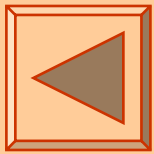


松山大学 経営学部

情報処理論（応用）



第14回 グラフィックス処理（3）



講師 檀 裕也

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

2006年11月14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。

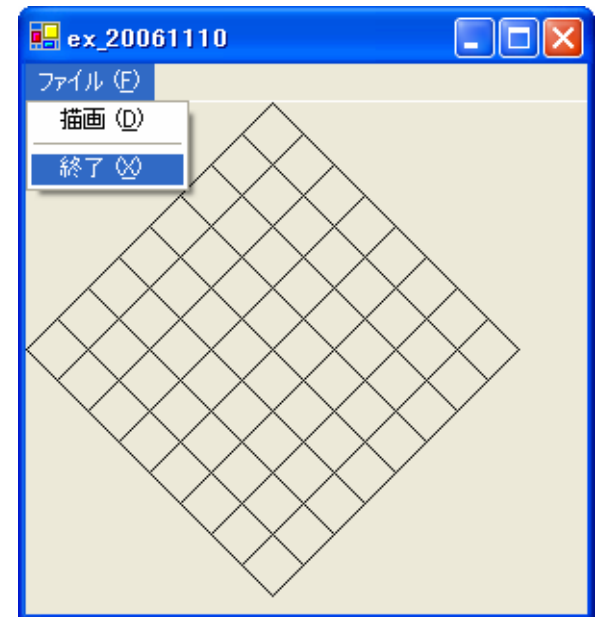
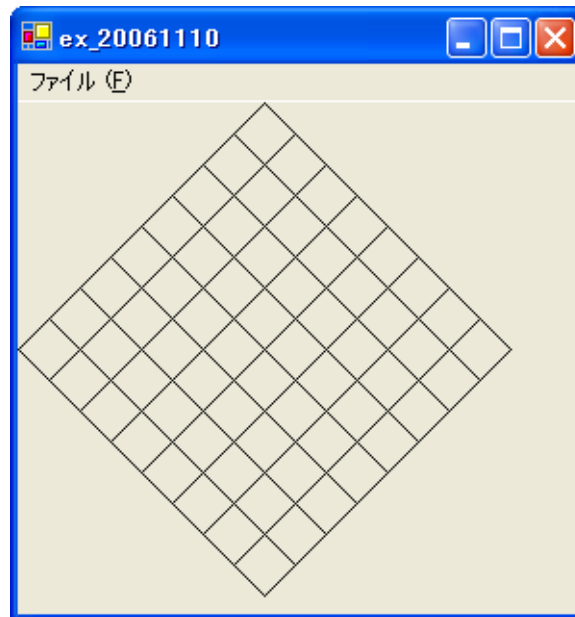
出席確認フォーム

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

前回の課題

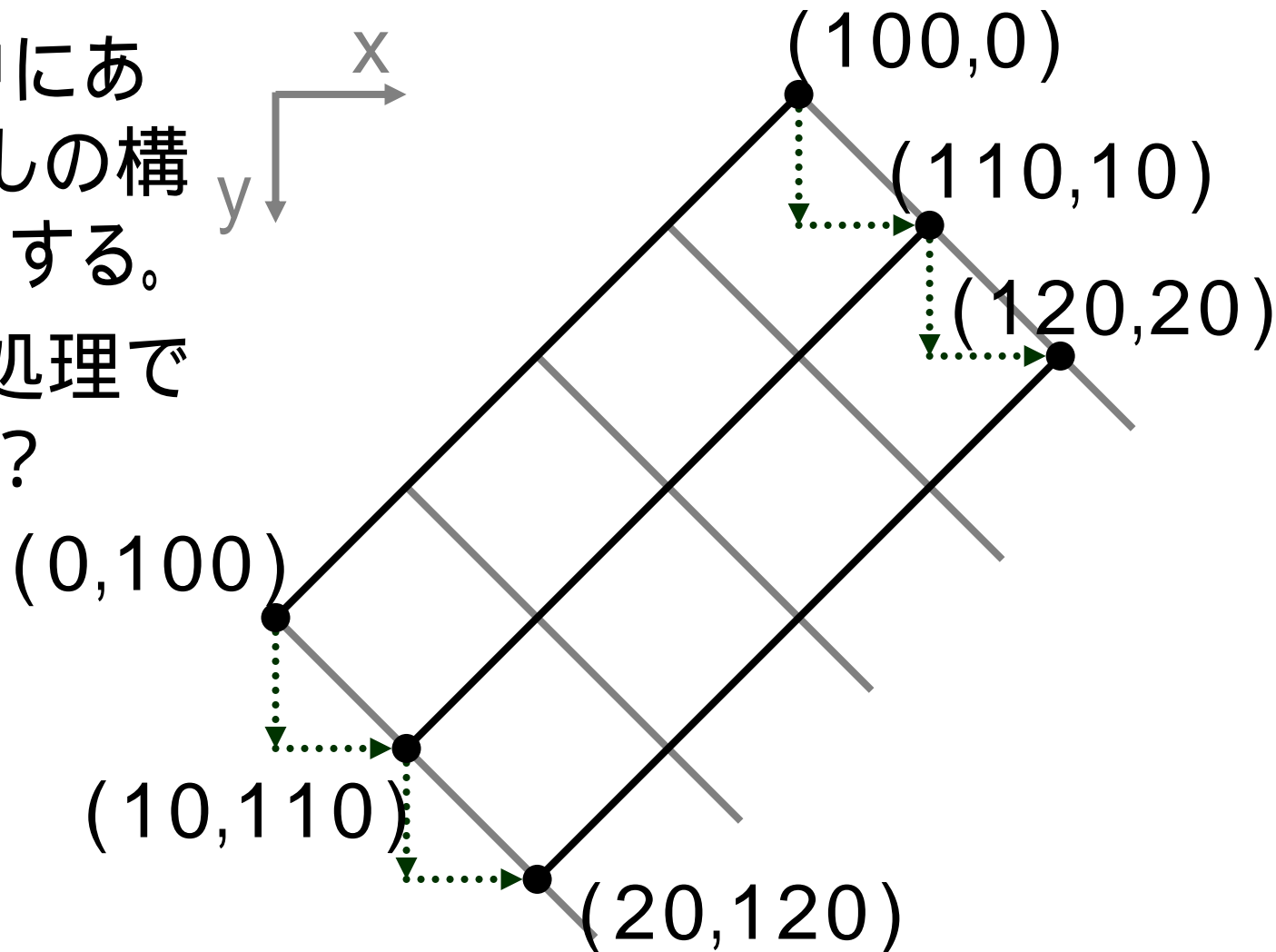
ラインアート (ex_20061110)

