

## บทที่ 4

# คอมโพเนนต์ของเดสก์ท็อป

### 4.1 ความนำ

ในการพัฒนาแอปพลิเคชันด้วยโปรแกรมเดสก์ท็อป การใช้คอมโพเนนต์เป็นส่วนที่สำคัญ เพราะนอกจากจะใช้เป็นส่วนที่ติดต่อกับผู้ใช้ (User Interface) ยังใช้ในการควบคุมการทำงานของแอปพลิเคชันอีกด้วย ดังนั้น ถ้ารู้วิธีการใช้คอมโพเนนต์ได้อย่างเหมาะสมจะช่วยให้การพัฒนาแอปพลิเคชันทำให้ง่ายขึ้น

เนื่องจากเดสก์ท็อปมีคอมโพเนนต์จำนวนมาก ดังนั้นจึงมีการจัดคอมโพเนนต์เป็นกลุ่ม ๆ โดยพิจารณาจากความจำเป็นขั้นพื้นฐานในการใช้งาน วัตถุประสงค์ในการใช้งาน หรือความซับซ้อนในการใช้งาน อย่างไรก็ตาม เนื่องจากคอมโพเนนต์มีคุณสมบัติเชิงวัตถุ จึงสามารถทำการควบคุมการทำงานของคอมโพเนนต์ผ่านทาง การกำหนดคุณสมบัติ (Properties) พฤติกรรม (Method) และเหตุการณ์ (Events) จึงมีความจำเป็นต้องเรียนรู้ก่อนการใช้งานคอมโพเนนต์

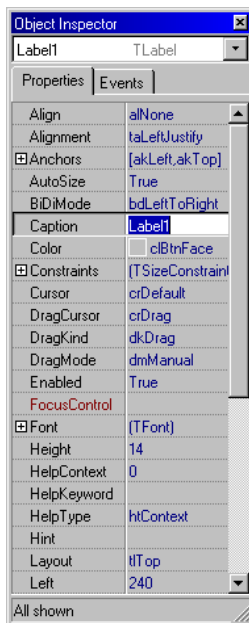
### 4.2 การควบคุมการทำงานของคอมโพเนนต์

#### 4.2.1 คุณสมบัติ (Properties)

คือส่วนที่ใช้ในการกำหนดลักษณะของคอมโพเนนต์ที่ต้องการ เช่น คุณสมบัติของ Label ที่ต้องการให้ปรากฏข้อความโดยมีรูปแบบอักษร (Font) แบบหนึ่ง ๆ ซึ่งคุณสมบัติเช่นนี้ จะช่วยให้สามารถพัฒนาแอปพลิเคชันได้สะดวก โดยสามารถแก้ไขค่าคุณสมบัติของคอมโพเนนต์ได้โดยไม่ต้องเขียนโปรแกรม และเห็นผลลัพธ์ของการกำหนดค่านั้นทันทีที่ทางจอภาพ ซึ่งเป็นข้อเด่นของโปรแกรมภาษาภาพ และช่วยป้องกันการเกิดข้อผิดพลาดได้อีกวิธีหนึ่ง สำหรับการกำหนดค่าคุณสมบัติสามารถทำได้ 2 วิธี คือ

- กำหนดค่าคุณสมบัติขณะออกแบบ

การกำหนดค่าคุณสมบัติด้วยวิธีนี้ สามารถดำเนินการผ่านออปเจ็คอินสเป็คเตอร์ดังรูป



1. โดยเลือกคอมโพเนนท์หนึ่ง ๆ ด้วยการคลิก และไปยังออปเจ็คอินสเป็คเตอร์ที่มีการกำหนดค่าปกติให้แก่คอมโพเนนท์ดังกล่าวแล้ว
2. ผู้พัฒนาสามารถแก้ไขช่องทางด้านขวาได้ตามรายการคุณสมบัติบรรทัดนั้น ๆ สำหรับบางคุณสมบัติสามารถเลือกค่าจากรายที่มีให้ เช่น Autosize มีรายการ True หรือ False ให้เลือก และบางคุณสมบัติที่มีเครื่องหมาย + นำหน้าชื่อคุณสมบัติ แสดงว่ามีคุณสมบัติย่อยให้ ผู้พัฒนาคลิกที่คุณสมบัตินั้น จะปรากฏรายละเอียดของคุณสมบัติย่อยให้เรากำหนด เช่น BorderIcon, Options เป็นต้น

- กำหนดค่าคุณสมบัติในขณะรันแอปพลิเคชัน

การกำหนดค่าคุณสมบัติที่ต้องการด้วยวิธีนี้ ต้องเปลี่ยนแปลงค่าในขณะรันแอปพลิเคชัน โดยที่มีรูปแบบการกำหนดโครงสร้างภาษาดังนี้

[ชื่อคอมโพเนนท์].[ชื่อคุณสมบัติ] := ค่าที่ต้องการกำหนด;

ตัวอย่างเช่น

Label1.Caption := 'สวัสดี...ค่ะ';

Edit1.Text := 'สวัสดี...ครับ';

Panel1.Color := clBlue;

#### 4.2.2 พฤติกรรม (Method)

คือลักษณะของฟังก์ชันแบบหนึ่งที่ใช้ประกอบการทำงานของคอมโพเนนต์นั้น ๆ ในการเรียกใช้ Methods ก็เหมือนกับการเรียกใช้ฟังก์ชัน คือสามารถส่งตัวแปรไปและคอยรับค่าได้จากการคำนวณกลับมา โดยที่พฤติกรรมดังกล่าวจะสามารถเรียกใช้ได้เฉพาะขณะรันแอปพลิเคชันเท่านั้น ซึ่งมีรูปแบบการเรียกใช้พฤติกรรม ดังนี้

รูปแบบ

```
[ชื่อคอมโพเนนต์].[ชื่อพฤติกรรม]([ค่าพารามิเตอร์ที่ต้องการส่ง]);
```

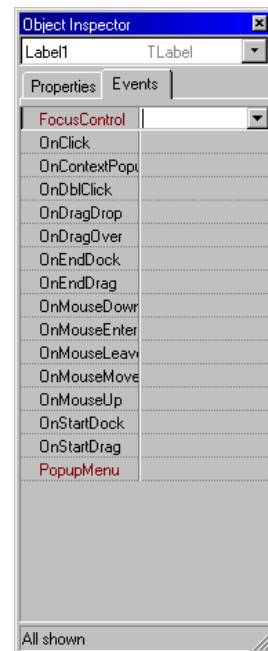
ตัวอย่าง

```
Label1.Clear;
```

```
Label1.Refresh;
```

#### 4.2.3 เหตุการณ์ (Events)

คือส่วนของคอมโพเนนต์ที่ใช้ในการรับการตอบสนองต่อการกระทำอย่างใดอย่างหนึ่งที่เกิดขึ้นกับคอมโพเนนต์ เช่น เมื่อทำการคลิกที่คอมโพเนนต์ Label1 จะเกิดเหตุการณ์ OnClick ของคอมโพเนนต์ Label1 ขึ้น ซึ่งในบางเหตุการณ์อาจมีการส่งค่าพารามิเตอร์ที่เป็นข้อมูลที่เกี่ยวข้องกับเหตุการณ์ดังกล่าวมาให้ และสามารถตรวจสอบได้ว่าคอมโพเนนต์นั้น ๆ มีความสามารถในการเกิดเหตุการณ์ได้บ้างโดยดูที่แท็บเหตุการณ์ของหน้าต่างออปเจ็คอินสเป็คเตอร์



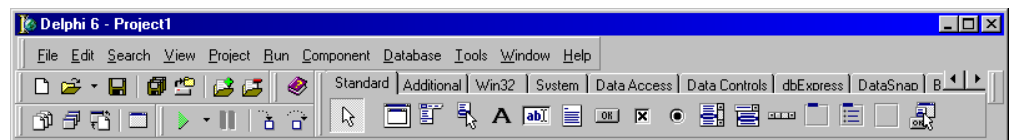
### 4.3 คอมโพเนนต์ (Component)

คอมโพเนนต์ต่าง ๆ ที่อยู่ในคอมโพเนนต์พาเลตจะถูกจัดแยกเป็นกลุ่ม ๆ เพื่อให้สะดวกต่อการใช้งาน โดยมีความหมายของแต่ละกลุ่มดังนี้

<ul style="list-style-type: none"> <li>▪ Standard</li> </ul>	กลุ่มของคอมโพเนนต์มาตรฐาน ใช้ในการพัฒนาแอปพลิเคชันบน Windows ทั่วไป
<ul style="list-style-type: none"> <li>▪ Additional</li> </ul>	กลุ่มของคอมโพเนนต์ที่เพิ่มเติมจากกลุ่ม Standard จะช่วยให้แอปพลิเคชันบน Windows มีความหลากหลายขึ้น
<ul style="list-style-type: none"> <li>▪ Win32</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้ได้เฉพาะระบบปฏิบัติการ Windows 95-98 หรือ NT4
<ul style="list-style-type: none"> <li>▪ System</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้สำหรับควบคุมการทำงานเกี่ยวกับระบบ
<ul style="list-style-type: none"> <li>▪ Data Access</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้ติดต่อกับข้อมูลในฐานข้อมูล
<ul style="list-style-type: none"> <li>▪ Data Controls</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้แสดงผลและทำงานกับข้อมูลที่อยู่ในฐานข้อมูล
<ul style="list-style-type: none"> <li>▪ ADO</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้ติดต่อกับฐานข้อมูล
<ul style="list-style-type: none"> <li>▪ InterBase</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้ติดต่อกับ InterBase โดยไม่ต้องใช้ BDE หรือ ADO
<ul style="list-style-type: none"> <li>▪ Midas</li> </ul>	กลุ่มของคอมโพเนนต์ที่ช่วยในการสร้างแอปพลิเคชันเพื่อติดต่อกับฐานข้อมูลแบบ Multi-tier
<ul style="list-style-type: none"> <li>▪ InternetExpress</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้สร้างเว็บเซิร์ฟเวอร์แอปพลิเคชันและ Client แบบ Multi-tier
<ul style="list-style-type: none"> <li>▪ Internet</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้สร้างเว็บเซิร์ฟเวอร์แอปพลิเคชัน
<ul style="list-style-type: none"> <li>▪ FastNet</li> </ul>	กลุ่มของคอมโพเนนต์ที่ใช้เป็นโปรโตคอลในการเข้าถึงข้อมูลบนอินเทอร์เน็ต
<ul style="list-style-type: none"> <li>▪ Decision Cube</li> </ul>	กลุ่มของคอมโพเนนต์ที่มีเฉพาะในชุด Enterprise ใช้สำหรับการวิเคราะห์ข้อมูลเพื่อการตัดสินใจ

