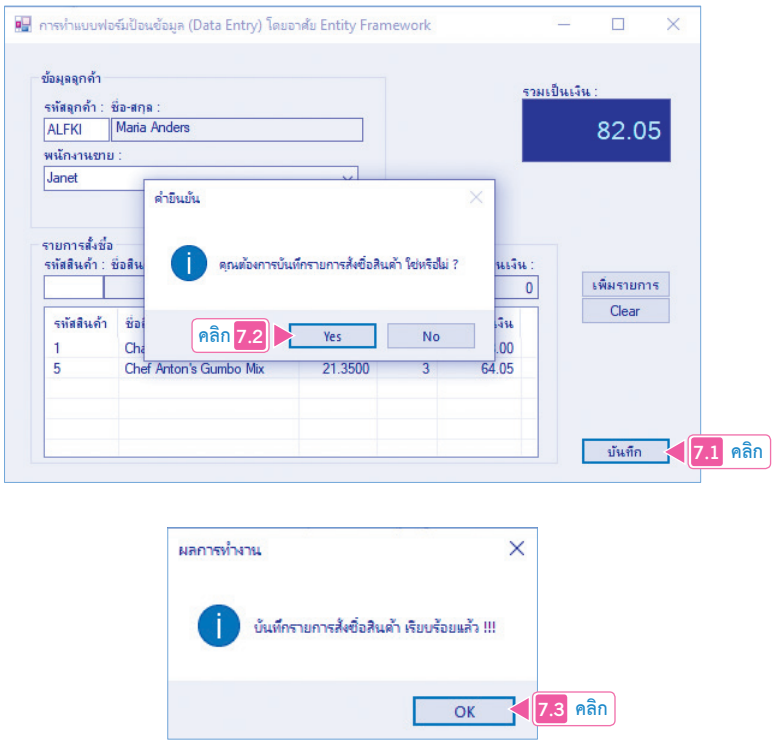

□□□□□ 16.indd 319 10/9/2563 BE 16:57

5. ให้ป้อนรหัสสินค้าและจำนวนที่สั่งซื้อ แล้วคลิกที่ปุ่ม เพิ่มรายการ เพื่อเพิ่มรายการสินค้าที่สั่งซื้อ ดังรูปที่ 16-13

รูปที่ 16-13 แสดงการป้อนรายการสินค้าที่สั่งซื้อ

□□□□□ 16.indd 337 10/9/2563 BE 16:58

7. ท้ายที่สุด คลิกที่ปุ่ม บันทึก เพื่อบันทึกการสั่งซื้อสินค้า ดังรูปที่ 16-15



รูปที่ 16-15 แสดงการบันทึกรายการสั่งซื้อสินค้า

อธิบายการทำงานของโค้ด

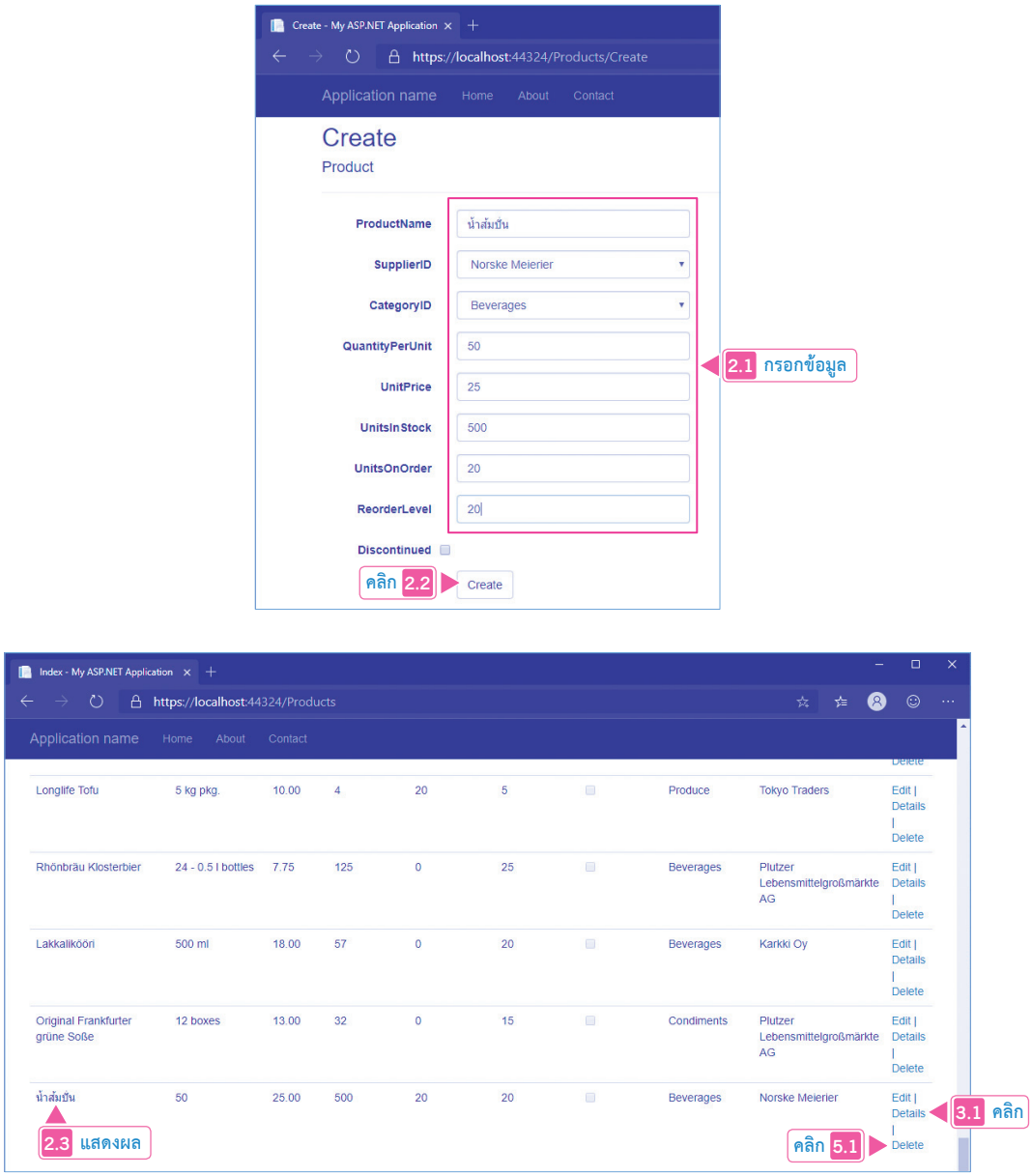
1. ในเหตุการณ์ Form_Load() ให้คิวรีข้อมูลพนักงานออกมาจากคุณสมบัติ Employees ของออบเจ็กต์ db เก็บไว้ที่ตัวแปร es และสั่งให้แสดงในคอนโทรล ComboBox ที่ชื่อว่า cboEmployee