- 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。



考えかた

- 図形の中にある繰り返しの構造に着目する。
- まとめて処理できないか？



コード化

- 線分を描画する処理

```
g.DrawLine(Pen, 100, 0, 0, 100)
```

```
g.DrawLine(Pen, 110, 10, 10, 120)
```

```
g.DrawLine(Pen, 120, 20, 20, 130)
```

...以下省略...

- For ステートメントの利用

```
For i=0 To 100 Step 10
```

```
    g.DrawLine(Pen, 100 + i, i, i, 100 + i)
```

```
Next
```

解答例 (コード)

- 繰り返し処理の For ステートメントを使って複数の線分をまとめて描くとよい。

```
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()

        ' 斜線 (順方向) を描画する
        For i = 0 To 128 Step 16
            g.DrawLine(Pens.Black, 128 + i, i, i, 128 + i)
        Next

        ' 斜線 (逆方向) を描画する
        For i = 0 To 128 Step 16
            g.DrawLine(Pens.Black, 128 - i, i, 256 - i, 128 + i)
        Next

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

    End Sub

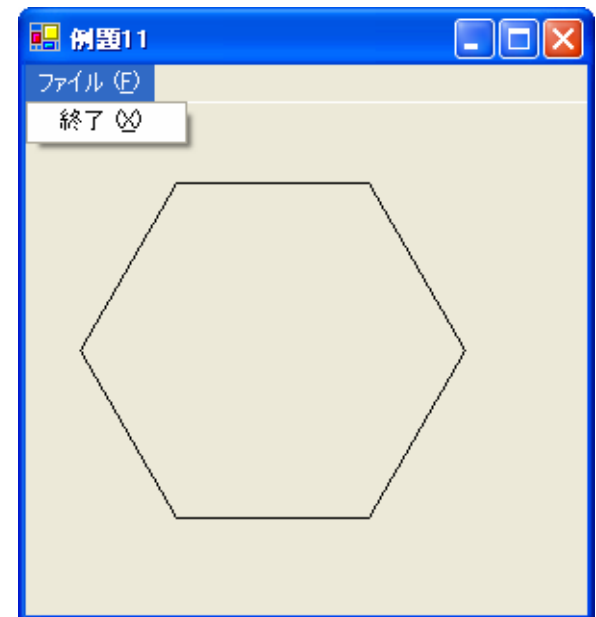
    Private Sub MenuItem4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MenuItem4.Click
        Me.Dispose()
    End Sub
End Class
```

