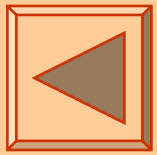
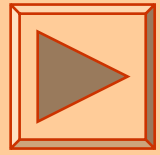
松山大学 経営学部

情報処理論（応用）



第12回　グラフィックス処理（1）



講師　檀 裕也

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

2006年11月 7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。

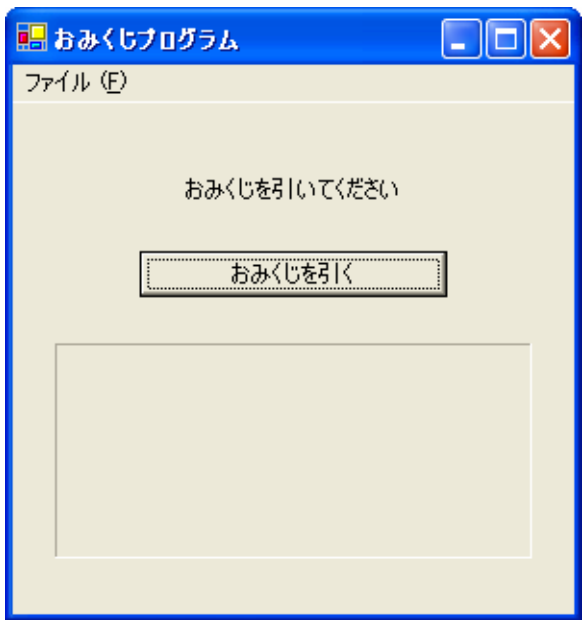
出席確認フォーム

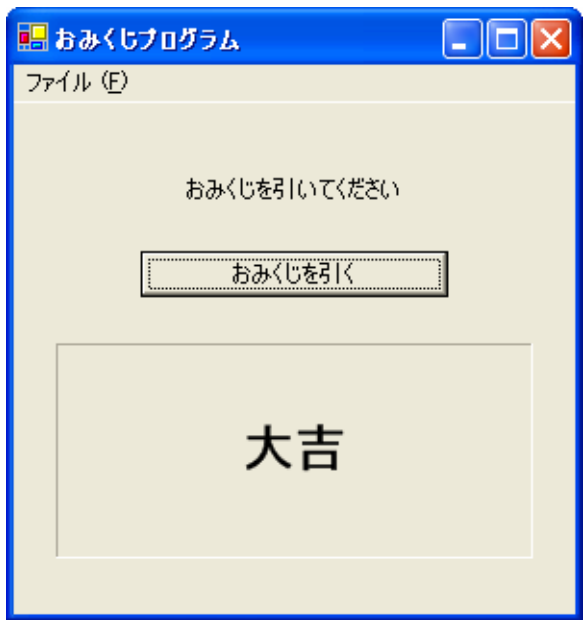
http://www.cc.matsuyama-u.ac.jp/~dan/application/attendance.html

