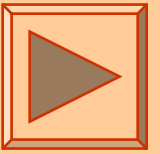


松山大学 経営学部

情報処理論（応用）



第13回 グラフィックス処理（2）



講師 檀 裕也

<http://www.cc.matsuyama-u.ac.jp/~dan/application/>

2006年11月10日

出席確認

- 出席確認フォームから学籍番号および氏名を送信せよ。

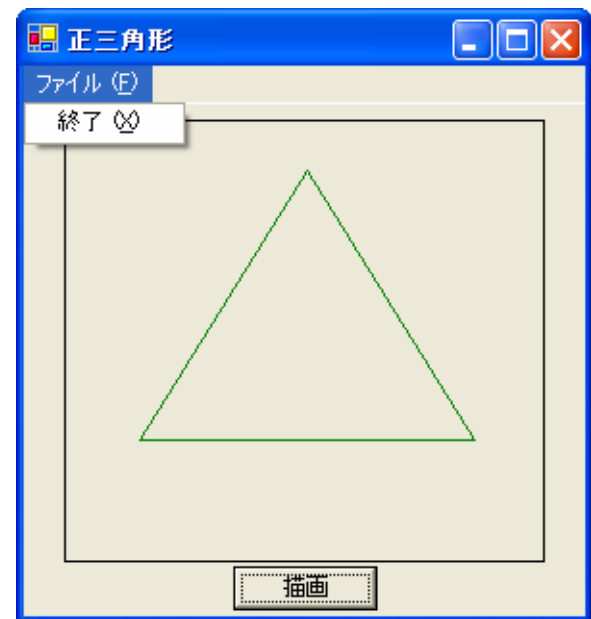
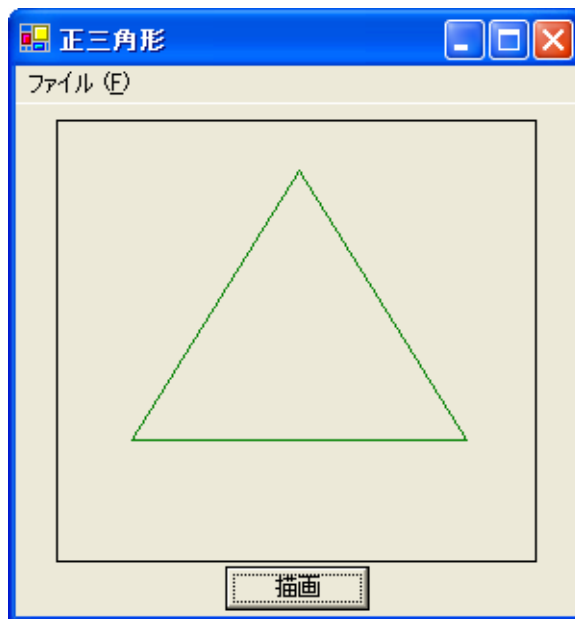
出席確認フォーム