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。

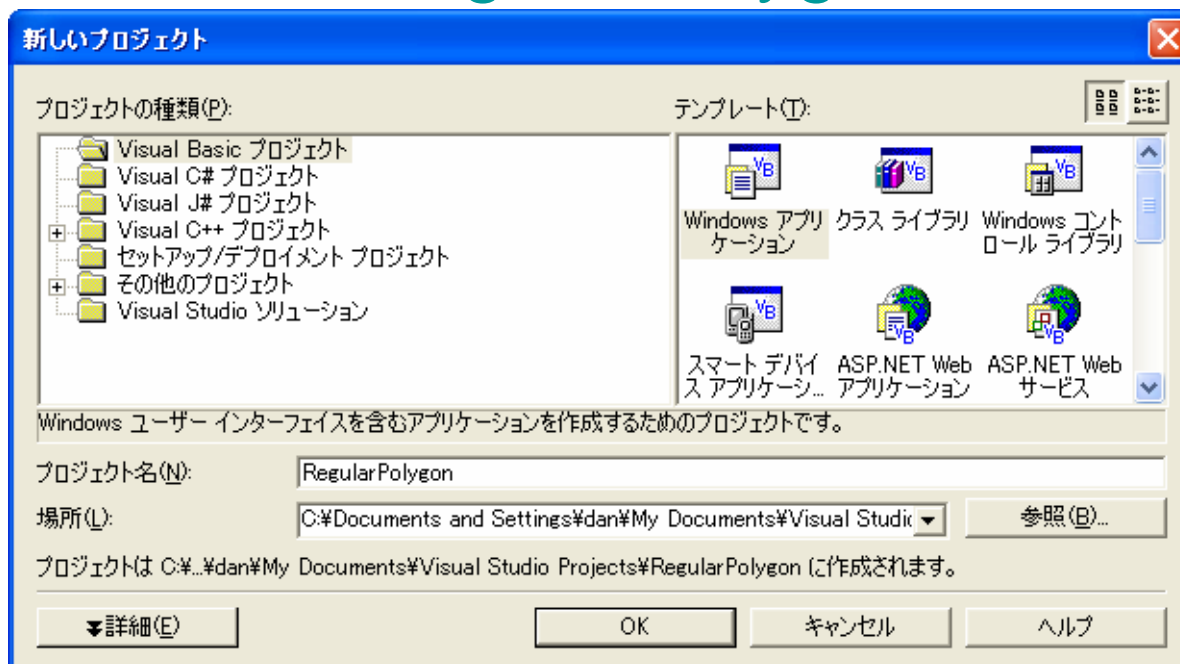
例題11

- 計算によって正多角形を描くプログラム
- 手順：
 - プロジェクトの新規作成
(RegularPolygon)
 - 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

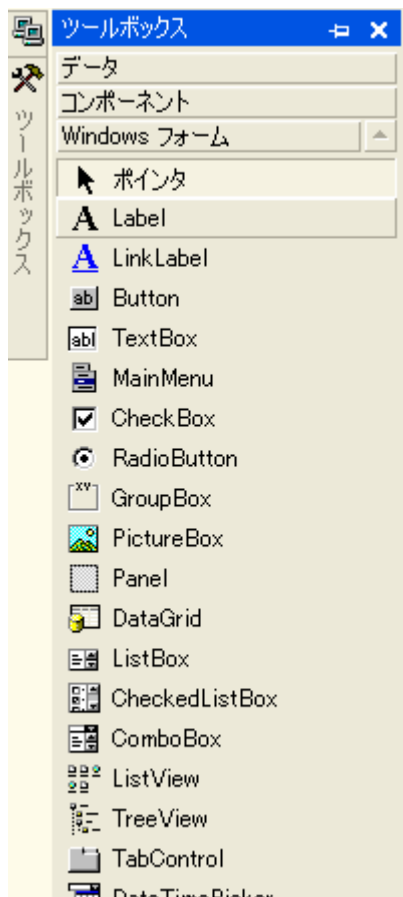


新しいプロジェクト

- 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「RegularPolygon」



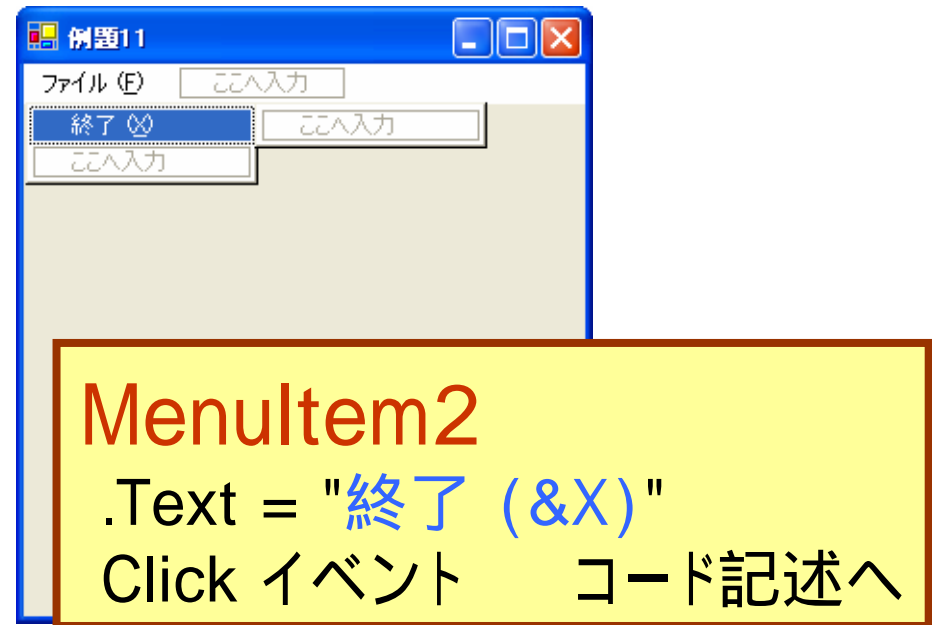
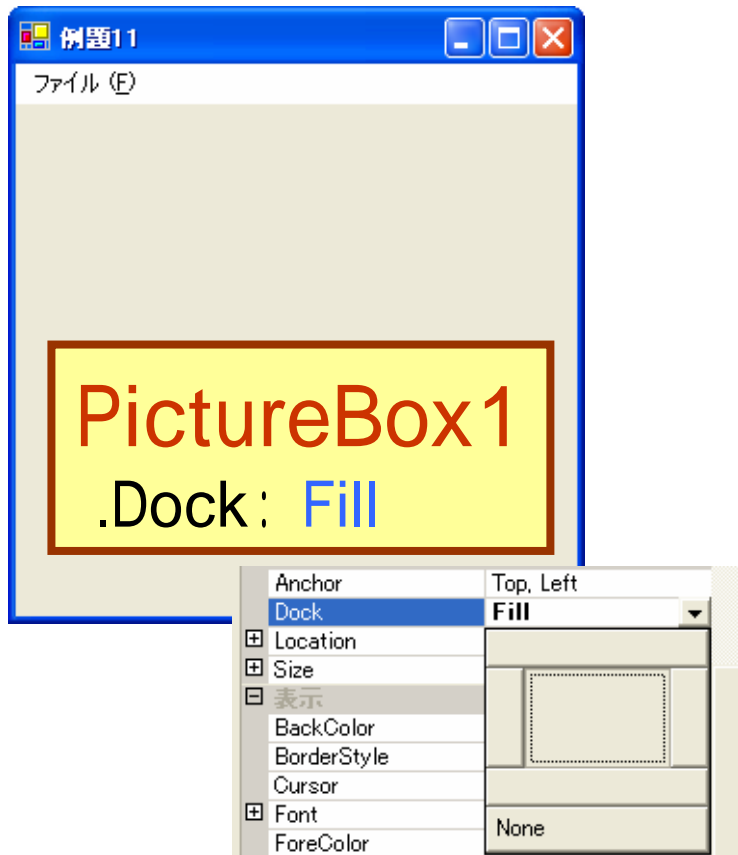
画面の構成要素