▪ Qreport	กลุ่มของคอมโพเนนท์ที่ใช้สำหรับสร้างรายงานพิมพ์
▪ Dialogs	กลุ่มของคอมโพเนนท์ที่ใช้แสดง Dialog Box มาตรฐาน
▪ Win3.1	กลุ่มของคอมโพเนนท์ที่ใช้กับระบบปฏิบัติการ Window 3.1
▪ Samples	กลุ่มของคอมโพเนนท์ตัวอย่างที่ถูกสร้างขึ้นเพื่อใช้งานเฉพาะอย่าง เช่น Calendar เป็นต้น
▪ ActiveX	กลุ่มของคอมโพเนนท์ประเภท ActiveX ที่นำมาใช้กับ Delphi ได้
▪ Server	กลุ่มของคอมโพเนนท์ใหม่ที่ใช้ติดต่อกับแอปพลิเคชันในรูปแบบของ COM Server

#### ▪ Standard



#### A Label : ป้ายข้อความ

ใช้สำหรับแสดงข้อความเพียงอย่างเดียว ไม่สามารถรับค่าจากผู้ใช้ได้ โดยส่วนใหญ่ มักใช้ในการแสดงคำอธิบาย เช่น แสดงความหมายของค่าที่ต้องการรับเข้ามา แสดงหมายเหตุ แสดงหัวข้อเรื่อง เป็นต้น

#### คุณสมบัติที่สำคัญ (Properties)

Caption สำหรับกำหนดข้อความที่ต้องการแสดงบน Label

Alignment สำหรับกำหนดการจัดเรียงข้อความตามแนวนอน มี 3 แบบให้เลือก คือ

- กำหนดข้อความชิดซ้าย (taLeftJustify)
- กำหนดข้อความตรงกลาง (taCenter)
- กำหนดข้อความชิดขวา (taRightJustify)

Layout	สำหรับกำหนดการจัดเรียงข้อความตามแนวตั้ง มี 3 แบบให้เลือก คือ <ul style="list-style-type: none"> <li>- กำหนดให้ข้อความอยู่ด้านบน (tlTop)</li> <li>- กำหนดให้ข้อความอยู่ตรงกลาง (tlCenter)</li> <li>- กำหนดให้ข้อความอยู่ด้านล่าง (tlBottom)</li> </ul>
Color	สำหรับกำหนดสีพื้นของป้ายข้อความ (ดับเบิ้ลคลิก!!)
Font	สำหรับกำหนดแบบตัวอักษรและสีตัวอักษรในป้ายข้อความ (ดับเบิ้ลคลิก!!)
Transparent	สำหรับกำหนดให้พื้นหลังของข้อความโปร่งใส
Visible	สำหรับกำหนดการแสดงผลของป้ายข้อความ (True แสดง/False ไม่แสดง !!)
Hints	สำหรับกำหนดข้อความหมายเหตุ (Comments) พิเศษเมื่อรัน
Showhint	สำหรับควบคุมการแสดงผลข้อความหมายเหตุ (True แสดง/False ไม่แสดง !!)

### พฤติกรรมที่สำคัญ (Methods)

Clear	สำหรับลบข้อความที่อยู่ใน Label เช่น Label1.Clear;
Caption	สำหรับกำหนดค่าให้กับ Label (เฉพาะข้อความ) เช่น Label.Caption := '5';



Edit : แถบรับ/แสดงข้อความ

ใช้สำหรับสร้างส่วนติดต่อกับผู้ใช้ ซึ่งสามารถใช้แสดงข้อความและรับข้อมูลจากผู้ใช้  
ป้อนเข้ามา

### คุณสมบัติที่สำคัญ (Properties)

Text	สำหรับเก็บข้อความที่อยู่ใน Edit
<i>Color , Font, Transparent, Visible, Hints, Showhint (เช่นเดียวกับ Label)</i>	

### เหตุการณ์ที่สำคัญ (Events)

OnChange	กำหนดให้ดำเนินการ เมื่อข้อความใน Edit เกิดการเปลี่ยนแปลง
OnEnter	กำหนดให้ดำเนินการ เมื่อมีการย้ายการทำงานเข้าไปใน Edit
OnExit	กำหนดให้ดำเนินการ เมื่อมีการย้ายการทำงานออกไปจาก Edit
OnKeyPress	กำหนดให้ดำเนินการ เมื่อมีการกดคีย์

(ข้อความระวัง !! ควรกำหนดคีย์ที่ต้องการตรวจสอบ เช่น กดคีย์ Enter (#13, กดคีย์ Esc (#27) เป็นต้น หากไม่กำหนดคีย์จะหมายความว่า กดคีย์ใด ๆ ก็ถูกดำเนินการได้หมด)

### พฤติกรรมที่สำคัญ (Methods)

Clear	สำหรับลบข้อความที่อยู่ใน Edit เช่น Edit1.Clear;
Text	สำหรับกำหนดค่าให้กับ Edit (เฉพาะข้อความ) เช่น Edit1.text := '5';
Setfocus	สำหรับกำหนดให้ไปยังคอมโพเนนท์นั้น เช่น Edit1.Setfocus;



### Memo : เมโม

คอมโพเนนท์ที่มีลักษณะเหมือนกับ Edit แต่สามารถทำการใส่ข้อความได้มากกว่า 1 บรรทัด ให้สำหรับแสดงข้อความที่มีความจุจำนวนมาก

### คุณสมบัติที่สำคัญ (Properties)

Lines	สำหรับป้อนข้อความที่ต้องการใน Memo โดยมี String List BoEditor เป็นเครื่องมือช่วยแก้ไข
Scrollbars	สำหรับใช้กำหนดว่าจะให้มี Scrollbar ภายใน Memo หรือไม่ ถ้ามีให้อยู่ในรูปแบบใด ซึ่งมีให้เลือก 4 รูปแบบดังนี้

- SsNone ไม่มี Scrollbar
- SsHorizontal มีเพียง Scrollbar ในแนวนอน
- SsVertical มีเพียง Scrollbar ในแนวตั้ง
- SsBoth มี Scrollbar ทั้งแนวตั้งและแนวนอน

Text สำหรับเก็บข้อความที่จะใช้แสดงใน Memo ซึ่งคุณสมบัตินี้จะไม่พบในออปเจ็คอินสเป็คเตอร์แต่ละพบในขณะรันโปรแกรม

*Font, Visible, Hints, Showhint (เช่นเดียวกับ Label)*



Button : ปุ่ม

ใช้กำหนดปุ่มกดสำหรับผู้ใช้คลิกเพื่อสั่งในการทำงาน โดยปกติเมื่อดับเบิลคลิกที่

Button ในออกแบบระบบ Delphi จะสร้างโค้ด Events Onclick ให้โดยอัตโนมัติ

#### คุณสมบัติที่สำคัญ (Properties)

Caption สำหรับเก็บข้อความที่อยู่ใน Button

*Font, Visible, Hints, Showhint (เช่นเดียวกับ Label)*

#### เหตุการณ์ที่สำคัญ (Events)

OnClick กำหนดให้ดำเนินการใด ๆ เมื่อคลิกปุ่ม Button

#### พฤติกรรมที่สำคัญ (Methods)

Setfocus สำหรับกำหนดให้ไปยังคอมโพเนนท์นั้น เช่น Button1.Setfocus;



CheckBox : ปุ่มตรวจสอบ

คอมโพเนนท์ที่ใช้สำหรับสร้างระบบติดต่อผู้ใช้ โดยการให้ผู้ใช้สามารถเลือกตรวจสอบค่าด้วยการคลิกให้ปรากฏเครื่องหมาย ถูก หรือ ผิด (จะไม่ปรากฏเครื่องหมายใด ๆ )

โดยสามารถเลือกวางคอมโพเนนท์ได้หลายตัว

#### คุณสมบัติที่สำคัญ (Properties)

Caption สำหรับเก็บข้อความที่อยู่ใน Check Box

Checked สำหรับกำหนดค่าเริ่มต้นของ Check Box ว่าจะเป็น True หรือ False และในขณะรัน ก็สามารถใช้คุณสมบัตินี้อ่านค่า Check Box ได้ว่าอยู่ในสถานะใด

State สำหรับตรวจสอบว่า ขณะนี้ Check Box มีสถานะอะไร ดังนี้

- cbUnchecked Check Box นั้นไม่ถูกเลือก
- cbChecked Check Box นั้นถูกเลือก



- cbGrayed Check Box นั้นถูกเลือก แต่มีสถานะเป็น Grayed (คือมีบางส่วนใช่และไม่ใช่)

*Font, Visible, Hints, Showhint (เช่นเดียวกับ Label)*

### เหตุการณ์ที่สำคัญ (Events)

OnClick กำหนดให้ดำเนินการใด ๆ เมื่อคลิกปุ่ม Check Box หรือมีการเปลี่ยนแปลงค่า Checked ของคอมโพเนนต์ Check Box ในขณะรันโปรแกรม

### พฤติกรรมที่สำคัญ (Methods)

Setfocus สำหรับกำหนดให้ไปยังคอมโพเนนต์นั้น  
เช่น CheckBox1.Setfocus;



### RadioButton : ปุ่มเรดิโอ

คอมโพเนนต์ที่ใช้สำหรับสร้างระบบติดต่อผู้ใช้ โดยการให้ผู้ใช้สามารถเลือกตรวจสอบค่าด้วยการคลิกให้ปรากฏจุดดำ โดยมีสถานะเช่นเดียวกับ Check Box แต่การใช้งานจะใช้ตัวเดียวๆ ไม่ได้ จะต้องมีย่างน้อย 2 ตัวขึ้นไป เพราะเมื่อเลือกตัวใดตัวหนึ่ง จะทำให้อีกตัวที่เคยถูกเลือกเปลี่ยนค่าเป็นไม่ถูกเลือก

### คุณสมบัติที่สำคัญ (Properties)

Caption สำหรับเก็บข้อความที่อยู่ใน RadioButton  
Checked มีคุณสมบัติเหมือน CheckBox ซึ่งมีค่าเพียง True สำหรับ RadioButton ตัวที่ถูกเลือก ส่วนตัวอื่น ๆ จะเป็นค่าเป็น False

### เหตุการณ์ที่สำคัญ (Events)

OnClick กำหนดให้ดำเนินการใด ๆ เมื่อคลิกที่ RadioButton หรือเมื่อมีการเปลี่ยนค่าของ RadioButton



### Listbox : ลิสต์บ็อก

คอมโพเนนต์ที่ใช้สำหรับแสดงข้อมูลที่หลาย ๆ บรรทัด โดยที่ผู้ใช้สามารถเลือกข้อมูลนั้น ๆ ได้ ถ้ามีการบรรจุข้อมูลมากกว่าขนาดความสูงของ List Box จะปรากฏ

Scrollbar แนวตั้งขึ้นมาโดยอัตโนมัติ มีคุณสมบัติให้แสดงได้หลายคอลัมน์ รวมทั้งเลือกที่ละหลายข้อความได้