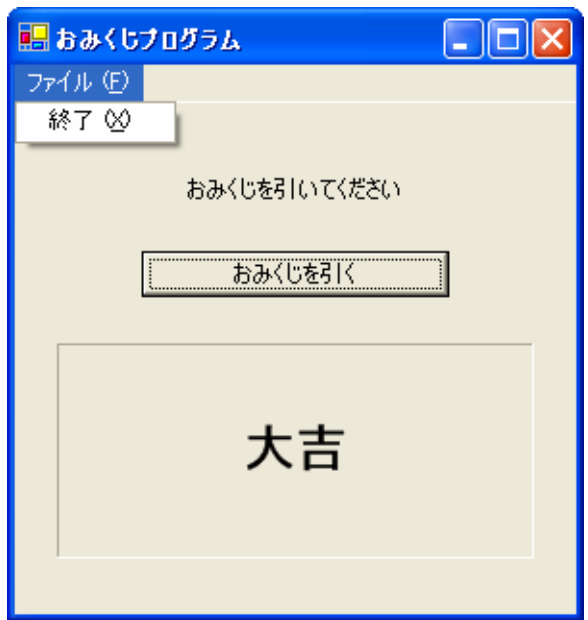
前回の課題

おみくじの統計（ex_20061031）

- 前回のおみくじプログラムについて、大吉･吉･凶･大凶の統計データを取得せよ。なお、画面は自由にレイアウトしてよい。

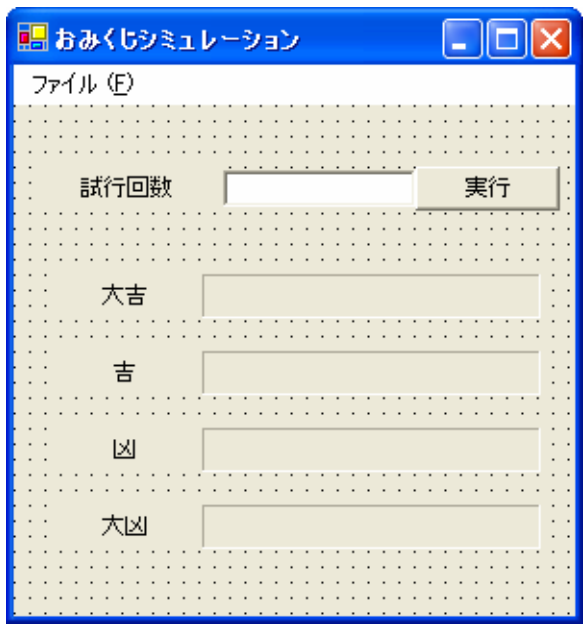






解答例（フォーム）

- 適切な位置にラベル・ボタン・テキストボックスを配置し、メインメニューをつける。



試行回数を入力して［実行］ボタンを押すと、それぞれの結果が出た回数を表示する。

解答例（コード前半）

- 必要な変数を宣言し、それぞれの結果を格納する変数を初期化（値をゼロにする）する。
- 整数値への変換には例外処理を記述する。

```
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 Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim r As New System.Random      '乱数生成
        Dim n As Integer                'おみくじ生成
        Dim a1, a2, a3, a4 As Integer   '各結果の出現回数
        Dim c As Integer                'シミュレーション回数
        Dim i As Integer                'カウンタ変数

        '変数を初期化する
        a1 = 0
        a2 = 0
        a3 = 0
        a4 = 0

        'テキストボックスの文字列を整数に変換する
        Try
            c = CInt(TextBox1.Text)
        Catch
            TextBox1.Text = ""
        End Try
```

解答例（コード後半）

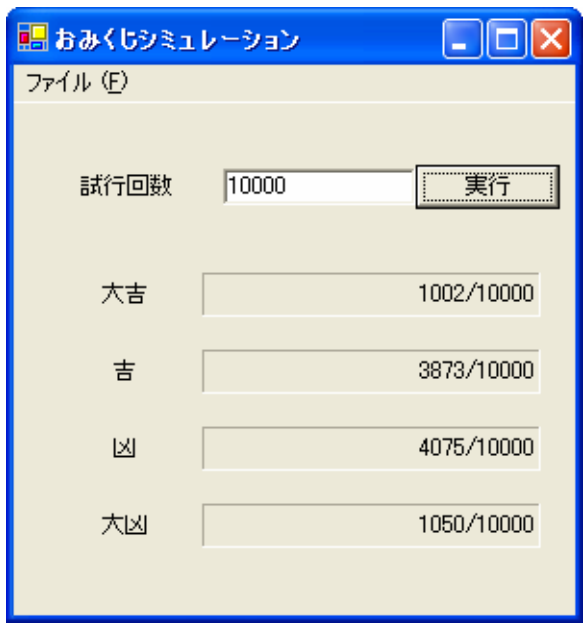
- それぞれの結果に応じて出現回数を記録する。
- 最後に、シミュレーションの結果を表示する。

```
        'シミュレーションの繰り返し処理（回数 c）
        For i = 1 To c
            n = r.Next() Mod 10
            If n = 0 Then
                a1 += 1     '大吉
            ElseIf n < 5 Then
                a2 += 1     '吉
            ElseIf n < 9 Then
                a3 += 1     '凶
            Else
                a4 += 1     '大凶
            End If
        Next

        '結果を表示する
        Label3.Text = a1.ToString() & "/" & c.ToString()
        Label4.Text = a2.ToString() & "/" & c.ToString()
        Label6.Text = a3.ToString() & "/" & c.ToString()
        Label8.Text = a4.ToString() & "/" & c.ToString()
    End Sub

    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click
        Me.Dispose()
    End Sub
End Class
```

