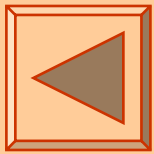


松山大学 経営学部

情報処理論（応用）



第8回 テキストエディタ（1）



講師 檀 裕也

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

2006年10月20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。

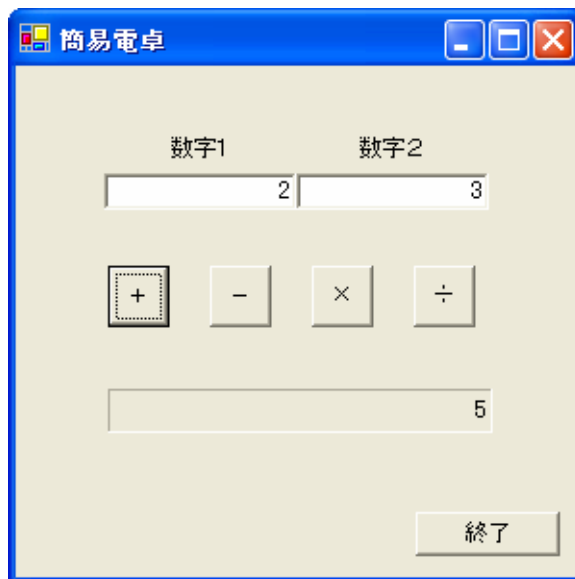
出席確認フォーム

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

前回の課題

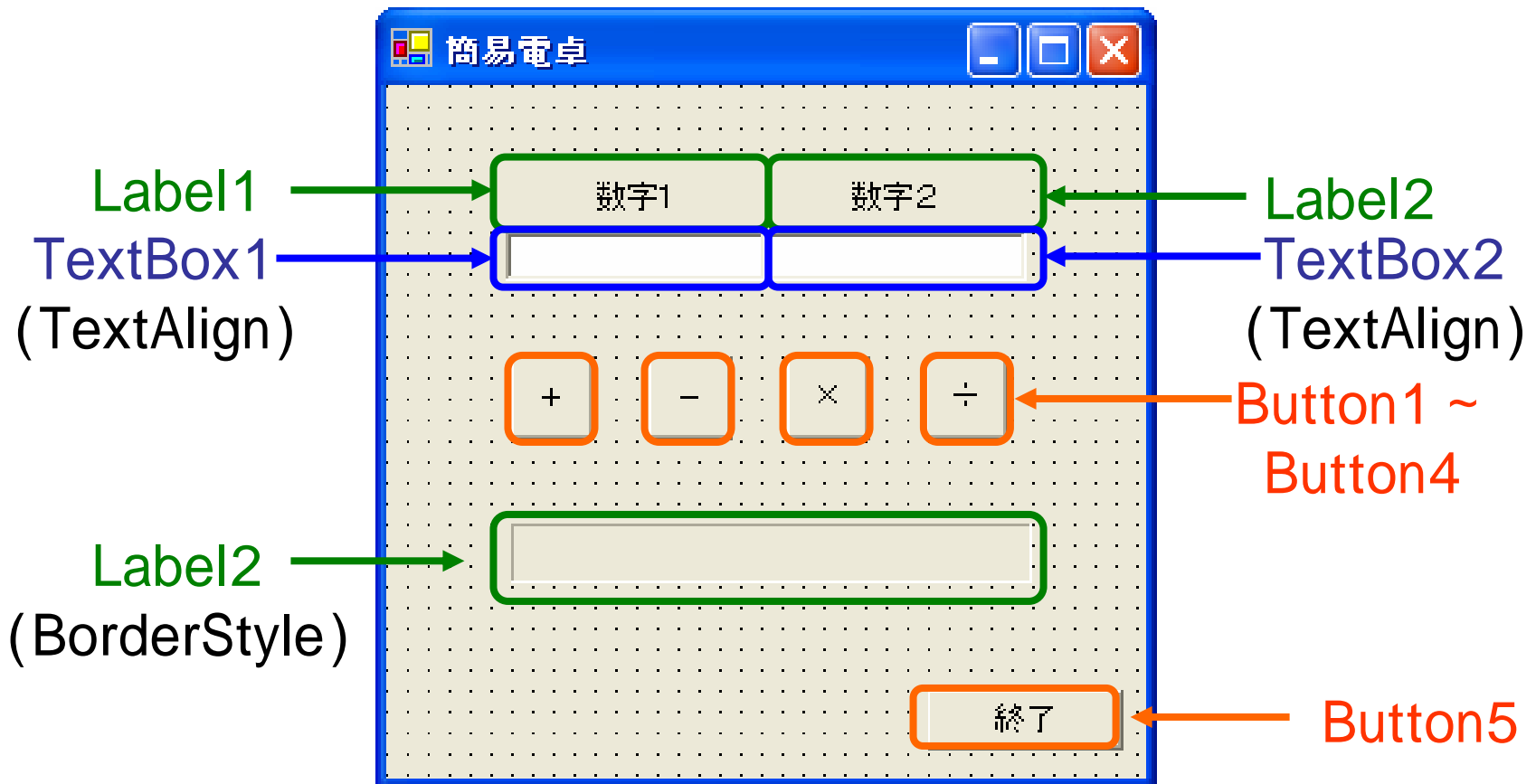
簡易電卓プログラム (ex_20061017)