<http://www.cc.matsuyama-u.ac.jp/~dan/application/attendance.html>

前回の課題

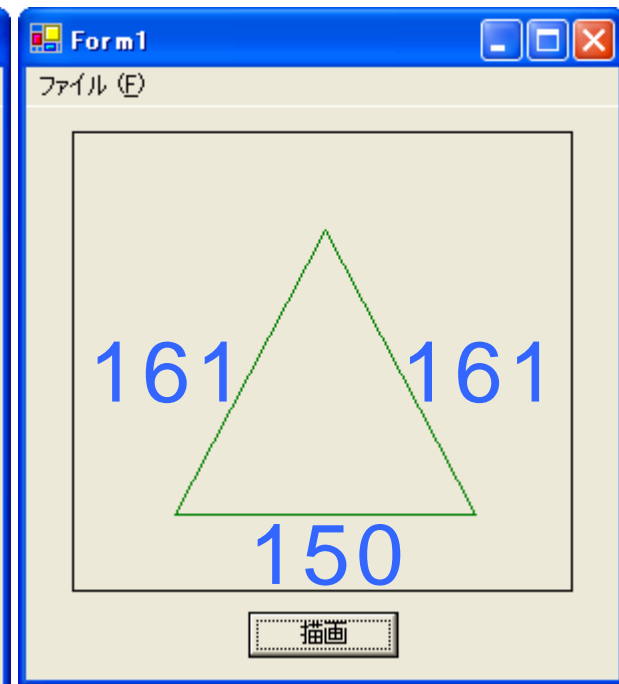
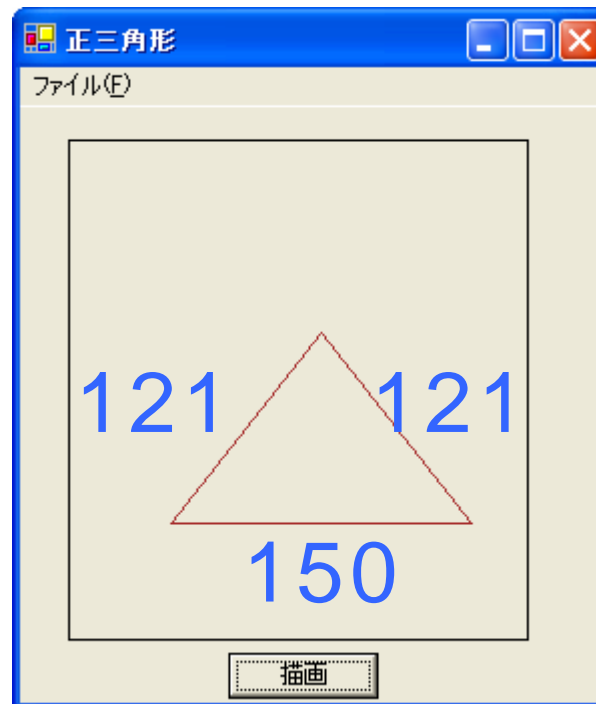
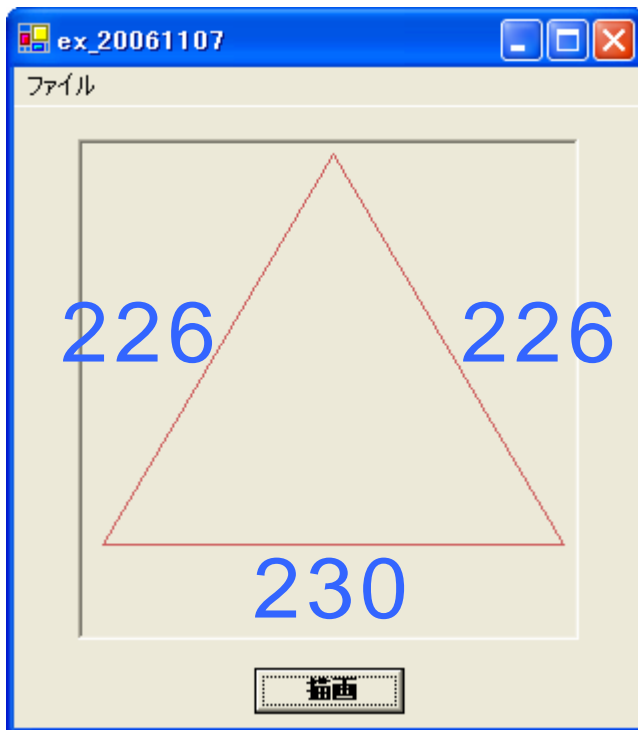
正三角形の描画プログラム (ex_20061107)