結果の検討

- 指定した確率でそれぞれの結果が出現していることを確認せよ。また、試行回数を増やして傾向を観察せよ。

種類	確率
大吉	１０％
吉	４０％
凶	４０％
大凶	１０％



今回の予定

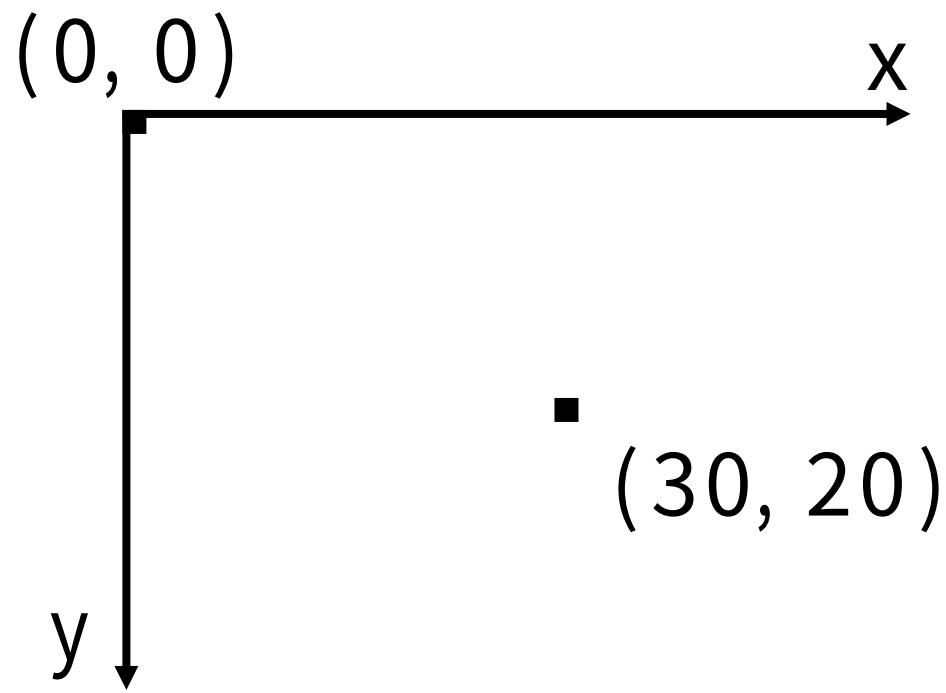
- コンピュータグラフィックス
  - コンピュータ上で図形を描くには？
- 到達目標
  - Windows におけるグラフィックス処理の機能を理解し、線分を描く機能を使って自由に多角形を描くことができる。

GDI+

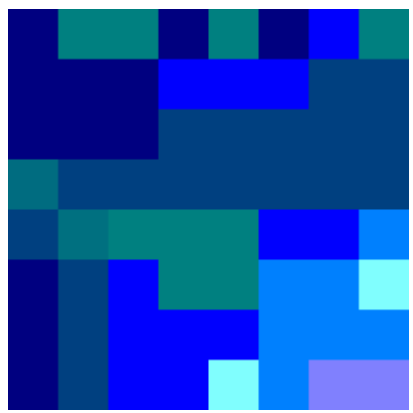
- Windows の標準的なグラフィックス機能
- .NET Framework から利用可能
- Graphics Device Interface