### คุณสมบัติที่สำคัญ (Properties)

BorderStyle	<p>สำหรับกำหนดค่ากรอบของ List Box ว่าควรมีลักษณะอย่างไร</p> <ul style="list-style-type: none"> <li>- bsNone ไม่มีกรอบ</li> <li>- bsSingle มีกรอบเส้นเดียว</li> </ul>
Columns	<p>สำหรับกำหนดค่าจำนวนแถวที่ใช้แสดงใน List Box สามารถกำหนดได้ตั้งแต่ 0 ถึง 255 (ตัวเลข) ซึ่งจะมีค่าเริ่มต้นเป็น 0 หมายถึงมีคอลัมน์เดียว</p>
Items	<p>สำหรับเก็บข้อความที่ใช้แสดงใน List Box โดยปกติ สามารถใช้ String List Editor เป็นเครื่องมือช่วยเช่นเดียวกับคอมพิวเตอร์ Memo แต่ยังสามารถใช้การกำหนดค่าคุณสมบัติขณะรันได้ด้วยคำสั่งดังรูปแบบ</p> <pre>[ชื่อ List Box].Items.Add('ข้อความที่ต้องการเพิ่ม');</pre> <p>ตัวอย่าง</p> <pre>ListBox1.Items.Add('มหาวิทยาลัยราชภัฏลำปาง');</pre>
ItemIndex	<p>สำหรับใช้เก็บค่าข้อความที่ถูกเลือกว่าอยู่อันดับที่เท่าไรของ List Box ซึ่งคุณสมบัตินี้จะพบในขณะรันโปรแกรมเท่านั้น</p>
Sorted	<p>สำหรับกำหนดให้เรียงลำดับตามพยางค์ขณะเมื่อกำหนดค่าเป็น True</p>
Style	<p>สำหรับกำหนดความสูงของแต่ละบรรทัดว่าจะให้เป็นแบบมาตรฐานหรือกำหนดขึ้นเอง ดังนี้</p> <ul style="list-style-type: none"> <li>- lbStandard ทุกบรรทัดจะมีค่าที่เท่ากับเพียงค่าเดียว</li> <li>- lbOwnerDrawFixed ค่าความสูงของทุกบรรทัดจะมีค่าเดียว และเท่ากับค่าที่กำหนดในคุณสมบัติ ItemHeight</li> </ul>

- `IbOwnerDrawVariable` ค่าความสูงของแต่ละบรรทัดจะไม่เท่ากัน โดยสามารถกำหนดผ่านเหตุการณ์ `OnMeasureItem` ของ `List Box`
- `ItemHeight` สำหรับกำหนดความสูงของบรรทัด จะทำงานก็ต่อเมื่อกำหนดให้คุณสมบัติของ `Style` เป็น `IbOwnerDrawFixed`

### พฤติกรรมที่สำคัญ (Methods)

- `Clear` สำหรับลบข้อมูลใน `List Box` ทั้งหมด เช่น `ListBox1.Clear;`



### **Combobox : คอมโบบ็อกซ์**

คอมโพเนนต์ที่มีลักษณะคล้ายกับ `Edit` แต่จะสามารถเรียกรายการออกมาเลือกได้ เมื่อเลือกเสร็จแล้วข้อความที่อยู่ในรายการเลือกจะถูกนำมาใส่ที่ `Edit Box` ดังนั้นจึงเป็นคอมโพเนนต์ที่รวมข้อดีจากคอมโพเนนต์ `List Box` และ `Edit` เข้าด้วยกัน เนื่องจากสามารถแสดงข้อความได้หลายรายการ แต่ประหยัดพื้นที่แสดงเท่ากับ `Edit`

### คุณสมบัติที่สำคัญ (Properties)

- `DropDownCount` สำหรับแสดงค่าจำนวนสูงสุดที่ได้แสดงรายการใน `Combobox` ซึ่งปกติจะกำหนดค่าเท่ากับ 8 แต่ถ้ามีรายการเลือกมากกว่าค่าที่กำหนดในคุณสมบัตินี้ จะปรากฏ `Scrollbar` ให้ แต่ถ้ามีจำนวนน้อยกว่าหรือเท่ากับจะแสดงตามปริมาณดังกล่าว
- `Items` สำหรับเก็บข้อความที่แสดงใน `Combobox`
- `Style` สำหรับกำหนดรูปแบบที่จะใช้แสดงใน `Combobox`
  - `csDropDown` ประกอบด้วย `List` ที่มีขนาดเท่ากับทุกบรรทัด และ `Edit` สามารถพิมพ์เองได้
  - `csSimple` จะมี `Edit` และ `List` ที่มีความสูงเท่ากับขึ้นอยู่กับความสูงของ `Combobox`
  - `csDropDownList` จะมีลักษณะ
  - `csOwnerDrawFixed`

- csOwnerDrawVariable

*Font, Visible, Hints, Showhint* (เช่นเดียวกับ Label)

### เหตุการณ์ที่สำคัญ (Events)

OnClick กำหนดให้ดำเนินการใด ๆ เมื่อคลิกปุ่ม Button

### พฤติกรรมที่สำคัญ (Methods)

Setfocus สำหรับกำหนดให้ไปยังคอมโพเนนต์นั้น เช่น Button1.Setfocus;



**GroupBox** : กรู๊ปบ็อกซ์

ใช้สำหรับจัดกลุ่มคอมโพเนนต์ โดยให้คอมโพเนนต์ที่ทำงานสัมพันธ์กันอยู่ในกลุ่มเดียวกันในขอบเขตที่กำหนดขึ้น

### คุณสมบัติที่สำคัญ (Properties)

Caption สำหรับกำหนดข้อความที่อยู่ด้านบนซ้ายของ GroupBox เพื่อแสดงชื่อของกรู๊ป



**RadioGroup** : เรดิโอกรู๊ป

เป็นคอมโพเนนต์ที่รวม RadioButton ไว้ในกรู๊ปบ็อกซ์ ซึ่งช่วยให้ใช้งานได้ง่ายและสะดวกกว่าการนำ RadioButton หลาย ๆ ตัววางลงบน GroupBox เอง เนื่องจากการจัดตำแหน่งจะเป็นไปโดยอัตโนมัติและไม่ต้องเขียนโปรแกรมการดำเนินการกับ RadioButton แต่ละตัว แต่ใช้วิธีการตรวจสอบลำดับภายในของ RadioGroup แทน

### คุณสมบัติที่สำคัญ (Properties)

Caption สำหรับกำหนดข้อความซึ่งแสดงอยู่ด้านบนซ้ายของ RadioGroup

Columns สำหรับกำหนดจำนวนคอลัมน์ที่จะแสดงใน RadioGroup

Items สำหรับกำหนดข้อความของ RadioButton แต่ละตัวที่แสดงอยู่ใน RadioGroup มีวิธีใช้เหมือนกับ ListBox

ItemIndex สำหรับแสดง/ระบุค่าที่แสดงว่า RadioButton ใดที่ถูกเลือก ซึ่งมีค่าเริ่มต้น 0 แต่กรณีที่ไม่เลือก RadioButton ใด ๆ ค่า ItemIndex จะมีค่าเป็น -1

**เหตุการณ์ที่สำคัญ (Events)**

OnClick      กำหนดให้ดำเนินการใด ๆ เมื่อคลิก RadioButton    ใด ๆ ใน  
RadioGroup

**พฤติกรรมที่สำคัญ (Methods)**

Setfocus      สำหรับกำหนดให้ไปยังคอมโพเนนต์นั้น เช่น Button1.Setfocus;



**Panel** : กรอบงานหรือแผ่นกระดาษ

ใช้กำหนดสร้างแผ่นกระดาษสำหรับใส่คอมโพเนนต์อื่น ๆ รวมทั้ง Panel เองก็ได้  
เพื่อให้เกิดขอบเขตของงานขึ้น

**คุณสมบัติที่สำคัญ (Properties)**

Caption      สำหรับเก็บข้อความที่อยู่ใน Panel

BevelInver      สำหรับกำหนดลักษณะขอบด้านในของ Panel มี 3 แบบ ดังนี้

- ไม่แสดงขอบด้านใน (bvNone)
- ขอบด้านในยุบต่ำเข้าไป (bvLowered)
- ขอบด้านในยกสูงขึ้นมา (bvRaised)

BevelOuter      สำหรับกำหนดลักษณะขอบด้านนอกของ Panel มี 3 แบบ ดังนี้

- ไม่แสดงขอบด้านนอก (bvNone)
- ขอบด้านนอกยุบต่ำเข้าไป (bvLowered)
- ขอบด้านนอกยกสูงขึ้นมา (bvRaised)

BevelWidth      สำหรับกำหนดความหนาของขอบด้านในและด้านนอก

(Default = 1)

BorderWidth      สำหรับกำหนดระยะห่างระหว่างขอบด้านในและขอบด้านนอก

(Default = 0)

Alignment, Font, Color, Visible, Hints, Showhint (เช่นเดียวกับ Label)

### ทดสอบ 4.1 จงสร้างแบบฟอร์มสำหรับการติดต่อกับผู้ใช้เพื่อสร้างโปรแกรมตาม



วัตถุประสงค์ดังรูปต่อไปนี้

ขั้นตอนการพัฒนา

1. สร้างปุ่ม “บันทึกรายละเอียด”, “แสดงรายละเอียด”, “ทดสอบการแสดงผล”
2. สร้างพื้นที่งานด้วย Panel จำนวน 3 Panel ตามลำดับ Page 1-3 ที่ปรากฏบนรูป
3. สร้างพื้นที่พร้อมแสดงชื่อผู้พัฒนาโปรแกรมด้านล่าง
4. กำหนดการทำงานดังนี้
  - a. หากประมวลผลโปรแกรมให้พื้นที่งานทั้ง 3 ไม่ปรากฏ
  - b. หากคลิกปุ่ม “บันทึกรายละเอียด” ให้ปรากฏพื้นที่ Page 1 พร้อมเคลียร์ข้อมูลใน Edit 1-3 ให้อว่าง และเมื่อคลิกปุ่ม “เก็บ” ในกลุ่มประเภทนักศึกษาให้เก็บข้อมูลใน Edit1 ลงใน Combobox1 และเมื่อคลิกปุ่ม “เก็บ” ในกลุ่มระดับนักศึกษาให้เก็บข้อมูลใน Edit2 ลงใน Combobox2 และเมื่อคลิกปุ่ม “เก็บ” ในกลุ่มระดับนักศึกษาให้เก็บข้อมูลใน Edit3 ลงใน Combobox3
  - c. หากคลิกปุ่ม “แสดงรายละเอียด” ให้ปรากฏพื้นที่ Page 2 เพื่อแสดงข้อมูลใน Combobox1-3

d. หากคลิกปุ่ม “ทดสอบการแสดงผล” ให้ปรากฏพื้นที่ Page 3 เพื่อให้  
 รอรับรหัสนักศึกษาทันที เมื่อคลิกปุ่ม “แสดงข้อมูล” ให้แสดงข้อมูล  
 ออกเป็นข้อมูลโดยเปรียบเทียบข้อมูลที่เคยจัดเก็บใน Combobox ที่  
 เตรียมไว้ทางจอภาพตามลำดับ Label 9-13