- 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

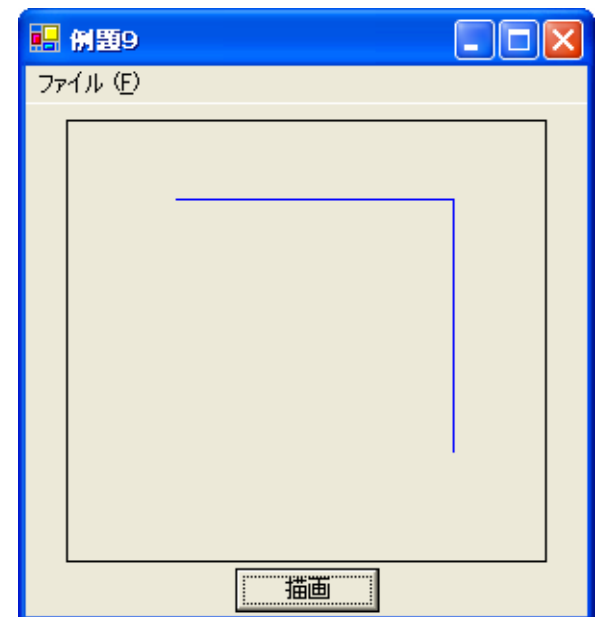
フォームの構成要素

ピクチャボックスを配置し、メニューを設定する



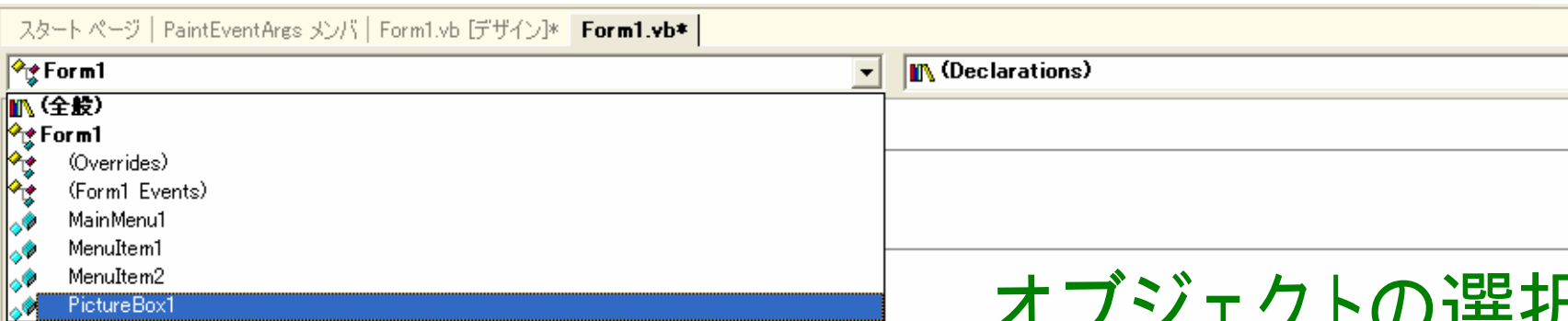
Paint イベント

- プログラムの実行時に、オブジェクトを再描画する必要があるタイミングで Paint イベントが発生する。
 - プログラム開始のとき
 - ウィンドウの大きさを変えるとき
 - 他のウィンドウと重なったとき
- Paint イベントの発生時にオブジェクトを再描画する。