座標系

- 画素 (ピクセル) を単位とする画面上の座標系
- 画面上の点を2つの数値によって指定する



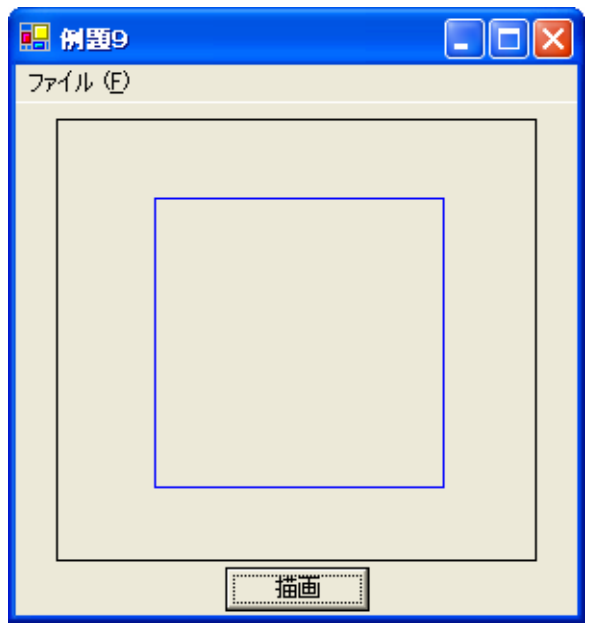
画面は色のついた
画素の集合である



例題9

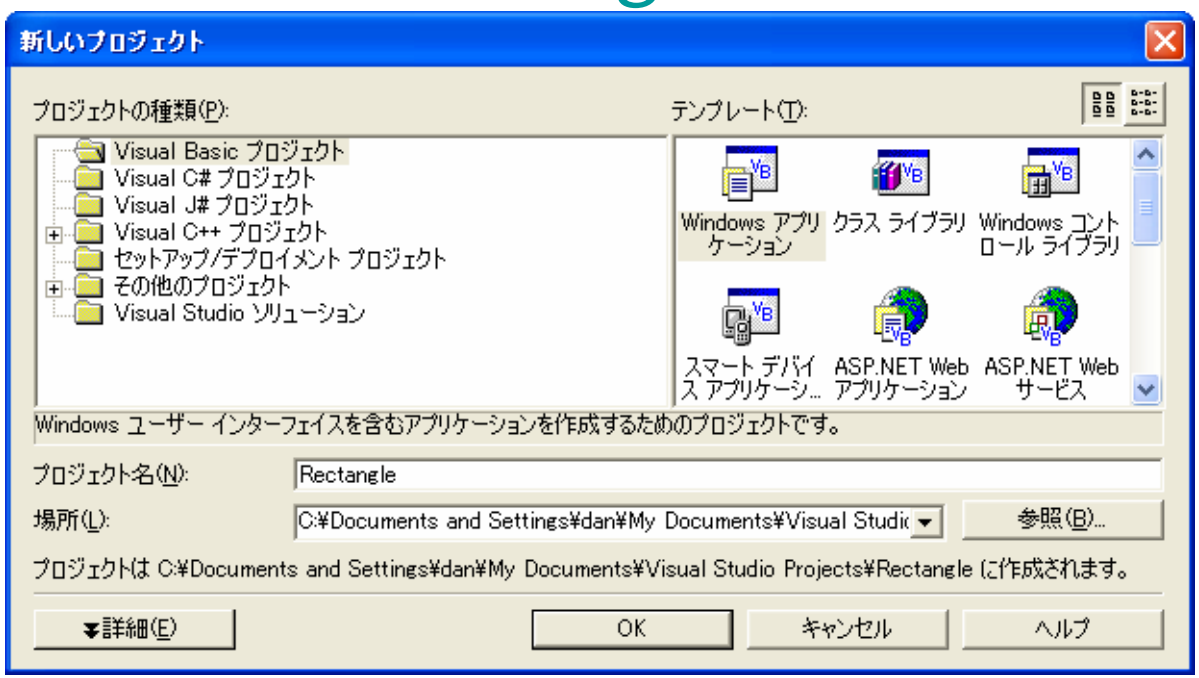
- 画面上に正方形を描くプログラムを作成する

- 手順:
  - プロジェクトの新規作成
    (Rectangle)
  - 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