- 好きな色の線を使って正三角形を描け。



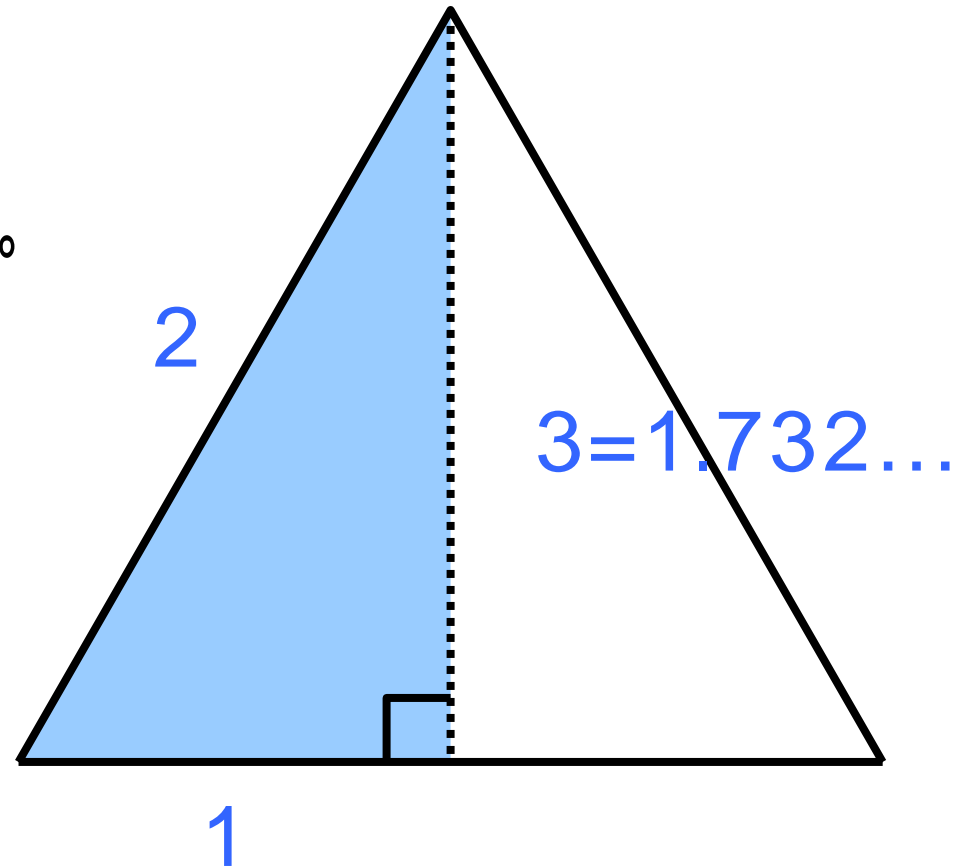
作品集

- 正三角形とは、3つの辺の長さが同じ三角形である。



考え方

- 例えば、底辺の長さが200の正三角形の高さは、約173となる。
- 正三角形の3つの頂点を計算すればよい。



解答例 (コード)

- DrawLine メソッドの引数に、計算で求めた正三角形の3つの頂点の座標を使えばよい。

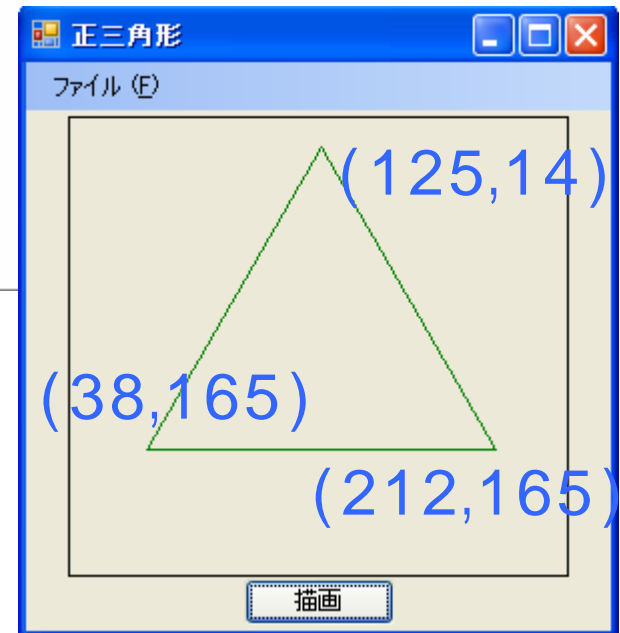
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim g As Graphics