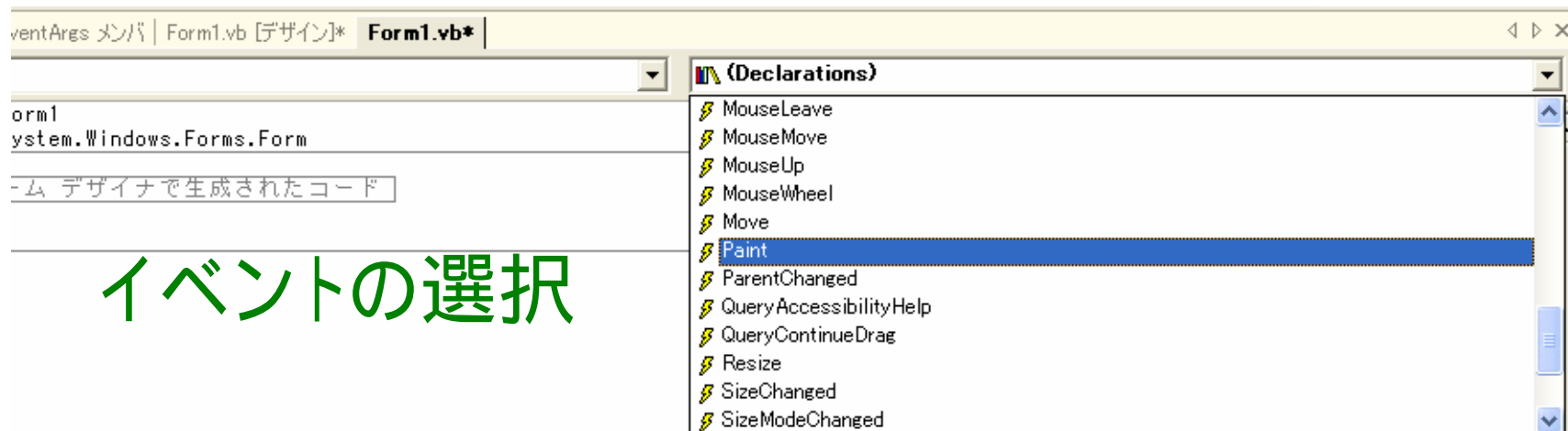


Paint イベントの処理コード

- コード画面上部のプルダウンメニューから PictureBox および Paint イベントを選択する。



オブジェクトの選択



イベントの選択

コードの記述

PictureBox に対する Paint イベントハンドラ

```
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 PictureBox1_Paint(ByVal sender As Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
        Dim i As Integer          'カウンタ変数
        Dim n As Integer          '頂点数
        Dim x1, y1, x2, y2 As Integer '頂点の座標
        Dim theta As Double       '角度

        n = 6

        For i = 1 To n
            '座標の計算
            theta = 2 * Math.PI * i / n
            x1 = 128 + 100 * Math.Cos(theta)
            y1 = 128 + 100 * Math.Sin(theta)
            theta = 2 * Math.PI * (i + 1) / n
            x2 = 128 + 100 * Math.Cos(theta)
            y2 = 128 + 100 * Math.Sin(theta)

            '正多角形の辺を描画する
            e.Graphics.DrawLine(Pens.Black, x1, y1, x2, y2)
        Next
    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
```