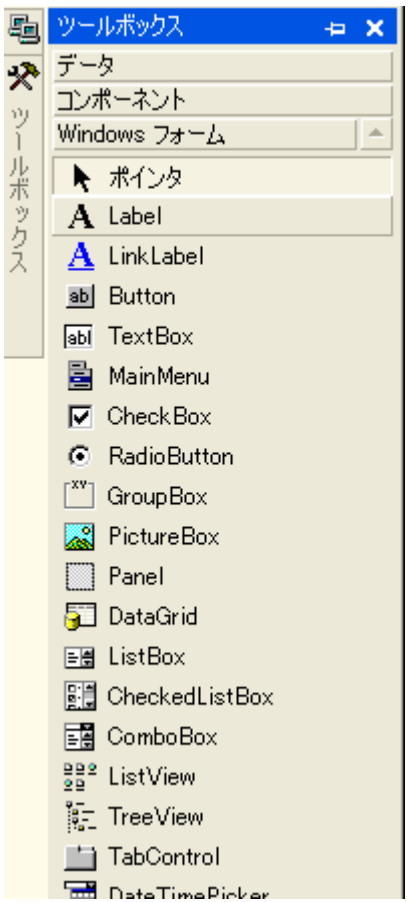


新しいプロジェクト

- 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「Rectangle」



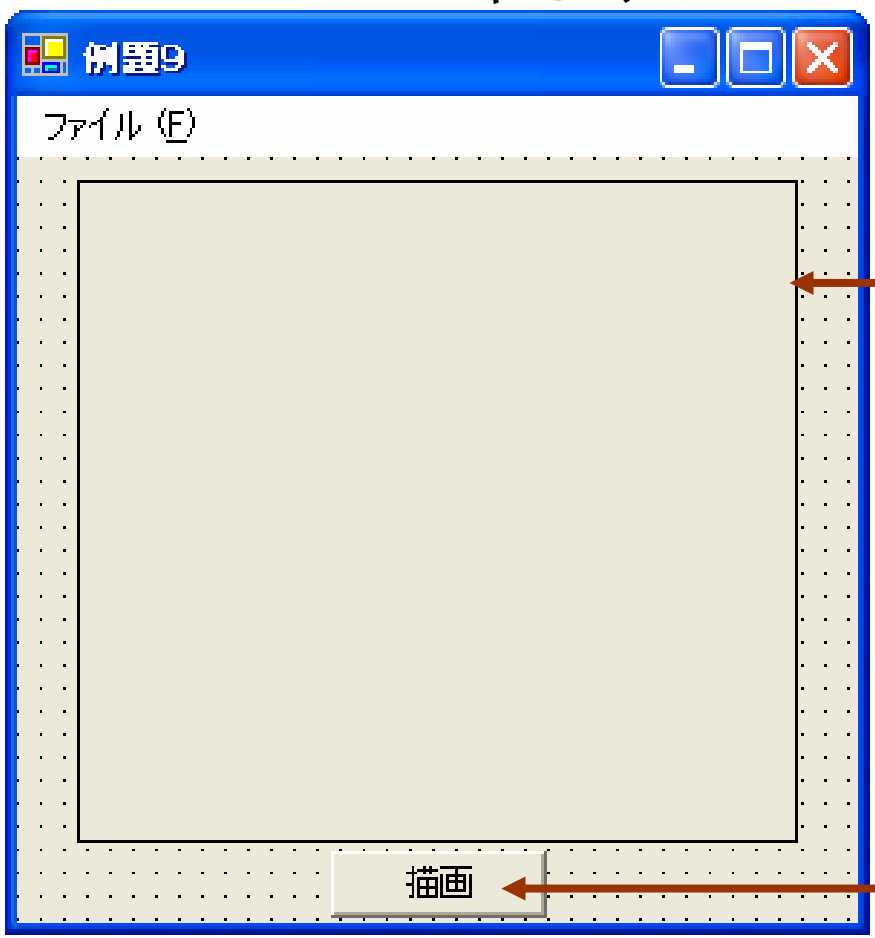
画面の構成要素



- 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
  - 新登場：PictureBox

フォームの構成要素

ピクチャボックスとボタンを配置する



PictureBox1
.BorderStyle：FixedSingle
.Size：250, 230

Button1
.Text = "描画"
Click イベント → コード記述

コードの記述

ボタンのイベントハンドラとしてコードを記述する

```
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 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.Blue, 50, 40, 200, 40)
        g.DrawLine(Pens.Blue, 200, 40, 200, 190)
        g.DrawLine(Pens.Blue, 200, 190, 50, 190)
        g.DrawLine(Pens.Blue, 50, 190, 50, 40)

        ' リソース（オブジェクト）を開放する
        g.Dispose()
    End Sub

    Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem2.Click
        Me.Dispose()
    End Sub
End Class
```

基本機能

- 描画機能の Graphics クラスの利用を宣言

  ```
  Dim g As Graphics
  ```

- PictureBox1 に Graphics オブジェクトを生成し、描画機能を使えるようにする（初期化）

  ```
  g = PictureBox1.CreateGraphics()
  ```