หมายเหตุ : หากไม่พบข้อมูลใน Combobox ที่เตรียมไว้ให้ขึ้น  
 สัญลักษณ์ - แทน

**ทดสอบ 4.2** จงสร้างแบบฟอร์มสำหรับสำรวจการจัดงานนิทรรศการราชภัฏ IT Fair ของ  
 มหาวิทยาลัยราชภัฏลำปาง ดังรูปต่อไปนี้



**ตัวอย่างการใช้งาน Component 2**

**แบบสำรวจสาขาวิชาวิศวกรรมซอฟต์แวร์  
 คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏลำปาง  
 งานนิทรรศการราชภัฏ IT Fair**

ข้อมูลทั่วไป

เพศ :  อาชีพ :  ข้าราชการ/รัฐวิสาหกิจ  0-5,000 บาท  
 ธุรกิจส่วนตัว  5,001-10,000 บาท  
 รับจ้าง/เกษตรกร  10,000 - 30,000 บาท  
 แม่บ้าน/พอบ้าน  30,000 บาทขึ้นไป

สถานภาพสมรส :  รายได้ :

ความสนใจ  Notebook  คอมพิวเตอร์  ชิ้นส่วน/อุปกรณ์อื่น ๆ

รายการ	5	4	3	2	1
1. สถานที่การจัดงานนิทรรศการราชภัฏ IT Fair มีความเหมาะสม	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2. กิจกรรมภายในงานนิทรรศการราชภัฏ IT Fair มีความเหมาะสม	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3. สินค้าภายในงานนิทรรศการทันสมัยและมีราคาย่อมเยา	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4. ระยะเวลาที่ใช้จัดงานนิทรรศการมีความเหมาะสม	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5. ท่านได้รับความรู้ด้านเทคโนโลยีภายในงานนิทรรศการ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6. ท่านได้รับความสะดวก และมีแหล่งบริการด้านสาธารณูปโภคเพียงพอ	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

ข้อเสนอแนะ/ความคิดเห็น :

ขั้นตอนการพัฒนา

1. กำหนดการตัวเลือกแบบต่าง ๆ โดยใช้คอมพิวเตอร์ในกลุ่ม Standard นำมาออกแบบฟอร์มดังรูปข้างต้น
2. กำหนดตัวแปรแถวลำดับ 2 มิติชื่อ ITFair ชนิดตัวเลข
3. โดยให้บันทึกเฉพาะระดับคะแนนแต่ละรายการสำรวจ 6 ข้อของแบบสำรวจ 10 ใบ เมื่อคลิกปุ่ม 'บันทึก' เพื่อนำมาคำนวณค่าเฉลี่ยและ

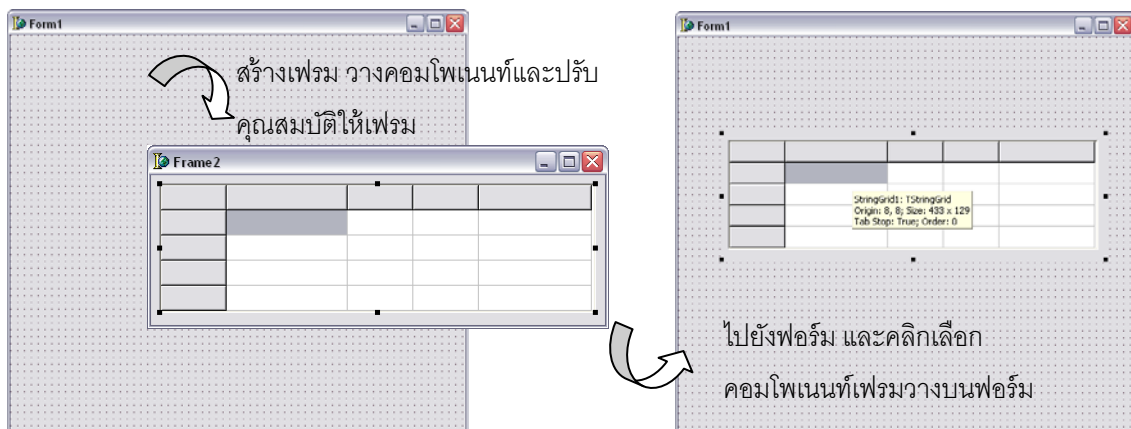
แสดงทางจอภาพเมื่อคลิกปุ่ม 'ประมวลผล' ในรูปแบบไดอะล็อก  
ข้อความ และเมื่อคลิกปุ่ม 'จบงาน' ให้ปิดโปรแกรม

### Frame : เฟรม

ใช้กำหนดสร้างลักษณะและกลไกการทำงานเหมือนฟอร์ม โดยสามารถใส่คอมโพเนนต์ต่าง ๆ ลงไปได้ และยังสามารถใส่ไว้ในฟอร์มหรือเฟรมอื่น ๆ ได้

สำหรับการเปลี่ยนแปลงใด ๆ ของเฟรมที่สร้างขึ้น จะมีผลต่อคอมโพเนนต์เฟรม ที่อยู่บนฟอร์มด้วย เช่น การเพิ่มปุ่มบนเฟรม จะมีผลให้ปรากฏบนฟอร์มที่เฟรมนั้นปรากฏอยู่ แต่การเปลี่ยนแปลงบนฟอร์มจะไม่มีผลต่อเฟรม วิธีการสร้างเฟรมสามารถจัดการได้ ดังนี้

1. เปิดโปรเจ็คด้วย File -> New -> Application
2. เปิดเฟรมด้วย File -> New -> Frame
3. วางคอมโพเนนต์และกำหนดค่าคุณสมบัติและเหตุการณ์
4. บันทึกเฟรม File -> Save
5. เลือกลงไปที่ฟอร์มที่ต้องการวางเฟรม
6. คลิกที่คอมโพเนนต์ Frame จะปรากฏไดอะล็อกบ็อกซ์ให้เลือกเฟรมโดยปรากฏรายชื่อของเฟรมที่สร้างขึ้น เลือกและคลิกปุ่ม OK
7. ปรับขนาดหน้าจอลงและวางคอมโพเนนต์เฟรมให้เหมาะสม



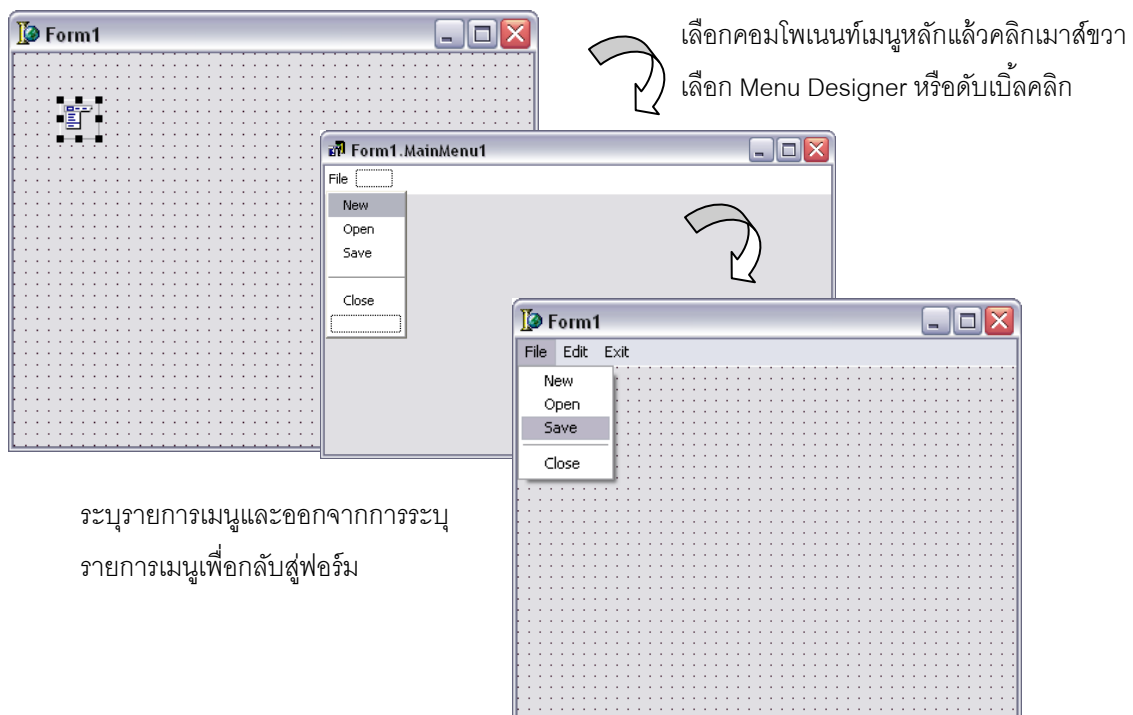




### MainMenu : เมนูหลัก

ใช้สำหรับสร้างเมนูหลัก เช่นเมนูของเดสทอป ซึ่งแสดงอยู่บนวินโดว์หลัก โดยภายในเมนูจะประกอบด้วยคำสั่งที่ใช้สำหรับการทำงานของแอปพลิเคชัน วิธีการสร้างเมนูภายหลังจากการเปิดโปรเจ็คแล้ว จัดการตามขั้นตอนได้ดังนี้

1. คลิกเลือกคอมโพเนนท์ MainMenu วางบนฟอร์ม
2. ดับเบิลคลิกที่คอมโพเนนท์ MainMenu เพื่อระบุรายการงานเมนูในคุณสมบัติ Caption และกดคีย์ Enter เพื่อระบุรายการงานเมนูต่อไป หากต้องการสร้างเส้นแบ่งเมนู ทำได้ด้วยการระบุเครื่องหมาย — ที่คุณสมบัติ Caption



### คุณสมบัติที่สำคัญ (Properties)

- |          |  |
|----------|--|
| Caption  | สำหรับกำหนดรายการงานของเมนูหลัก  |
| ShortCut | สำหรับกำหนดคีย์ลัดให้กับรายการเมนูด้วยวิธีการเลือกรายการในเมนูที่เตรียมไว้ |

Checked	สำหรับกำหนดเครื่องหมายถูกหน้ารายการเมนูหรือไม่ โดยกำหนดค่าเป็น True หรือ False
Enabled	สำหรับกำหนดว่าสามารถทำงานกับเมนูรายการนั้นได้หรือไม่ กรณีกำหนดเป็น False จะไม่สามารถทำงานได้ เมื่อแอปพลิเคชันทำงานจะไม่อนุญาตให้เลือกกรายงานเมนูนั้น

### เหตุการณ์ที่สำคัญ (Events)

OnClick	สำหรับกำหนดรายการงานของเมนูหลัก
---------	---------------------------------