三角関数

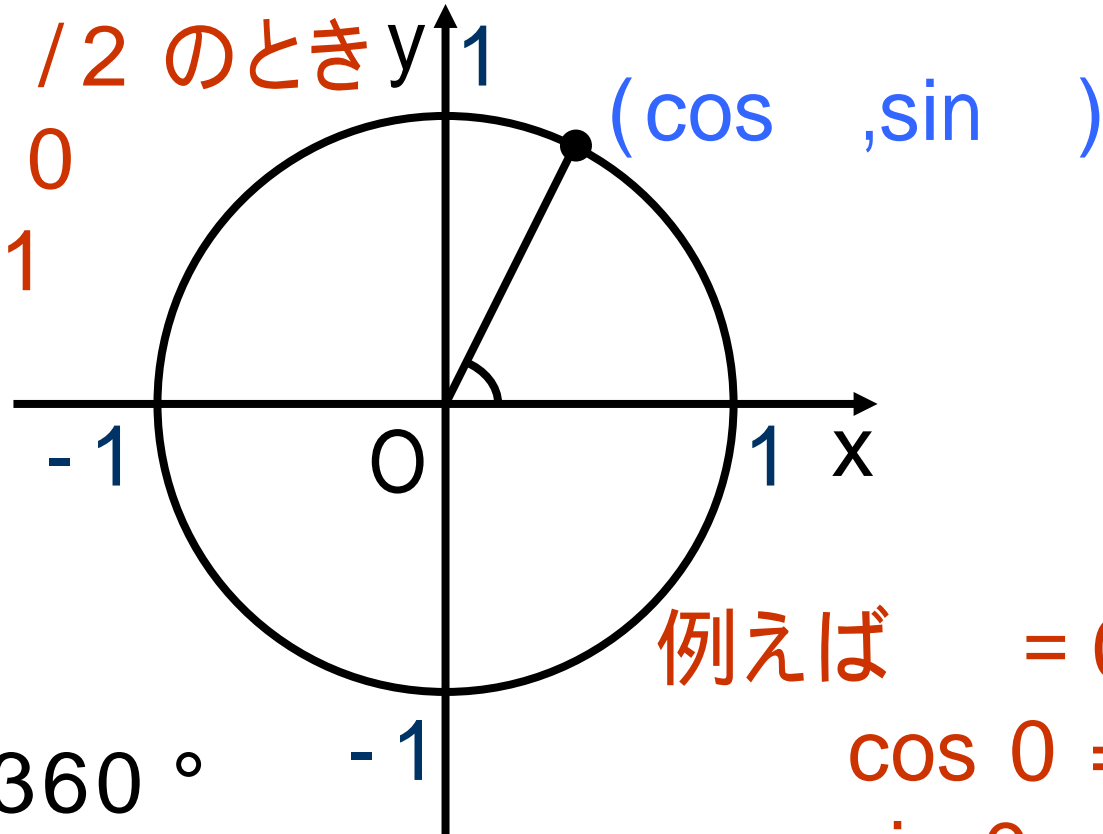
単位円上の点の座標によって定義する

例えば

$\theta = \pi/2$ のとき

$$\cos \theta = 0$$

$$\sin \theta = 1$$



弧度法

$$2\pi = 360^\circ$$

例えば $\theta = 0$ のとき

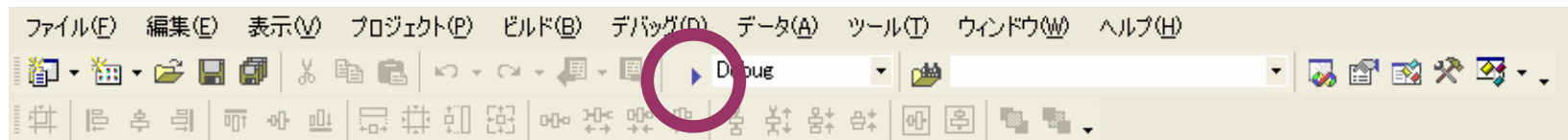
$$\cos 0 = 1$$

$$\sin 0 = 0$$

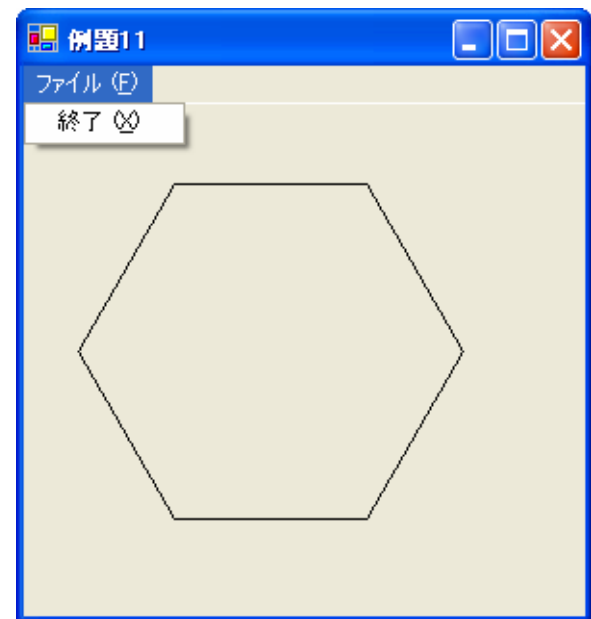
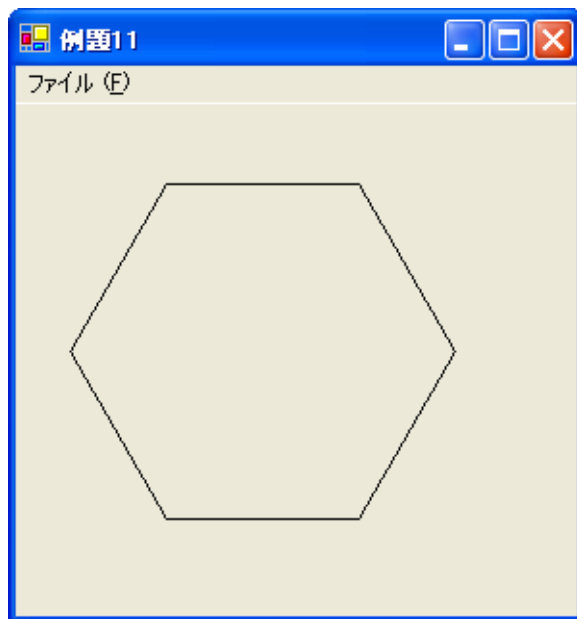
Math クラス

- 数学計算の機能を提供する
 - Math.Sin(*double*)
引数の sin を計算するメソッド
 - Math.Cos(*double*)
引数の cos を計算するメソッド
 - Math.PI
円周率 の値

実行 (デバッグ)