Form1.vb

```
Dim es = From em In db.Employees
         Select em.EmployeeID, em.FirstName
With cboEmployee
   .BeginUpdate()
   .DisplayMember = "FirstName"
   .ValueMember = "EmployeeID"
   .DataSource = es.ToList()
   .EndUpdate()
End With
```


□□□□□ 16.indd 339 10/9/2563 BE 16:58

4. **การแก้ไขข้อมูลสินค้า** เกิดจากการคลิกที่ลิงค์ Edit ก็จะเข้าสู่แบบฟอร์มแก้ไขข้อมูล ดังรูปที่ 20-8
5. **การลบข้อมูล** เกิดจากการคลิกที่ลิงค์ Delete เป็นขั้นตอนการลบข้อมูลสินค้า 1 รายการที่ไม่เกี่ยวข้องกับตารางอื่นเท่านั้น ในกรณีนี้น้ำปั่นเป็นสินค้าใหม่เพิ่งถูกเพิ่มเข้ามา คุณสามารถลบข้อมูลสินค้ารายการนี้ได้เลย ดังรูปที่ 20-9



รูปที่ 20-6 กรณีเพิ่มสินค้าใหม่

□□□□□ 20.indd 409 10/9/2563 BE 16:58

อธิบายการทำงานของโค้ด

1. การแสดงข้อมูลสินค้าเริ่มต้นที่เมธอด Index() ของคอนโทรลเลอร์ ProductsController ทำหน้าที่คิวรีข้อมูลสินค้าโดยอาศัยฟังก์ชัน Include() เข้ามาทำหน้าที่เชื่อมโยงความสัมพันธ์ทั้ง 3 ตาราง คือ ตาราง Product (db.Products) เป็นข้อมูลหลัก, ตารางประเภทสินค้าจากตาราง Categories (Function(p) p.Category) และตารางบริษัทจัดจำหน่ายสินค้าจากตาราง Suppliers (Function(p) p.Supplier)

ข้อมูลที่ได้มาเก็บไว้ในตัวแปรที่ชื่อว่า products และส่งข้อมูลนี้ไปที่ส่วนแสดงผลโดยอาศัยฟังก์ชัน View()

\Controllers\ProductsController.vb

```
Function Index() As ActionResult
Dim products = db.Products.Include(Function(p) p.Category).Include(Function(p) p.Supplier)

    Return View(products.ToList())
End Function
```

2. ส่วนแสดงผลที่ทำหน้าที่แสดงรายการสินค้าแบบไม่มีเงื่อนไข เป็นหน้าที่ของไฟล์ Index.vbhtml โดยการอ้างอิงชนิดข้อมูลที่ต้องทำงานในส่วนแสดงผลนี้ก่อน นั่นคือ คลาส Product

\Views\Products\Index.vbhtml

```
@ModelType IEnumerable(Of BasicCRUD.BasicCRUD.Product)
```

3. เริ่มต้นสร้างตารางด้วยอีลีเม็นต์ <table> สร้างแต่ละคอลัมน์ด้วยอีลีเม็นต์ <th>...</th> จากนั้นกำหนดข้อความในแต่ละคอลัมน์ด้วยคุณสมบัติต่างๆ ของคลาส Product ผ่านทางตัวแปรออบเจ็กต์ model ซึ่งทำหน้าที่แทนคลาส Product ก็จะได้หัวตารางมาก่อน

\Views\Products\Index.vbhtml

```
<table class="table">
  <tr>
    <th>
      @Html.DisplayNameFor(Function(model) model.ProductName)
    </th>
    <th>
      @Html.DisplayNameFor(Function(model) model.QuantityPerUnit)
    </th>
    <th>
      @Html.DisplayNameFor(Function(model) model.UnitPrice)
```


□□□□□ 20.indd 412 10/9/2563 BE 16:59