ใบงานที่ 17

การเขียนโปรแกรมกราฟฟิกส์ด้วย GDI+ (1)

1. **กล่าวนำ**

ใบงานนี้ มีวัตถุประสงค์ เพื่อให้นักศึกษา ได้รู้จักกับ GDI+ ซึ่งจะช่วยให้นักษาสามารถ

* อธิบายระบบพิกัดใน GDI+ ได้
* สามารถใช้คำสั่งพื้นฐานของ GDI+ ในการวาดกราฟฟิกส์อย่างง่ายได้

1. **เนื้อหา**

**2.1 เกี่ยวกับ GDI+**

GDI+ (Graphics Device Interface Plus) เป็นกราฟฟิกส์ไลบรารี่ ซึ่ง Microsoft จัดเตรียมไว้ให้นักพัฒนาโปรแกรมได้ใช้ทำงานด้านกราฟฟิกส์ โดยที่ GDI+ จะมีลักษณะเป็น OOP เต็มตัว

ก่อนที่จะมี GDI+ ไมโครซอฟท์มีไลบรารี่ชื่อ GDI ซึ่งอยู่ในไฟล์ gdi32.dll ติดตั้งมาบน Windows ทุกตัวจนกระทั่งมาถึงยุค Windows XP ทางไมโครซอฟท์ก็ได้ปรับปรุง GDI ให้มีความสามารถมากขึ้น โดยให้ชื่อว่า GDI+

**2.2 ความสามารถของ GDI+**

ความสามารถของ GDI+ จะมีอยู่ 3 ส่วนหลักๆ คือ

* Vector **graphics** เป็นการสร้างกราฟฟิกส์ 2 มิติ ด้วยรูปทรงเรขาคณิตพื้นฐาน เช่น เส้นตรง(Line) เส้นโค้ง (Curve) สี่เหลี่ยม(Rectangle) วงรี (Ellipse) เป็นต้น
* **Imaging** นอกจากความสามารถในการวาดภาพด้วยรูปทรงเรขาคณิตแล้ว GDI+ ยังมีความสามารถในด้านการ แสดงผล, Load/Save ไฟล์ภาพชนิดต่างๆ ในรูปของ bitmap (bitmap ในที่นี้หมายถึงไฟล์ภาพทุกชนิด ที่แสดงเป็น pixel และไม่สามารถอธิบายด้วยรูปทรงเรขาคณิตอย่างง่ายๆ ได้ เช่น ภาพถ่าย ภาพสแกน เป็นต้น)
* **Typology** เป็นการจัดการเกี่ยวกับฟอนต์ ซึ่งมีความสามารถในการแสดงผลแบบ anti aliasing ทำให้ขอบของตัวอักษรดูเรียบขึ้นมากกว่าการเขียนโปรแกรมด้วย GDI

**2.3 ระบบพิกัดใน GDI+**

ใน GDI+ จะมีระบบพิกัดที่ใช้งานอยู่เป็นจำนวน 3 ระบบ คือ World coordinate, Page coordinate และ Device coordinate

World coordinate จะทำงานกับระบบหน่วยที่ใช้ในชีวิตประจำวัน เช่น หน่วยมิลลิเมตร หรือ หน่วยนิ้ว เป็นต้น

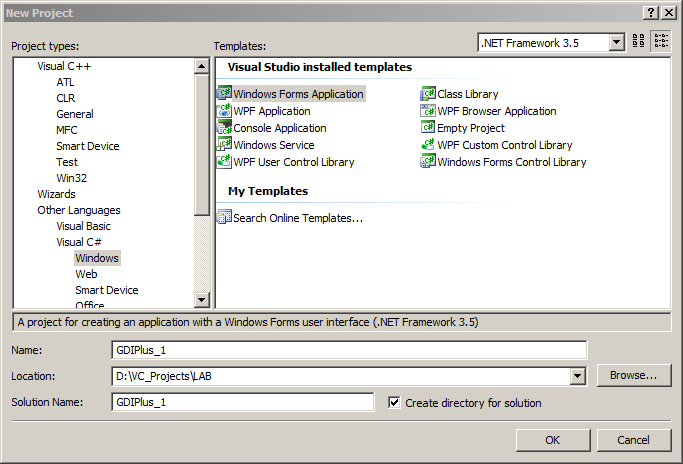
Page coordinate จะทำงานกับระบบ coordinate บนหน้ากระดาษ และจะสามารถพิมพ์ออกทางเครื่องพิมพ์ตามขนาดที่กำหนดบน pagecoordinate