- 入力された2つの数字を使って、和・差・積・商を計算する。ただし、例外処理を記述せよ。



解答例 (フォーム)

ラベル・ボタン・テキストボックスを配置する



解答例 (コード)

- 例外処理の記述

- テキストボックスから入力された文字列を整数に変換する CInt 関数
- 整数を文字列に変換する ToString メソッド

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim a, b, c As Integer
    Dim str As String

    Try
        a = CInt(TextBox1.Text)
        b = CInt(TextBox2.Text)
        c = a + b
        str = c.ToString()
    Catch ex As Exception
        str = "正しく数字を入力してください。"
    End Try

    Label3.Text = str
End Sub
```



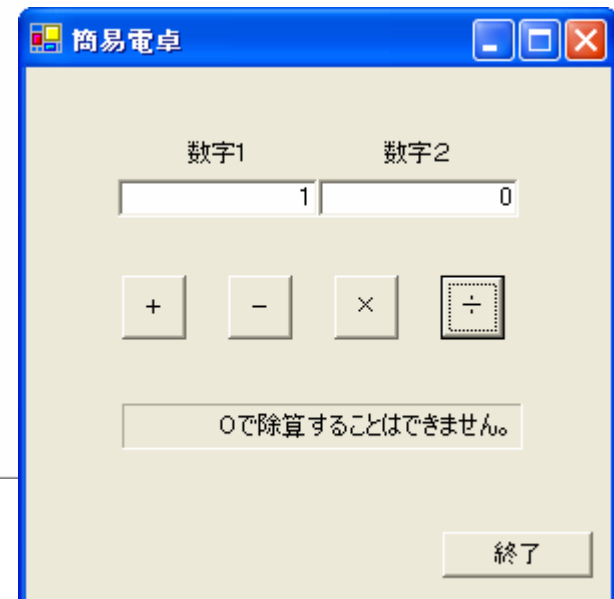
解答例（コード）追加

- 0 による除算の場合
 - 割る数が0 のとき、除算できないと表示する
 - 割る数が0 でないとき、除算の結果を表示する

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
    Dim a, b, c As Integer
    Dim str As String

    Try
        a = CInt(TextBox1.Text)
        b = CInt(TextBox2.Text)
        If b = 0 Then
            str = "0で除算することはできません。"
        Else
            c = a / b
            str = c.ToString()
        End If
    Catch ex As Exception
        str = "正しく数字を入力してください。"
    End Try

    Label3.Text = str
End Sub
```