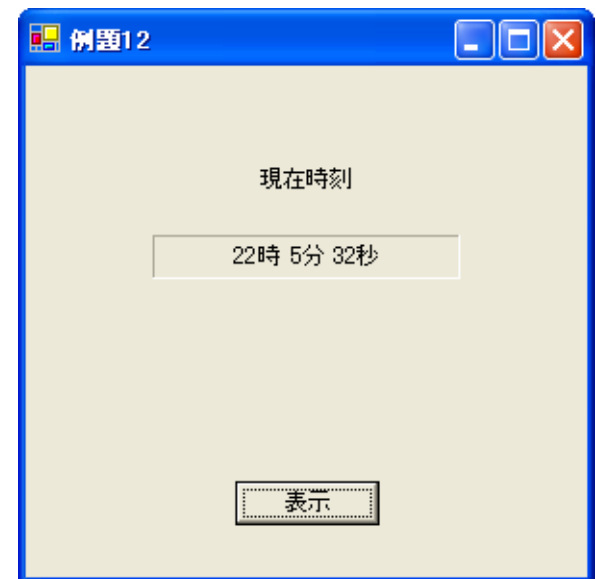


- [デバッグ] [開始]でプログラムを実行する
- ツールバーの**開始ボタン**をクリックしてもよい



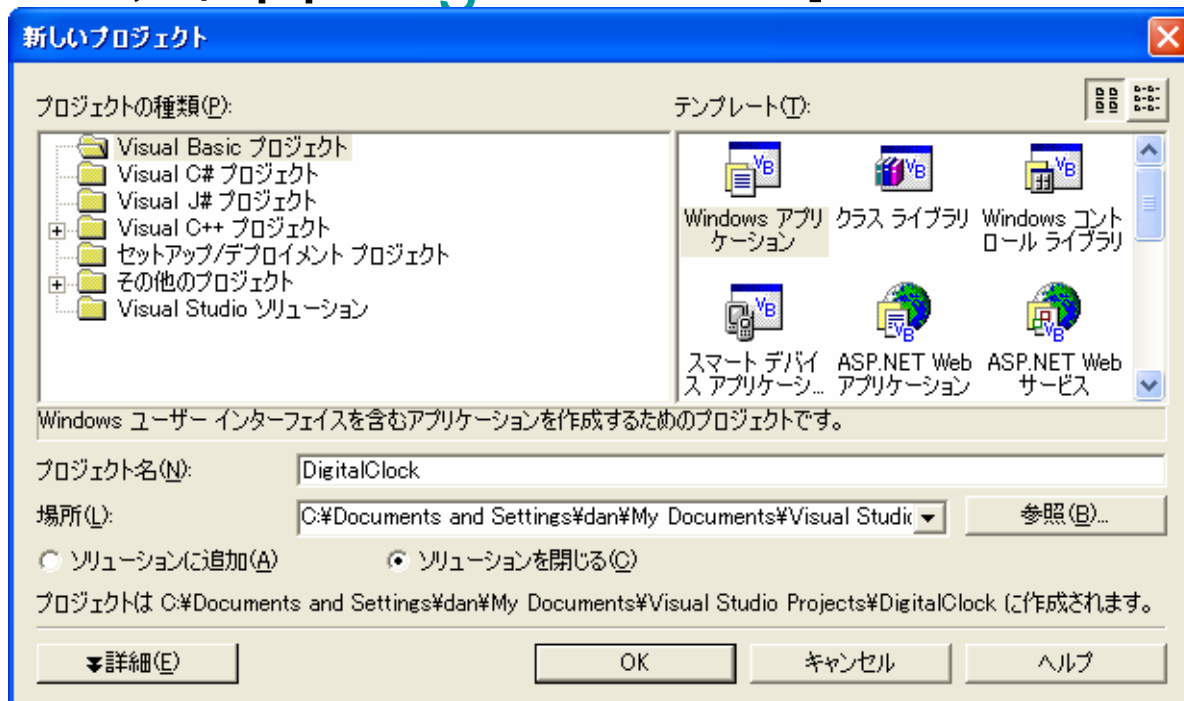
例題12

- 現在の時刻を表示するプログラム
- 手順：
 - プロジェクトの新規作成
(DigitalClock)
 - 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「DigitalClock」



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

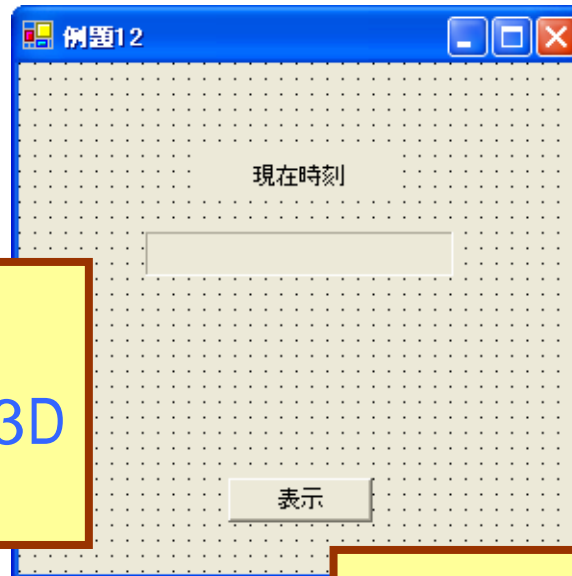
Label1

.Text: "現在時刻"

Label2

.BorderStyle: Fixed3D

.Text: ""



Button1

.Text = "表示"