    ' PictureBox の Graphics オブジェクトを生成する
    g = PictureBox1.CreateGraphics()

    ' 直線を描画する
    g.DrawLine(Pens.Green, 125, 14, 212, 165)
    g.DrawLine(Pens.Green, 212, 165, 38, 165)
    g.DrawLine(Pens.Green, 38, 165, 125, 14)

    ' リソース (オブジェクト) を解放する
    g.Dispose()

End Sub
```

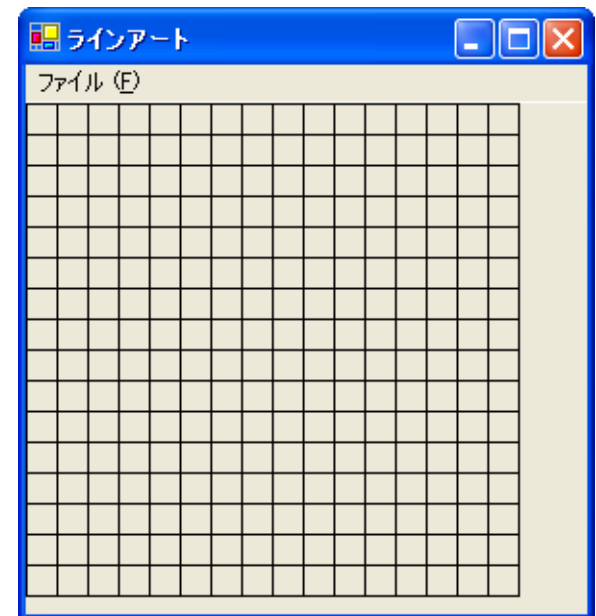


今回の予定

- コンピュータグラフィックス（続き）
 - コンピュータ上で図形を描くには？
- 到達目標
 - Windows におけるグラフィックス処理の機能を理解し、線分を描く機能を使って自由に多角形を描くことができる。

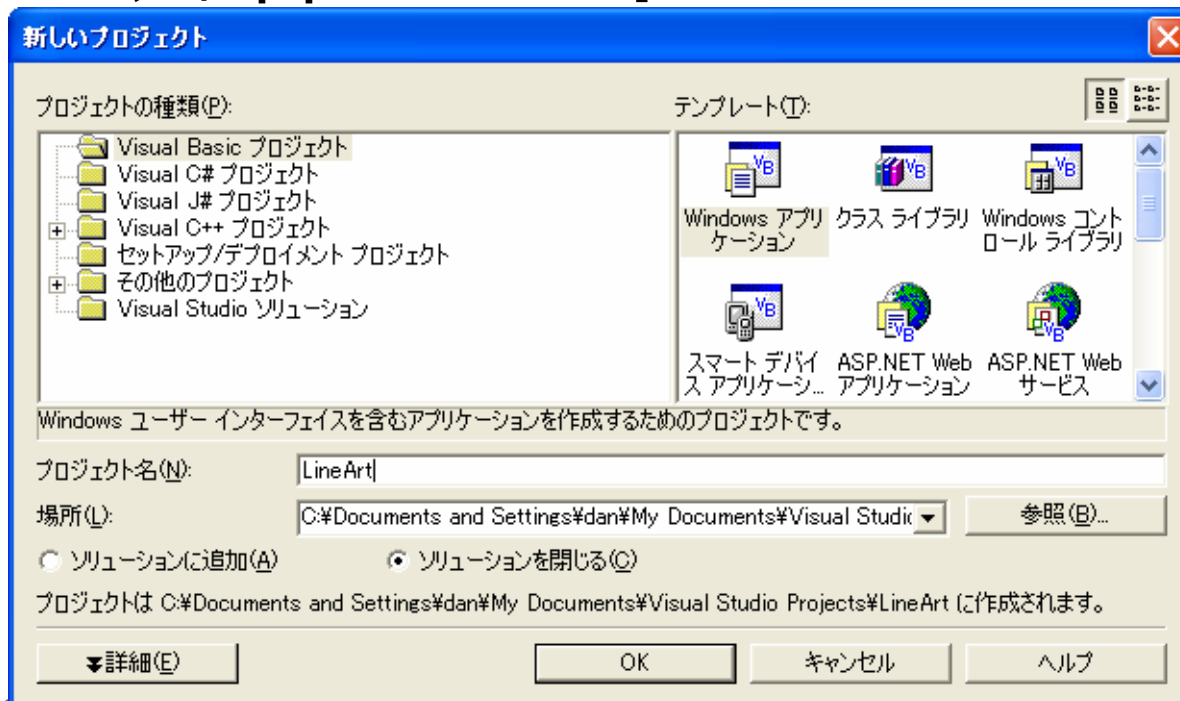
例題10

- 画面以上にマス目を描くプログラムを作成する
- 手順：
 - プロジェクトの新規作成