- Graphicsオブジェクトを破棄し資源を開放する

  ```
  g.Dispose()
  ```

DrawLine メソッドの解説

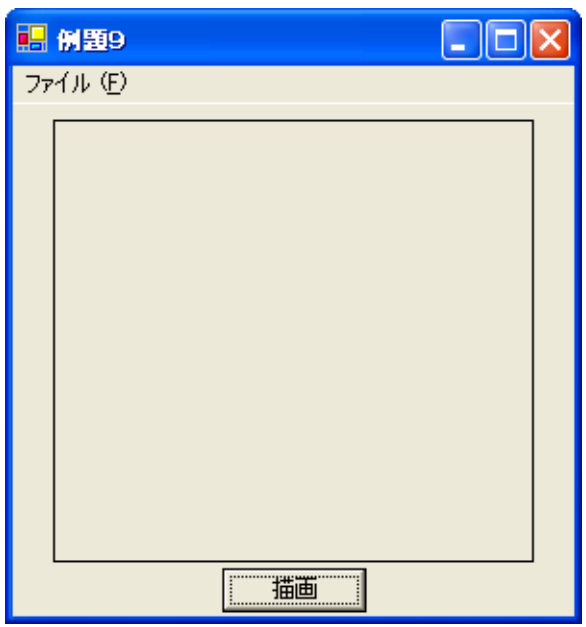
g.DrawLine(*Pen*, x_1, y_1, x_2, y_2)

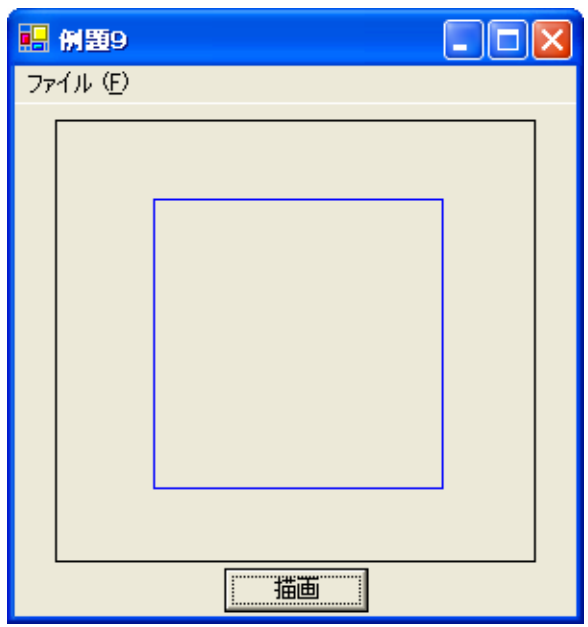
- *Pen* で指定したペンで座標(x_1, y_1)から座標(x_2, y_2)までの線分を描く。
- Pens クラスには、さまざまな色のペン（幅1ピクセル）がプロパティとして定義されている。
  - AliceBlue, AntiqueWhite, Aqua, Aquamarine, Azure, …詳細はヘルプを参照せよ
- 使用例:

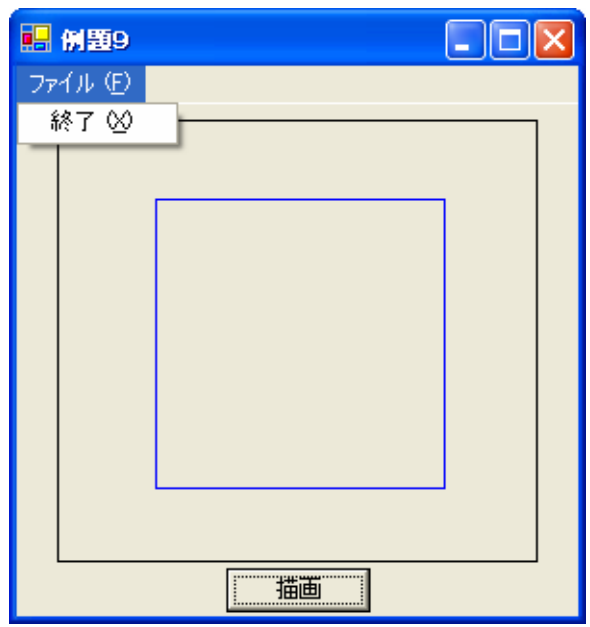
```
g.DrawLine(Pens.Blue, 50, 40, 200, 40)
```