Device coordinate จะทำงานกับ pixel บนจอหรือบนเครื่องพิมพ์

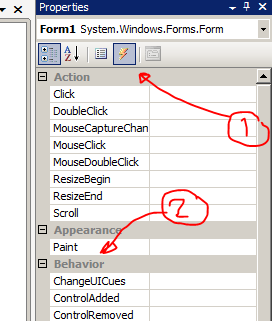
2.4 C-sharp.NET

เพื่อให้สามารถเขียนโปรแกรมได้ง่ายยิ่งขึ้น ใบงานที่เกี่ยวกับ GDI+ ทั้งหมดจะเขียนด้วยภาษา C# เป็นหลัก อย่างไรก็ตาม นักศึกษาสามารถนำ GDI+ ไปใช้กับโปรแกรมได้ทุกภาษา (รวมถึง C++ ด้วย แต่ต้องมีการเตรียมสภาพแวดล้อมที่ยุ่งยากกว่า)

1. **การทดลองย่อย 1**  เริ่มต้นกับ C# และ GDI+
   1. เรียกโปรแกรม Visual Studio 2008
   2. สร้าง Project ใหม่ เป็นชนิด C#

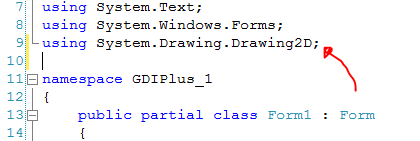


* 1. เมื่อ Wizard สร้าง Project เสร็จแล้ว จะนำเรามาที่หน้าต่าง Form1.cs[Design] ให้คลิกที่ปุ่ม Events ของ Properties pane ตาม (1) และ Double click ที่ Paint ตาม (2)

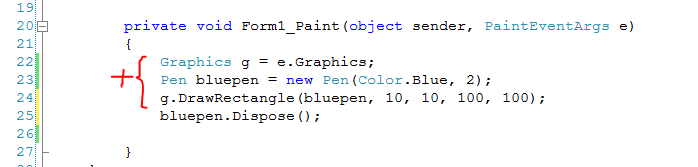


* 1. เพื่อให้โปรแกรมของเราสามารถใช้งาน GDI+ ในการวาดภาพ 2D ได้ ให้ทำการเพิ่ม

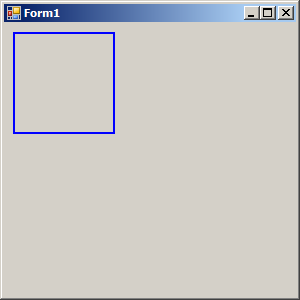
using System.Drawing.Drawing2D; ลงในบรรทัดที่ 9 ดังรูป



* 1. ให้เพิ่มบรรทัดต่อไปนี้ลงในฟังก์ชัน private void Form1\_Paint(object sender, PaintEventArgs e) (ในที่นี้จะอธิบายโปรแกรมไปด้วย การเพิ่มจริงๆ ให้ทำตามรูปข้างล่าง)
     1. สร้าง Object ของกราฟิกส์ โดยคำสั่ง Graphics g = e.Graphics;ซึ่ง Object ชื่อ e ถูกส่งผ่านมาทาง argument ของฟังก์ชัน
     2. เพิ่มออบเจกต์ของปากกา สีน้ำเงินขนาด 2 พิกเซล ด้วยคำสั่ง Pen = new Pen(Color.Blue, 2);
     3. วาดสี่เหลี่ยมด้วยคำสั่ง g.DrawRectangle(bluepen, 10, 10, 100, 100);
     4. คืนหน่วยความจำให้ระบบโดยการลบออบเจ็กต์ปากกาสีน้ำเงิน โดยคำสั่ง bluepen.Dispose();