(LineArt)
 - 画面レイアウト
 - イベントハンドラの記述
 - 実行および動作確認

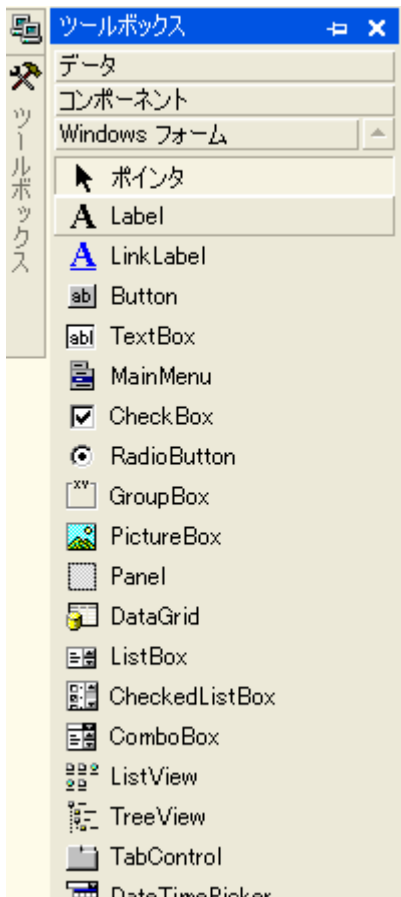


新しいプロジェクト

- プロジェクトの種類「Visual Basicプロジェクト」
- テンプレート「Windowsアプリケーション」
- プロジェクト名「LineArt」



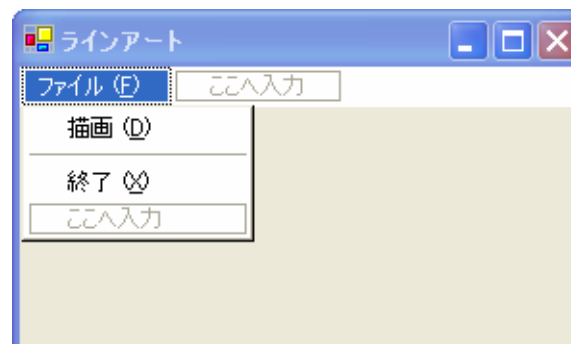
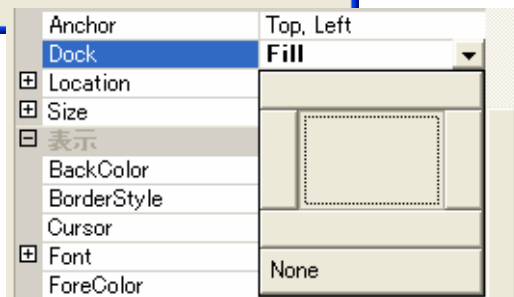
画面の構成要素



- ツールボックスの中から必要なコントロールを選択し、フォームに貼り付ける。
 - コントロールをダブルクリック
 - コントロールをフォーム上にドラッグ & ドロップ

フォームの構成要素

ピクチャボックスを配置し、メニューを設定する



MenuItem2

.Text = "描画 (&D)"

Click イベント コード記述へ

MenuItem4

.Text = "終了 (&X)"

Click イベント コード記述へ

コードの記述

描画コマンドのイベントハンドラを記述する