### PopupMenu: ป๊อปอัพเมนู

ใช้สำหรับสร้างเมนูลัด ซึ่งมีคำสั่งและการทำงานของเหมือนเมนูหลัก โดยสามารถกำหนดที่ป๊อปอัพเมนูให้กับคอมโพเนนต์ใด ๆ แล้วเมื่อคลิกเมาส์ขวามือบนคอมโพเนนต์นั้นจะปรากฏเมนูขึ้นมาให้เลือกคำสั่งตามที่กำหนดในป๊อปอัพเมนู

เลือกคอมโพเนนต์ที่ต้องการแสดงเมนูลัด และวางคอมโพเนนต์ที่ป๊อปอัพเมนู จากนั้นดับเบิลคลิกเพื่อใส่รายการเมนู

ใส่รายการเมนูลัด และกำหนดเหตุการณ์ในคอมโพเนนต์ชื่อว่า Popup

**คุณสมบัติที่สำคัญ (Properties)**

Alignment	สำหรับกำหนดให้แสดงฟ็อปอัพเมนู ณ ตำแหน่งใดเทียบกับคลิกเมาส์ลงไป โดยสามารถเลือกกำหนดได้ 3 ค่าดังนี้ <ul style="list-style-type: none"> <li>- paLeft กำหนดให้แสดงที่มุมบนซ้าย ณ ตำแหน่งที่คลิกเมาส์อยู่</li> <li>- paCenter กำหนดให้แสดงที่จุดกึ่งกลางของขอบด้านบนตรงตำแหน่งที่คลิกเมาส์</li> <li>- paRight กำหนดให้แสดงที่มุมบนขวา ณ ตำแหน่งที่คลิกเมาส์อยู่</li> </ul>
AutoPopup	สำหรับกำหนดให้แสดงฟ็อปอัพเมนูทันทีที่คลิกเมาส์ขวา กรณีที่กำหนดค่าคุณสมบัตินี้เป็น False แล้วผู้ใช้ต้องเรียกพฤติกรรม popup เองเมื่อต้องการแสดงเมนู
Items	สำหรับกำหนดค่ารายการหรือคำสั่งต่าง ๆ ในเมนู
Images	สำหรับใช้กำหนดรูปภาพประกอบรายการหรือคำสั่งในเมนู

**เหตุการณ์ที่สำคัญ (Events)**

OnPopup	สำหรับกำหนดเหตุการณ์ที่เกิดขึ้นก่อนที่จะแสดงฟ็อปอัพเมนู ซึ่งปกติแล้วจะใช้ในการกำหนดค่า Checked, Enabled และ Visible ให้กับแต่ละคำสั่งในฟ็อปอัพเมนู
---------	--

**พฤติกรรมที่สำคัญ (Methods)**

Popup	สำหรับแสดงฟ็อปอัพเมนู ซึ่งสามารถกำหนดตำแหน่ง (หน่วยพิกเซล) ของฟ็อปอัพเมนูได้ โดยมีรูปแบบดังนี้ <u>รูปแบบ</u> [ชื่อฟ็อปอัพเมนู].Popup(ตำแหน่งในแนวนอน,ตำแหน่งในแนวตั้ง) <u>ตัวอย่าง</u> Popup1.Popup(50,100);
-------	---



### ทดสอบ 4.3

จงสร้างโปรแกรมเพื่อคำนวณหาพื้นที่รูปทรงเรขาคณิต โดยกำหนดให้สร้างฟอร์มหลักสำหรับเรียกใช้คำนวณหาพื้นที่รูปทรงต่าง ๆ จากฟอร์มย่อย ดังนั้นจึงต้องกำหนดฟอร์มเท่ากับจำนวนพื้นที่ของรูปทรงเรขาคณิตดังนี้

- 1) พื้นที่สี่เหลี่ยมผืนผ้า จากสูตร กว้าง x ยาว x สูง
- 2) พื้นที่สามเหลี่ยม จากสูตร  $\frac{1}{2} \times$  ฐาน  $\times$  สูง
- 3) พื้นที่วงกลม

#### ■ Additional



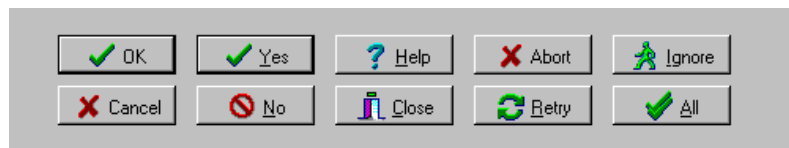
#### BitBtn : ปุ่ม

ใช้กำหนดปุ่มกดสำหรับผู้ใช้คลิกเหมือนกับ Button แต่มีส่วนเพิ่มเติมคือ สามารถแสดงภาพบนปุ่มได้

#### คุณสมบัติที่สำคัญ (Properties)

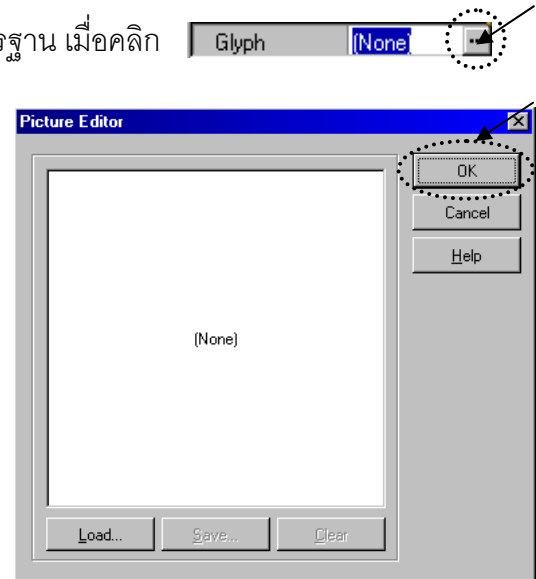
Caption สำหรับเก็บข้อความที่อยู่ใน BitBtn

Kind สำหรับกำหนดชนิดของปุ่ม เพื่อกำหนดข้อความ ภาพ และคุณสมบัติอื่น ๆ ตามมาตรฐานที่พบบ่อย สามารถเลือกชนิดของปุ่มได้ดังนี้



bkOK, bkCancel, bkYes, bkNo, bkHelp, bkClose, bkAbort, bkRetry, bkIgnore, bkAll และ bkCustom ปุ่มที่กำหนดเองดังคุณสมบัติข้างล่าง

Glyph สำหรับกำหนดรูปภาพที่แสดงบนปุ่ม สามารถเลือกรูปภาพจาก แหล่งรูปภาพมาตรฐาน เมื่อคลิก



(แหล่งรูปภาพมาตรฐาน)

C:\Program Files\Common Files\BorlandShared\Images\Buttons\\*.bmp

Layout สำหรับกำหนดตำแหน่งของรูปภาพที่ปรากฏบนปุ่ม ซึ่งมี 4 แบบ

- รูปภาพปรากฏอยู่ด้านซ้ายของข้อความ (blGlyphLeft)
- รูปภาพปรากฏอยู่ด้านขวาของข้อความ (blGlyphRight)
- รูปภาพปรากฏอยู่เหนือข้อความ (blGlyphTop)
- รูปภาพปรากฏอยู่ใต้ข้อความ (blGlyphBottom)


Font, Visible, Hints, Showhint (เช่นเดียวกับ Label)

#### เหตุการณ์ที่สำคัญ (Events)

OnClick กำหนดให้ดำเนินการใด ๆ เมื่อคลิกปุ่ม BitBtn1

#### พฤติกรรมที่สำคัญ (Methods)

Setfocus สำหรับกำหนดให้ไปยังคอมโพเนนต์นั้น เช่น BitBtn1.Setfocus;

 SpeedButton : ปุ่ม

ใช้กำหนดปุ่มกดที่มักใช้ในสปีดบาร์ โดยแสดงภาพบนปุ่มได้แต่ไม่สามารถใช้คีย์บอร์ดสั่งการได้ และไม่รับพฤติกรรม Focus

### คุณสมบัติที่สำคัญ (Properties)

Glyph	สำหรับกำหนดรูปภาพที่ปรากฏบนปุ่ม เช่นเดียวกับ BitBtn
GroupIndex	สำหรับกำหนดกลุ่มของ SpeedButton โดยกำหนดค่า GroupIndex เดียวกันที่ไม่เป็นศูนย์ให้กับ SpeedButton ที่อยู่ในกลุ่มเดียวกันเพื่อให้สามารถเลือกปุ่มใดปุ่มหนึ่งค้างได้เพียงปุ่มเดียว
Down	สำหรับกำหนดให้เป็นปุ่มยุบลงไป โดยกำหนดค่าเป็น True หรือ False และจะใช้ได้ก็ต่อเมื่อ GroupIndex ไม่เป็น 0 กรณีที่ถูกกำหนดให้เป็น True ปุ่มนี้จะยุบลงแสดงว่าถูกเลือก
Flat	สำหรับกำหนดให้เป็นปุ่มใส โดยกำหนดค่าเป็น True หรือ False โดยที่กำหนดเป็น True จะปรากฏรูปปุ่มเมื่อมีการใช้เมาส์เคลื่อนไปเหนือปุ่ม
AllowAllUp	สำหรับกำหนดให้ทุกปุ่มที่อยู่ในกลุ่มเดียวกันมีสถานะยกทั้งหมด (ไม่มีปุ่มใดยุบเลย) ได้หรือไม่ โดยกำหนดค่าเป็น True หรือ False

### เหตุการณ์ที่สำคัญ (Events)

OnClick	กำหนดให้ดำเนินการใด ๆ เมื่อคลิกปุ่ม SpeedButton
---------	---

### MaskEdit : ปุ่ม

ใช้กำหนดการรับข้อมูลเช่นเดียวกับ Edit แต่สามารถกำหนดรูปแบบของข้อมูลที่ต้องการรับได้เพื่อเป็นการบังคับลักษณะการป้อนข้อมูล เช่น กำหนดรูปแบบวันที่ เวลา หรือเบอร์โทรศัพท์ เป็นต้น วิธีใช้จะต้องใช้อักขระที่กำหนดรูปแบบให้กับคุณสมบัติ MaskEdit โดยประกอบด้วย 3 ส่วน แบ่งแยกด้วยเครื่องหมาย ; โดยส่วนแรกจะเป็นตัวอักขระกำหนดรูปแบบข้อมูล ซึ่งตัวอักขระที่ใช้จะมีความหมายดังนี้