Click イベント コード記述へ

コードの記述

- [表示] ボタンを押すと現在の時刻を表示する
 - DateTime 構造体を使って現在時刻を取得する
- `t = DateTime.Now`

```
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 t As DateTime
        Dim h, m, s As Integer

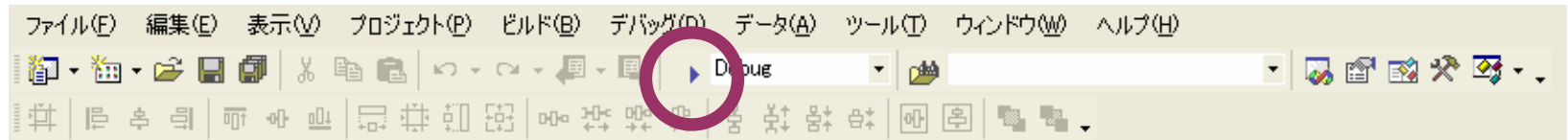
        '現在時刻を取得する
        t = DateTime.Now

        h = t.Hour      '何時?
        m = t.Minute    '何分?
        s = t.Second    '何秒?

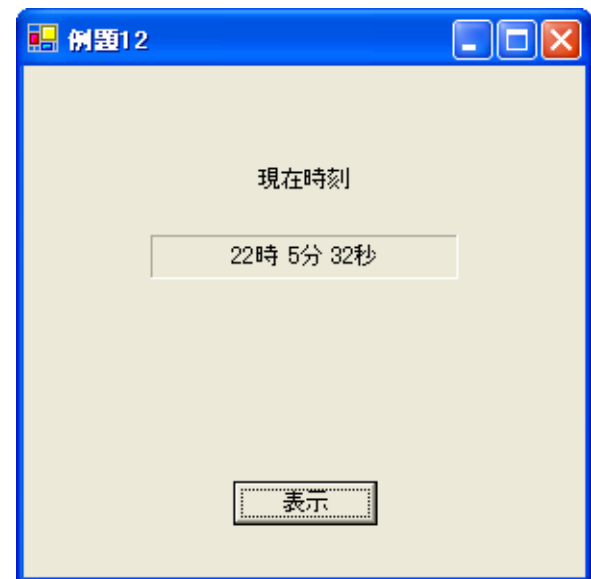
        '取得した時刻を表示する
        Label2.Text = h.ToString() & "時 " & m.ToString() & "分 " & s.ToString() & "秒"

    End Sub
End Class
```

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

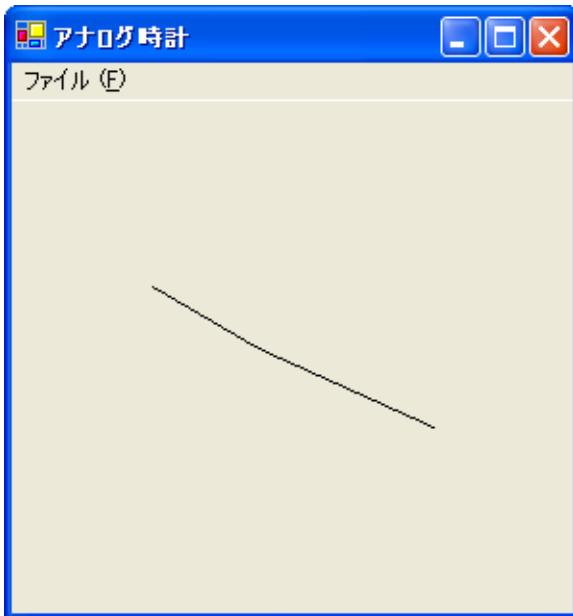


課題

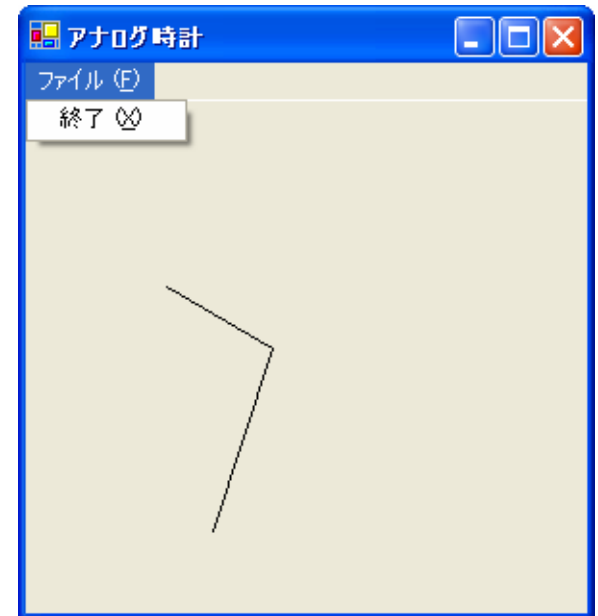
アナログ時計 (ex_20061114)

- アナログ時計の時針および分針を実装せよ。

10:20



10:33



まとめ

- 三角関数を利用して任意の正多角形を描画する処理について理解した。また、現在時刻を取得する方法を学んだ。
 - Paint イベント
 - Math クラス
 - DateTime 構造体

次回予定

- グラフィックス (4)
 - ペンとブラシ
 - Graphics クラスのメソッド