```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Windows フォーム デザイナで生成されたコード

    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click
        Dim g As Graphics
        Dim i As Integer

        ' PictureBox の Graphics オブジェクトを生成する
        g = PictureBox1.CreateGraphics()

        ' 横線を描画する
        For i = 0 To 256 Step 16
            g.DrawLine(Pens.Black, 0, i, 256, i)
        Next

        ' 縦線を描画する
        For i = 0 To 256 Step 16
            g.DrawLine(Pens.Black, i, 0, i, 256)
        Next

        ' リソース (オブジェクト) を解放する
        g.Dispose()

    End Sub

    Private Sub MenuItem4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem4.Click
        Me.Dispose()
    End Sub
End Class
```

For ステートメント (基本形)

指定した回数だけ同じ処理を繰り返す

```
For i=1 To 10  
    処理  
Next
```

カウンタ変数を使い、初期値から始めて終了値まで回数を数える。左の例は10回の繰り返し。

i=1: 1回目の繰り返し処理

i=2: 2回目の繰り返し処理

.....

i=10: 10回目の繰り返し処理

For ステートメント (応用形)

- For ステートメントでは、カウンタ変数の増分を指定できる。省略すると、Step 1 とみなす。

```
For i=0 To 256 Step 16  
  (0, i) ~ (256, i)の線分  
Next
```

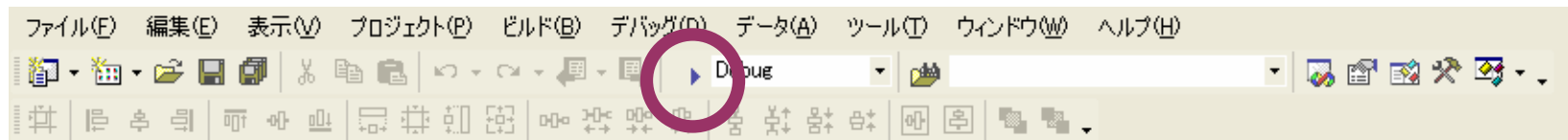
i= 0 のときの処理
i=16 のときの処理
i=32 のときの処理

.....

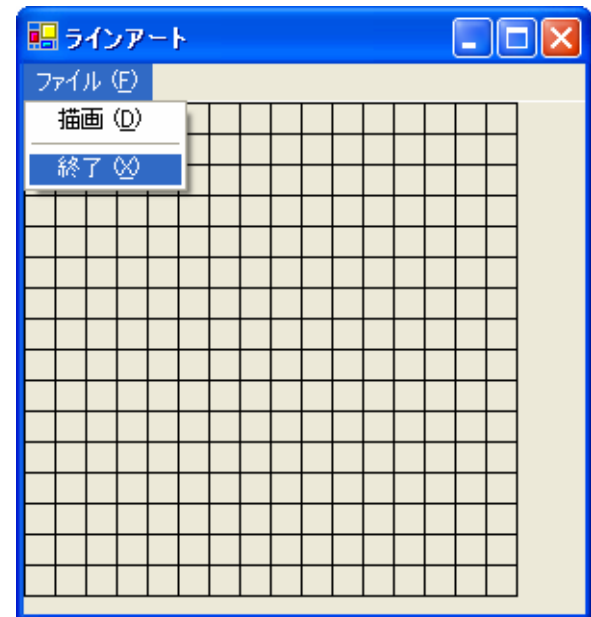
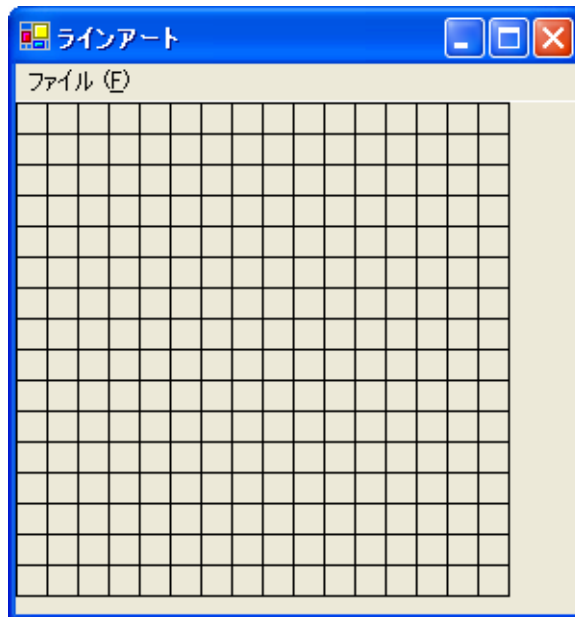
i=256 のときの処理
まで、17回の繰り返し



実行 (デバッグ)



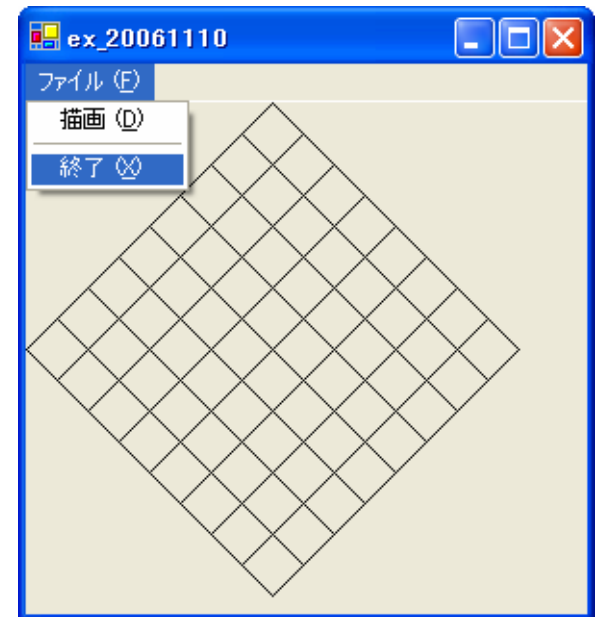
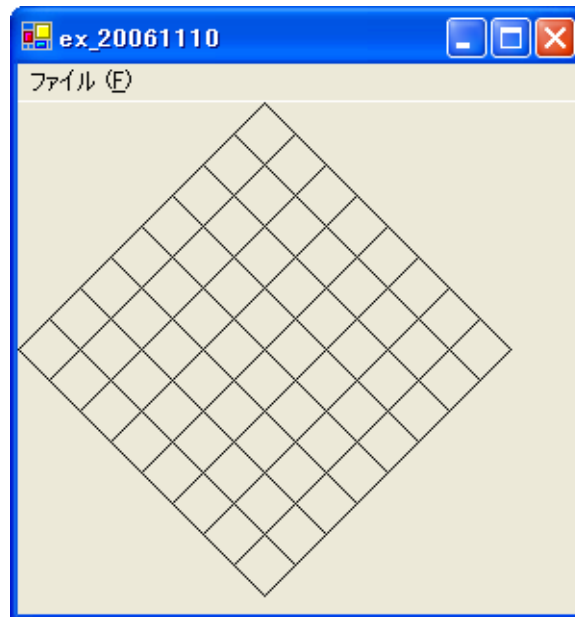
- [デバッグ] [開始]でプログラムを実行する
- ツールバーの**開始ボタン**をクリックしてもよい



課題

ラインアート (ex_20061110)

- 例題の図形を45度回転させて表示せよ。
ただし、マス目は何個でもよい。



まとめ

- For ステートメントを使って、複数の線分を効率よく描画する処理について学んだ。
 - DrawLine メソッド
 - For ~ Next ステートメント

次回予定

- グラフィックス (3)
 - Paint イベントの取得による再描画
 - 正多角形と三角関数