อักขระ	ความหมาย
!	กำหนดให้การป้อนไม่ครบข้อมูล

>	กำหนดให้ตัวอักษรที่ตามหลังเครื่องหมายนี้เป็นพิมพ์ใหญ่
<	กำหนดให้ตัวอักษรที่ตามหลังเครื่องหมายนี้เป็นพิมพ์เล็ก
<>	กำหนดให้ตัวอักษรที่ตามหลังเครื่องหมายนี้เป็นตัวปกติตามที่ผู้ใช้ป้อน
/	กำหนดให้แสดงตัวอักษรที่ตามหลังเครื่องหมายนี้เป็นส่วนหนึ่งของ Mask เช่น ต้องการแสดงตัวอักษร < จะต้องใส่ EditMask เป็น \< เราจะได้อักษร < ซึ่งไม่ได้เป็นการกำหนดการทำงานสำหรับเครื่องหมาย<
L	ตำแหน่งนี้จะต้องใส่ตัวอักษรเป็น A-Z หรือ a-z
I	ตำแหน่งนี้จะใส่ตัวอักษรเป็น A-Z หรือ a-z โดยจะไม่ใช่ก็ได้
A	ตำแหน่งนี้จะต้องใส่ตัวอักษรเป็น A-Z หรือ a-z หรือ 0-9
อักขระ	ความหมาย
A	ตำแหน่งนี้จะใส่ตัวอักษรเป็น A-Z หรือ a-z หรือ 0-9 โดยจะไม่ใช่ก็ได้
C	ตำแหน่งนี้จะต้องใส่ตัวอักษรแต่เป็นตัวอะไรก็ได้
c	ตำแหน่งนี้จะใส่ตัวอักษรใด ๆ หรือไม่ใช่ก็ได้
O	ตำแหน่งนี้จะต้องใส่ตัวเลขเท่านั้น
G	ตำแหน่งนี้จะใส่ตัวเลขเท่านั้น โดยจะไม่ใช่ก็ได้
#	ตำแหน่งจะใส่ตัวเลขหรือเครื่องหมาย + หรือ - ก็ได้ โดยจะไม่ใช่ก็ได้
:	เป็นตัวอักษรที่ใช้แสดงตัวคั่น มักจะใช้กับข้อมูลประเภทเวลา
/	เป็นตัวอักษรที่ใช้แสดงตัวคั่น มักจะใช้กับวัน เดือน ปี
_	เป็นเครื่องหมายที่ใช้แสดงแทนช่องว่าง

**ตัวอย่าง :** การนำอักขระต่าง ๆ มาจัดรูปแบบ

!(999\)\000-0000;1;\_                      ตัวอย่างข้อมูล (081)180-8336

!99/99/0000;1;\_                              ตัวอย่างข้อมูล 01/01/2551

!90:00:00>LL;1;\_                            ตัวอย่างข้อมูล 16:30:30PM

### คุณสมบัติที่สำคัญ (Properties)

EditMak	สำหรับกำหนดรูปแบบของตัวอักษรที่เป็น Mask
Text	เป็นช่องสำหรับใส่ค่าเพื่อเริ่มต้นตามรูปแบบ

### เหตุการณ์ที่สำคัญ (Events)

*OnChange, OnEnter, OnExit OnKeyPress* เช่นเดียวกับ Edit

### พฤติกรรมที่สำคัญ (Methods)

*Clear, Text, Setfocus* เช่นเดียวกับ Edit



StringGrid : ตาราง

ใช้แสดงข้อมูลแบบตาราง ซึ่งข้อมูลจะแสดงอยู่ในแต่ละช่องของตารางตัวอย่างเช่น การแสดงปฏิทินไว้ในตาราง

### คุณสมบัติที่สำคัญ (Properties)

Cells เป็นคุณสมบัติที่ไม่ปรากฏในออบเจ็กต์อินสเปคเตอร์ แต่สามารถเรียกใช้ได้ขณะรันแอปพลิเคชัน โดยที่ข้อมูลในแต่ละช่องของ StringGrid มีลักษณะเป็นอะเรย์ด้วยรูปแบบของคอลัมน์และแถว ดังรูปแบบต่อไปนี้

[ชื่อ StringGrid].Cells[ตำแหน่งคอลัมน์, ตำแหน่งแถว]

ตัวอย่างการกำหนดคุณสมบัติ เช่น

```
StringGrid1.Cells[0,0] := 'ค่าที่ 1';
```

ColCount ใช้กำหนดจำนวนคอลัมน์ของ StringGrid โดยคอลัมน์แรกเริ่มต้นที่ 0

Col เป็นคุณสมบัติที่ไม่ปรากฏในออบเจ็กต์อินสเปคเตอร์ แต่สามารถเรียกใช้ได้ขณะรันแอปพลิเคชันเป็นค่าคอลัมน์ปัจจุบันที่กำลังทำงานอยู่

Cols เป็นคุณสมบัติที่เป็นรายการข้อมูลของแต่ละคอลัมน์ มีลักษณะเป็นอะเรย์ของ TStrings ที่เป็นข้อมูลของทุกแถวที่อยู่ในคอลัมน์



RowCount	ใช้กำหนดจำนวนแถวของ StringGrid โดยแถวแรกเริ่มต้นที่ 0
Row	เป็นคุณสมบัติที่ไม่ปรากฏในออบเจ็กต์อินสเปคเตอร์ แต่สามารถเรียกใช้ได้ในขณะรันแอปพลิเคชันเป็นค่าแถวปัจจุบันที่กำหนดทำงานอยู่
Rows	เป็นคุณสมบัติที่เก็บรายการข้อมูลแต่ละแถวมีลักษณะเป็นอะเรย์ของ TStrings ที่เป็นข้อมูลของทุก ๆ คอลัมน์ที่อยู่ในแถว



### Image : รูปภาพ

ใช้กำหนดรูปภาพทั้งในรูปแบบ บิตแมพ (ไฟล์ .Bmp) ไอคอน (ไฟล์ .ico) และเมตาไฟล์ (ไฟล์ .wmf)

#### คุณสมบัติที่สำคัญ (Properties)

Picture	สำหรับเรียก/แสดงรูปภาพที่ต้องการ โดยดำเนินการเช่นเดียวกับการใช้ Glyph ของ BitBtn
Autosize	สำหรับปรับขนาดของรูปภาพที่แสดง
Stretch	สำหรับให้แสดงภาพย่อ/ขยายเต็มพื้นที่ภายในคอมโพเนนท์ Image <ul style="list-style-type: none"> <li>- True จะได้ภาพที่มีขนาดเท่ากับขนาดคอมโพเนนท์</li> <li>- False จะได้ภาพที่มีขนาดตามจริงของภาพนั้น</li> </ul>



### Shape : รูปทรง

ใช้สำหรับวาดรูปทรงเรขาคณิตในแบบต่าง ๆ เช่น เส้นตรง สี่เหลี่ยม วงกลม เป็นต้น

#### คุณสมบัติที่สำคัญ (Properties)

Shape	สำหรับใช้กำหนดรูปทรงในแบบต่าง ๆ มีค่าให้เลือกคือ <ul style="list-style-type: none"> <li>- stCircle แสดงรูปวงกลม</li> <li>- stEllipse แสดงรูปวงรี</li> <li>- stRectangle แสดงรูปสี่เหลี่ยมผืนผ้าแบบมุมฉาก</li> <li>- stRoundRect แสดงรูปสี่เหลี่ยมผืนผ้าแบบมุมมน</li> </ul>
-------	--

	- stRoundSquare แสดงรูปสี่เหลี่ยมจัตุรัส แบบมุมมน
	- stSquare แสดงรูปสี่เหลี่ยมจัตุรัสแบบมุมฉาก
Brush	สำหรับใช้กำหนดสีและรูปแบบลวดลายของพื้นที่ภายใน Shape
Pen	สำหรับใช้กำหนดสีและรูปแบบชนิดของเส้นขอบของ Shape



### Bevel : กรอบ

ใช้สำหรับแสดงกรอบหรือเส้นตรงแบบที่มีเงาสามมิติ ซึ่งมีประโยชน์ในการตีกรอบหรือสร้างเส้นแบ่งบนฟอร์ม

#### คุณสมบัติที่สำคัญ (Properties)

Shape	สำหรับใช้กำหนดลักษณะของ Bevel ซึ่งมีค่าให้เลือกดังนี้
	- bsBox แสดงกรอบสี่เหลี่ยม โดยจะแสดงเป็นพื้นที่ยกขึ้นหรือยุบลงขึ้นอยู่กับค่าคุณสมบัติ Style
	- bsFrame แสดงกรอบสี่เหลี่ยม โดยแสดงเป็นเส้นกรอบสามมิติ
	- bsTopline แสดงเส้นขอบด้านบนของกรอบ
	- bsBottomLine แสดงเส้นขอบด้านล่างของกรอบ
	- bsLeftLine แสดงเส้นขอบด้านซ้ายของกรอบ
	- bsRightLine แสดงเส้นขอบด้านขวาของกรอบ
	- bsSpacer ไม่มีการแสดงเส้นใด ๆ ของกรอบเลย โดยจะแสดงเป็นพื้นที่ว่าง
Style	สำหรับใช้กำหนดรูปแบบกรอบของพื้นที่สี่เหลี่ยม โดยมีค่าให้เลือก คือ
	- bsLowered แสดงกรอบในลักษณะยุบลง
	- bsRaised แสดงกรอบในลักษณะยกขึ้น



### ScrollBar : สโครลบาร์

ใช้สำหรับสร้างพื้นที่ที่ต้องการสโครลบาร์ โดยสามารถวางคอมโพเนนต์ต่าง ๆ ลงไปใน ScrollBox ได้เหมือน Panel แต่สามารถมีสโครลบาร์สำหรับเคลื่อนคอมโพเนนต์ที่ตกขอบเข้ามาดูและทำงานได้

#### คุณสมบัติที่สำคัญ (Properties)

AutoScroll	สำหรับใช้กำหนดให้แสดงสโครลบาร์โดยอัตโนมัติ
HorzScrollBar	สำหรับใช้กำหนดให้แสดงสโครลลารีในแนวนอน
VertScrollBar	สำหรับใช้กำหนดให้แสดงสโครลลารีในแนวตั้ง



#### ทดสอบ 4.4

จงสร้างโปรแกรมแสดงรายการข้อมูลใน External File ชื่อ MyFriend.txt ที่ได้จากแบบทดลอง 3.14 ลงในคอมโพเนนต์ตาราง StringGrid โดยออกแบบการแสดงผลให้เหมาะสม



#### ทดสอบ 4.5

จงสร้างโปรแกรมเพื่อคำนวณอัตราภาษี กำหนดให้ออกแบบให้ใกล้เคียงกับรูปต้นฉบับที่แนบท้ายภาคผนวก และกำหนดรูปแบบการรับค่าด้วยคอมโพเนนต์ MaskEdit และใช้ปุ่ม BitBtn หรือ SpeedButton เท่านั้น

## Win32



### PageControl : เพจคอนโทรล

ใช้สำหรับแสดงพื้นที่เสมือนมีหน้าต่างซ้อนกัน ซึ่งพื้นที่ของแต่ละหน้าจะสามารถวางคอมโพเนนต์แตกต่างกัน มีประโยชน์หลักเพื่อลดพื้นที่ในการออกแบบแอปพลิเคชัน โดยวิธีการเพิ่มหน้าต่างใน PageControl คือการคลิกเมาส์ขวาและเลือก New Page