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

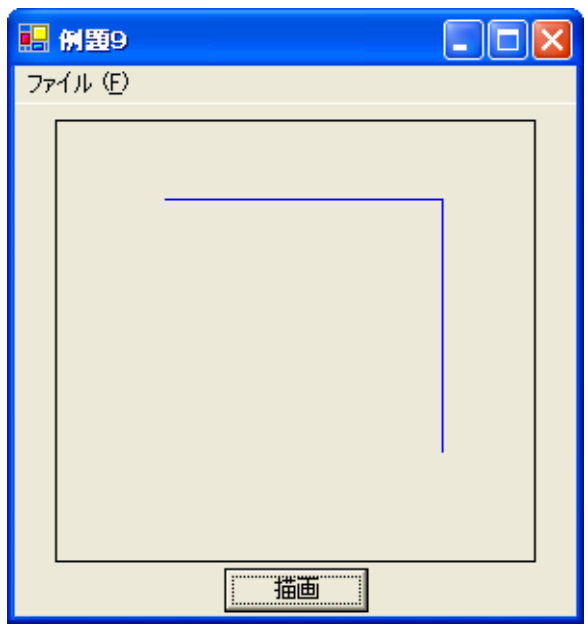






注意！

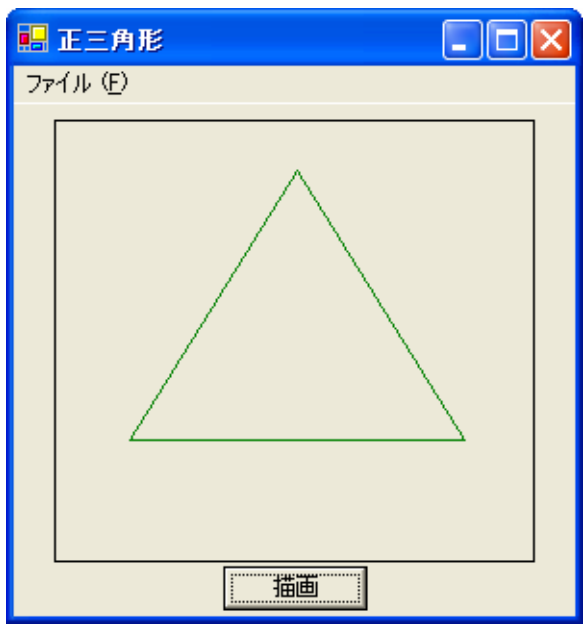
- フォームがデスクトップ画面からはみ出したり、他のウィンドウに重なったりすると、描画された図形が消えてしまう。この場合、再描画する必要がある。

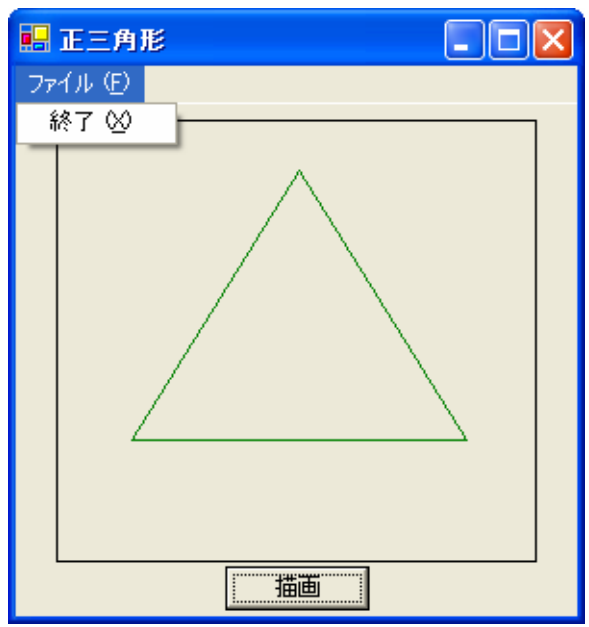


課題

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。





まとめ

- Windows アプリケーションで図形を描画するための基本を学んだ。
  - GDI+
  - Graphics クラス

次回予定

- グラフィックス（2）
  - Paint イベントの取得による再描画
  - 正多角形と三角関数