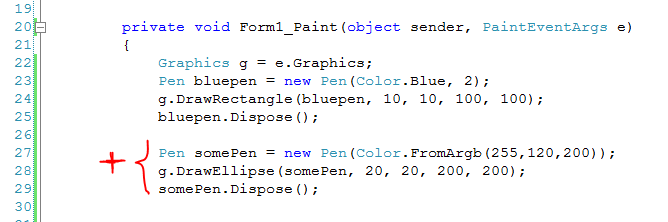
* 1. ทดลอง Build และ Run โปรแกรม



1. **การทดลองย่อย 2** การใช้สี

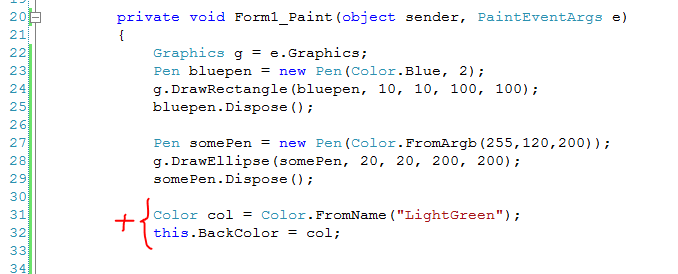
**4.1 โดยการผสมค่าสี**

เพิ่ม Code ต่อไปนี้ลงในฟังก์ชัน private void Form1\_Paint(object sender, PaintEventArgs e) แล้วทดลอง Run โปรแกรม



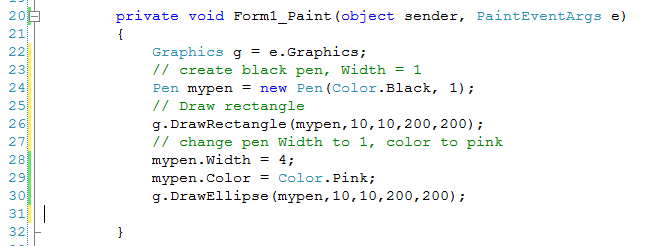
**4.2 โดยการใช้ methode FromName**

เพิ่ม Code ต่อไปนี้ลงในฟังก์ชัน private void Form1\_Paint(object sender, PaintEventArgs e) แล้วทดลอง Run โปรแกรม

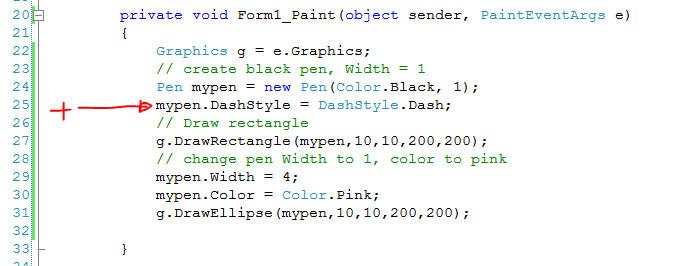


1. **การทดลองย่อย 3** การใช้ปากกา

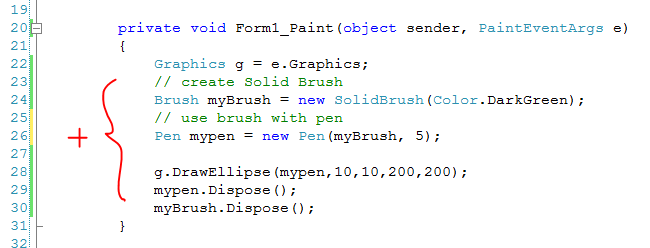
5.1 การทดลองเปลี่ยนขนาดและสีของปากกา โดยใช้ properties Width และ Color



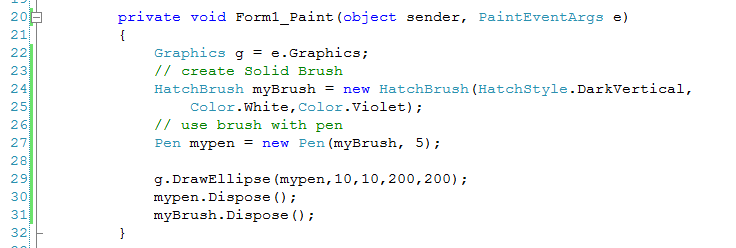
5.2 เปลี่ยนชนิดของปากกาเป็นเส้นประ



5.3 ใช้ Pen ร่วมกับ Brush



5.4 ใช้ Pen ร่วมกับ HatchBrush เพื่อสร้างลายเส้นแบบต่างๆ



ทดลองเปลี่ยน Color และ HatchStyle เป็นแบบต่างๆ เพื่อดูความเปลี่ยนแปลง