### คุณสมบัติที่สำคัญ (Properties)

ActivePage	สำหรับใช้แสดงพื้นที่หน้าที่ปัจจุบัน
Pages	สำหรับเก็บพื้นที่ทั้งหมด ถ้าค่าดัชนี (index) เป็น 0 แสดงว่าเป็นพื้นที่ที่ 1 และเรียงต่อไปเรื่อย ๆ โดยมีรูปแบบดังนี้ $[\text{ชื่อ PageControl}].\text{Pages}[\text{Index}]$ ตัวอย่างการกำหนดคุณสมบัติ เช่น $\text{PageControl1}.\text{Pages}[0] := \text{'พื้นที่ 1'};$
PageCount	สำหรับแสดงจำนวนพื้นที่ทั้งหมด



### **ImageList : รายการรูป**

ใช้สำหรับเก็บรายการรูปภาพต่าง ๆ เพื่อนำไปแสดงในคอมโพเนนต์อื่น ๆ เช่น ToolBar, CoolBar, TreeView , ListView เป็นต้น โดยจะใช้ ImageList Editor เป็นตัวจัดการเรียกใช้และจัดเก็บ โดยกำหนดค่าดัชนี (index) เริ่มต้นที่ 0 เพื่อง่ายต่อการเรียกใช้

### คุณสมบัติที่สำคัญ (Properties)

ImageList Editor	สำหรับกำหนดจัดการรูปภาพ โดยคลิกเมาส์ขวา และดำเนินการด้วยปุ่มงานดังนี้
- Add	ใช้เพิ่มรูปในรายการ
- Delete	ใช้ลบรูปในรายการตำแหน่งที่เลือก
- Clear	ใช้ลบรูปทั้งหมดในรายการ
- Export	ใช้บันทึกรูปที่เก็บใน ImageList ลงไฟล์



### **DateTimePicker : วันที่/เวลา**

ใช้สำหรับแสดงและรับค่าวันที่หรือเวลา

### คุณสมบัติที่สำคัญ (Properties)

Date	สำหรับกำหนดค่าวันที่/เวลาปัจจุบัน
DateFormat	สำหรับกำหนดรูปแบบของวันที่/เวลาที่มีให้เลือก แบบ คือ

	<ul style="list-style-type: none"> <li>- dfLong แสดงเต็มรูปแบบ เช่น 10 ตุลาคม 2550 (ขึ้นอยู่กับ การปรับวันที่ในระบบปฏิบัติการ)</li> <li>- dfShort แสดงแบบย่อ เช่น 10/10/2007</li> </ul>
DateMode	สำหรับเลือกโหมดปรับเปลี่ยนวันที่/เวลามีให้เลือก 2 แบบ คือ <ul style="list-style-type: none"> <li>- dmCombobox ปรับเปลี่ยนแบบให้เลือก</li> <li>- dmUpdown ปรับเปลี่ยนแบบเลือกขึ้นลง</li> </ul>
Kind	สำหรับกำหนดชนิดข้อมูลมีให้เลือก 2 แบบคือ <ul style="list-style-type: none"> <li>- dtkDate แสดงวันที่</li> <li>- dtkTime แสดงเวลา</li> </ul>

#### เหตุการณ์ที่สำคัญ (Events)

OnChange	กำหนดให้ดำเนินการ เมื่อวันที่/เวลาเกิดการเปลี่ยนแปลง
OnClick	กำหนดให้ดำเนินการเมื่อคลิกหรือกดคีย์ Enter



#### StatusBar :

ใช้สำหรับแสดงสถานะการทำงานของแอปพลิเคชัน โดยจะปรากฏด้านล่างของแอปพลิเคชัน

#### คุณสมบัติที่สำคัญ (Properties)

Caption	สำหรับกำหนดข้อความที่ต้องการแสดงบน Label
---------	--

#### เหตุการณ์ที่สำคัญ (Events)

OnChange	กำหนดให้ดำเนินการ เมื่อข้อความใน Edit เกิดการเปลี่ยนแปลง
----------	--

#### พฤติกรรมที่สำคัญ (Methods)

Clear	สำหรับลบข้อความที่อยู่ใน Label เช่น Label1.Clear;
Caption	สำหรับกำหนดค่าให้กับ Label (เฉพาะข้อความ) เช่น Label.Caption := '5';



### ToolBar : ทุลบาร์

ใช้สำหรับสร้างและเรียงปุ่มสปีดที่มีประสิทธิภาพ โดยสามารถจัดการขนาด ความสูง ระยะห่างของปุ่มต่าง ๆ ได้

### คุณสมบัติที่สำคัญ (Properties)

Images

ButtonCount

### เหตุการณ์ที่สำคัญ (Events)

OnChange กำหนดให้ดำเนินการ เมื่อข้อความใน Edit เกิดการเปลี่ยนแปลง

### พฤติกรรมที่สำคัญ (Methods)

Clear สำหรับลบข้อความที่อยู่ใน Label เช่น Label1.Clear;

Caption สำหรับกำหนดค่าให้กับ Label (เฉพาะข้อความ)

เช่น Label.Caption := '5';



### ทดสอบ 4.6

จงสร้างโปรแกรมเก็บประวัติบุคลากรของมหาวิทยาลัยราชภัฏ ลำปาง โดยกำหนดให้พื้นที่หลักเก็บข้อมูลทั่วไป ได้แก่ เลขที่ ตำแหน่ง, คำนำหน้า, ชื่อ, สกุล, ตำแหน่ง, ฝ่ายงานที่สังกัด, หน่วยงานที่สังกัด และแบ่งพื้นที่ย่อยในการเก็บประวัติอื่น ๆ ด้วยคอมโพเนนท์ PageControl คือ

- 1) ประวัติส่วนตัว ประกอบด้วย เลขที่บัตรประชาชน, เลขที่บัตรประกันสังคม, วันเดือนปีเกิด, เพศ, วันที่เข้าทำงาน, สถานภาพสมรส, ชื่อคู่สมรส, สถานภาพการมีบุตร, จำนวนบุตร, สถานภาพความพิการ
- 2) ประวัติการศึกษา ประกอบด้วย ระดับการศึกษาที่จบ, สาขาวิชาที่จบ, สถานศึกษาที่จบ, ประเทศที่จบ, ปีการศึกษาที่จบ และแสดงในรูปแบบตาราง

- 3) ประวัติการทำงาน ประกอบด้วย เลขที่คำสั่ง, วันเดือนปี, ข้อมูลความดีความชอบ, อัตราเงินเดือนที่ได้รับ และแสดงในรูปแบบตาราง

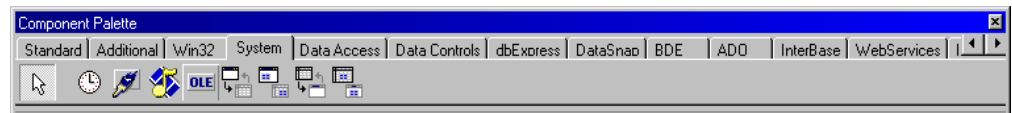


#### **ทดสอบ 4.7**

จงสร้างโปรแกรมนำเสนอสินค้า OTOP อย่างน้อย 2 ชนิดของจังหวัดหนึ่ง (นักศึกษาเลือกจังหวัดห้ามซ้ำกัน) โดยสร้างดังนี้

- 1) ฟอर्मหลักสำหรับการเรียกแสดงสินค้าแต่ละชนิดด้วยคอมโพเนนท์เมนูบาร์และเมนูทูลบาร์ และมีคอมโพเนนท์ StatusBar บอกสถานะของทุกฟอर्म โดยอธิบายจังหวัดและคำขวัญของจังหวัดนั้นด้วย
- 2) ฟอर्मย่อยสำหรับแสดงสินค้า OTOP ชนิดที่ 1
- 3) ฟอर्मย่อยสำหรับแสดงสินค้า OTOP ชนิดที่ 2

#### ■ System



Timer : เวลา

ใช้สำหรับวาดรูปโดยมีขอบเขตการวาดอยู่ภายใน PictureBox เท่านั้น ซึ่งอาจจะวางอยู่ในตำแหน่งใดก็ได้ และจุดอ้างอิงในการวาดใช้มุมบนด้านซ้ายของ PictureBox เป็นจุด 0,0 เสมอ (แทนค่า X คอสิมันน์ และ Y แกว)

#### **คุณสมบัติที่สำคัญ (Properties)**

Font, Hints, Showhint (เช่นเดียวกับ Label)

#### **เหตุการณ์ที่สำคัญ (Events)**

OnPaint กำหนดเมื่อต้องการให้มีการวาดหน้าจอ โดยจะใส่คำสั่งที่ใช้วาดรูปบน  
PictureBox เพื่อให้แสดงรูปตามที่ต้องการ

**พฤติกรรมที่สำคัญ (Methods)**

Canvas	สำหรับวาดรูปในรูปแบบที่ต้องการ
Repaint	สำหรับลบการแสดงผลทั้งหมดบน PictureBox

**PictureBox : การวาดรูป**

ใช้สำหรับวาดรูปโดยมีขอบเขตการวาดอยู่ภายใน PictureBox เท่านั้น ซึ่งอาจจะวางอยู่ในตำแหน่งใดก็ได้ และจุดอ้างอิงในการวาดใช้มุมบนด้านซ้ายของ PictureBox เป็นจุด 0,0 เสมอ (แทนค่า X คออดิเนต และ Y แกว)

**คุณสมบัติที่สำคัญ (Properties)**

Font, Hints, Showhint (เช่นเดียวกับ Label)

**เหตุการณ์ที่สำคัญ (Events)**

OnPaint กำหนดเมื่อต้องการให้มีการวาดหน้าจอ โดยจะใส่คำสั่งที่ใช้วาดรูปบน PictureBox เพื่อให้แสดงรูปตามที่ต้องการ

**พฤติกรรมที่สำคัญ (Methods)**

Canvas	สำหรับวาดรูปในรูปแบบที่ต้องการ
Repaint	สำหรับลบการแสดงผลทั้งหมดบน PictureBox

**พฤติกรรม (Method) สำหรับใช้คำสั่งวาดรูปบนพื้นที่การวาด (Canvas)**

- **MoveTo** สำหรับที่ใช้ในการย้ายตำแหน่ง (X,Y) เพื่อกำหนดจุดเริ่มต้นของการลากเส้น

รูปแบบ MoveTo( x , y );

ตัวอย่าง MoveTo(50, 50);

- **LineTo** สำหรับลากเส้นตรงจากตำแหน่งปัจจุบันไปยังตำแหน่ง (X,Y) และจะย้ายจุดปัจจุบันไปยังตำแหน่ง (X,Y) ด้วย

รูปแบบ LineTo( x , y );

ตัวอย่าง LineTo(100, 100);

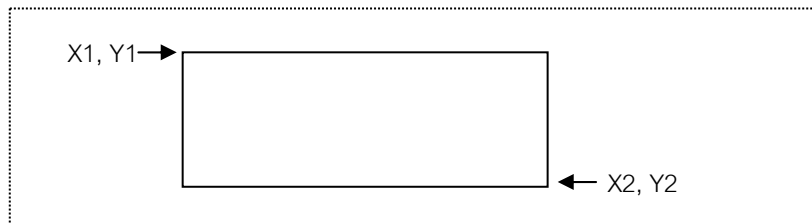




- **Rectangle** สำหรับวาดรูปสี่เหลี่ยม โดยกำหนดค่าตำแหน่ง X1, Y1 ของมุมสี่เหลี่ยมซ้ายสุด และค่าตำแหน่ง X2, Y2 ของมุมสี่เหลี่ยมขวาสุด

รูปแบบ            Rectangle( X1 ,Y1, X2, Y2 );

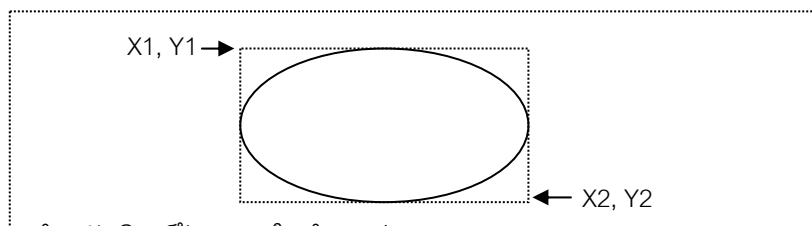
ตัวอย่าง            Rectangle(50, 50, 200,150);



- **Ellipse** สำหรับวาดรูปวงกลมหรือวงรี โดยกำหนดค่าตำแหน่ง X1, Y1 เป็นตำแหน่งมุมบนซ้าย และค่าตำแหน่ง X2, Y2 เป็นตำแหน่งมุมบนขวา

รูปแบบ            MoveTo( x , y );

ตัวอย่าง            MoveTo(50, 50);



- **TextOut** สำหรับพิมพ์ข้อความในตำแหน่ง X, Y

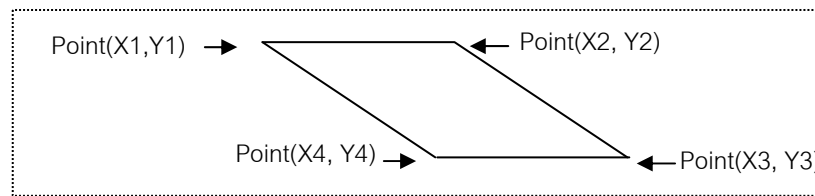
รูปแบบ            TextOut( X, Y, ข้อความ);

ตัวอย่าง            TextOut(50, 50, 'ดอกไม้ให้คุณ...');

- **Polygon** สำหรับวาดรูปปิดหลายเหลี่ยม โดยเชื่อมจุดต่าง ๆ พร้อมลากเส้นตรงต่อกัน

รูปแบบ Polygon([Point(X1,Y1), Point(X2, Y2), ..., Point(Xn..Yn)]);

ตัวอย่าง Polygon([Point(50, 50), Point(200, 50), Point(200, 100), Point(50, 100)]);



- **Brush** สำหรับระบายสีรูปทรง

รูปแบบ Brush.Color := รหัสสี ;

ตัวอย่าง Brush.Color := clYellow;

ตัวอย่างสี

clYellow	เหลือง	clBule	น้ำเงิน	clWhite	ขาว	clBlack	ดำ
clGreen	เขียว	clRed	แดง	clGray	เทา	clFuchsia	บานเย็น
clMaroon	เลือดหมู	clPurple	ม่วง	clSilver	เงิน	clCream	ครีม



**ทดสอบ 4.8**

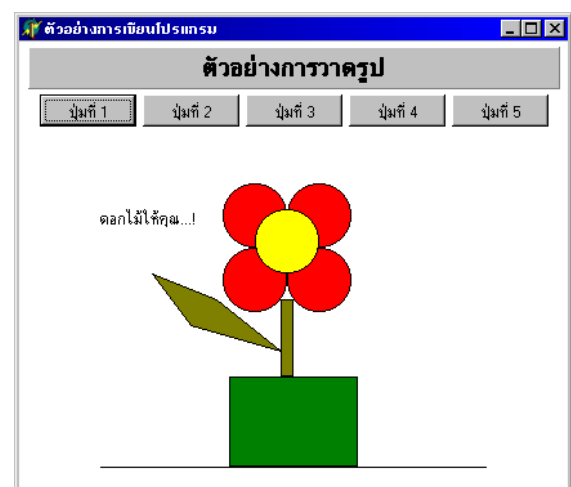
จงสร้างโปรแกรมโดยกำหนดพื้นที่แสดงรูปวาดขนาดความสูง 260 และความกว้าง 400 Point โดยเตรียมข้อความและปุ่มดำเนินการ 5 ปุ่ม ดังรูปที่ 4 เพื่อกำหนดเหตุการณ์ดังนี้

- หากคลิกปุ่มที่ 1 ให้ปรากฏเส้นตรงใน PaintBox
- หากคลิกปุ่มที่ 2 ให้ปรากฏสี่เหลี่ยมใน PaintBox
- หากคลิกปุ่มที่ 3 ให้ปรากฏวงกลมใน PaintBox
- หากคลิกปุ่มที่ 4 ให้ปรากฏข้อความใน PaintBox
- หากคลิกปุ่มที่ 5 ให้ลบรูปวาดทั้งหมดใน PaintBox

รูปที่ 4 ตัวอย่างการวาดรูป



รูปที่ 5 ตัวอย่างผลลัพธ์ของการวาดรูป



ตัวอย่างคำสั่ง

procedure TForm1.BitBtn1Click(Sender: TObject); ← สร้างเหตุการณ์เมื่อคลิกปุ่ม 1

begin

    PaintBox1.Canvas.MoveTo(50,250);

    PaintBox1.Canvas.LineTo(350,250);

end;

procedure TForm1.BitBtn2Click(Sender: TObject); ← สร้างเหตุการณ์เมื่อคลิกปุ่ม 2

begin

    PaintBox1.Canvas.Brush.Color := clgreen;

    PaintBox1.Canvas.Rectangle(150,180,250,250);

end;

```
procedure TForm1.BitBtn3Click(Sender: TObject); ← สร้างเหตุการณ์เมื่อคลิกปุ่ม 3
```

```
begin
```

```
    PaintBox1.Canvas.Brush.Color := clRed;
```

```
    PaintBox1.Canvas.Ellipse(145,30,195,80);
```

```
    PaintBox1.Canvas.Ellipse(195,30,245,80);
```

```
    PaintBox1.Canvas.Ellipse(145,80,195,130);
```

```
    PaintBox1.Canvas.Ellipse(195,80,245,130);
```

```
    PaintBox1.Canvas.Brush.Color := clYellow;
```

```
    PaintBox1.Canvas.Ellipse(170,50,220,100);
```

```
end;
```

```
procedure TForm1.BitBtn4Click(Sender: TObject); ← สร้างเหตุการณ์เมื่อคลิกปุ่ม 4
```

```
begin
```

```
    PaintBox1.Canvas.Brush.Color := clOlive;
```

```
    PaintBox1.Canvas.Rectangle(190,120,200,180);
```

```
    PaintBox1.Canvas.Polygon([Point(90,100),Point(140,120),Point(190,160),  
Point(120,140)]);
```

```
    PaintBox1.Canvas.Brush.Color := clWhite;
```

```
    PaintBox1.Canvas.TextOut(50,50,'ดอกไม้ให้คุณ...!');
```

```
end;
```

```
procedure TForm1.BitBtn5Click(Sender: TObject); ← สร้างเหตุการณ์เมื่อคลิกปุ่ม 5
```

```
begin
```

```
    PaintBox1.Repaint;
```

```
end;
```

## แบบฝึกหัดที่ 4

1. จงออกแบบและเขียนโปรแกรมชื่อ EX2\_1 เพื่อนำเสนอข้อมูลประวัติของตนเอง โดยต้องประกอบด้วยข้อมูลดังต่อไปนี้อย่างละเอียด
  - รหัสนักศึกษา
  - คติประจำใจ
  - ชื่อนักศึกษา
  - เป้าหมายในชีวิต
  - สกุลนักศึกษา
  - สิ่งที่บกพร่องของตนเอง
  - เพศ
  - ชื่อเพื่อนสนิท
  - วันเดือนปีเกิด
  - สาเหตุของการเข้าเรียนสาขาวิชา
  - ชื่อเล่น
  - วิศวกรรมซอฟต์แวร์ (Memo)
  - ที่อยู่ติดต่อได้
  - Gossip เพื่อน ๆ ในห้อง (Memo)
  - เบอร์โทรศัพท์
  - รูปบัตรแม่แบบ (รูปเดียว)
2. จงเขียนโปรแกรมชื่อ EX2\_2 โดยใช้แบบฟอร์มจากแบบทดสอบ 4.2 โดยออกแบบการประมวลผลของงานตามความต้องการดังนี้
  - ต้องการให้เก็บข้อมูลที่ได้จากการสำรวจทุกรายการที่ปรากฏบนฟอร์มใน Text File ชื่อ ITFair.txt
  - ต้องการให้มีปุ่ม 'แสดงผล' และเมื่อคลิกปุ่มดังกล่าวให้ปรากฏพื้นที่ใหม่สำหรับแสดงรายการผู้รับการสำรวจทางจอภาพทั้งหมดใน ITFair.txt ในรูปแบบตาราง
  - ต้องการให้มีปุ่ม 'ผลสำรวจ' และเมื่อคลิกปุ่มดังกล่าวให้ปรากฏพื้นที่ใหม่สำหรับแสดงผลการสำรวจโดยประมวลผลจาก ITFair.txt ประกอบด้วยจำนวนผู้รับการสำรวจ ระดับคะแนนเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน รายการที่ได้คะแนนมากที่สุด และรายการที่ได้คะแนนน้อยสุด ทางจอภาพ

3. จงเขียนโปรแกรมชื่อ EX2\_3 เพื่อแสดงการวาดรูปที่กำหนดด้วยค่าคงที่ โดยมีรายละเอียดกำหนดดังนี้

- ต้องการให้มีฟอร์มหลักสำหรับแสดงการวาดรูปด้วยคอมโพเนนต์เมนูบาร์และเมนูทูลบาร์ และคอมโพเนนต์ StatusBar StatusBar บอกสถานะของทุกฟอร์ม พร้อมทั้งข้อมูลผู้จัดทำโปรแกรมนี้
- เมนูบาร์/ทูลบาร์ กำหนดคลิกเพื่อแสดงข้อมูลแต่ละฟอร์มเพื่อให้สามารถวาดรูปได้ดังนี้
  - 1) รูปคน
  - 2) รูปสัตว์
  - 3) รูปวิว