暗黙的な型変換の失敗例

- 文字列同士での + 演算子は、連結演算子 & と同じ機能になる

Label3.Text = TextBox1.Text + TextBox2.Text

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Try
        Label3.Text = TextBox1.Text + TextBox2.Text
    Catch ex As Exception
        Label3.Text = "正しく数字を入力してください。"
    End Try
End Sub
```

1 + 2 = 1 2 ?



今回の予定

- テキストエディタの作成
 - テキストボックスの拡張機能
 - メニュー
 - ファイルを開く
 - ファイルに保存する
- 到達目標
 - テキストエディタの基本機能を実装し、コンポーネントを使った処理について理解する

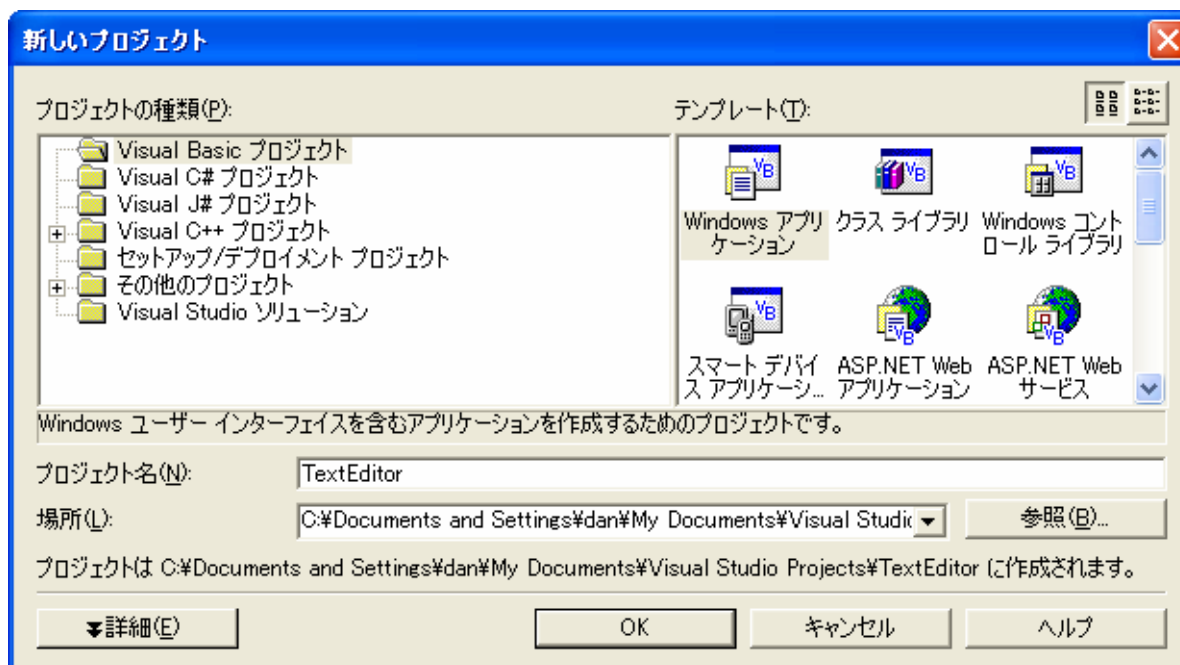
サンプルプログラム 1

テキストエディタ

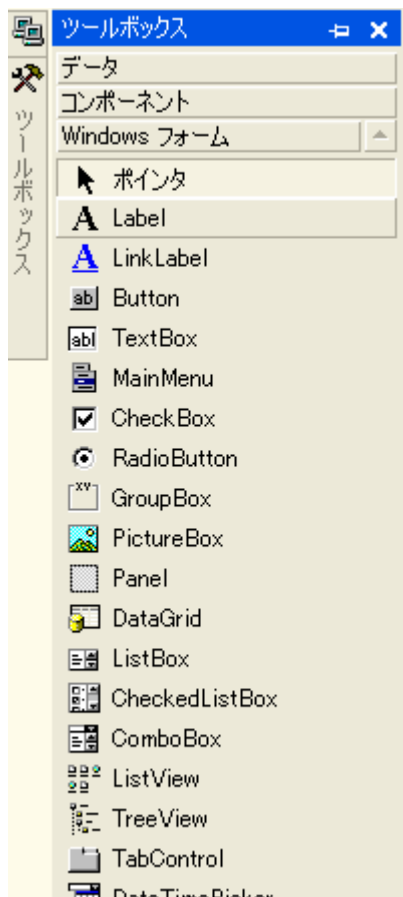
- 手順：
 - プロジェクトの新規作成 (**TextEditor**)
 - 画面レイアウト (テキストボックス・メニューの配置)
 - イベントハンドラの記述
 - 実行および動作確認

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「TextEditor」



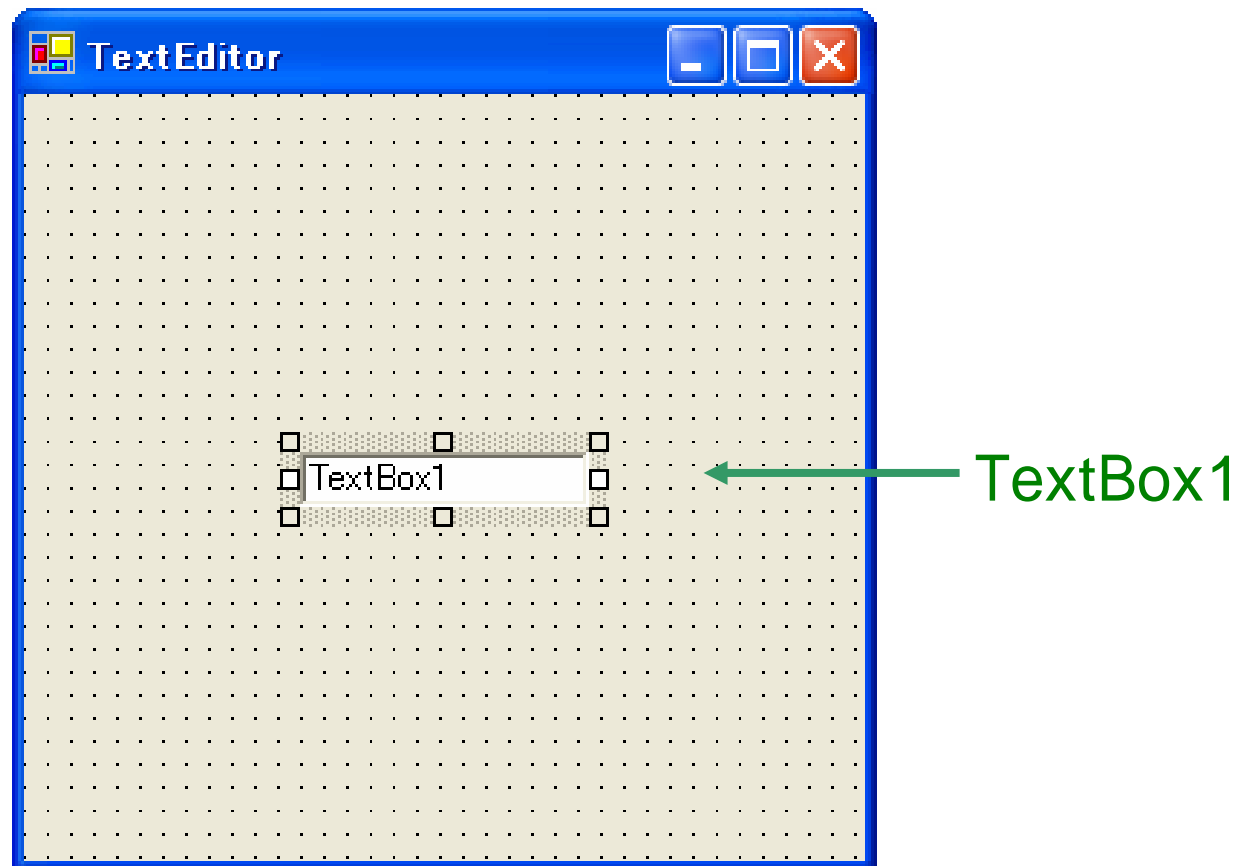
画面の構成要素



- 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

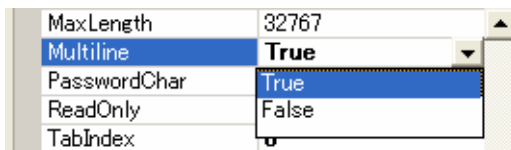
任意の位置にテキストボックスを配置する



プロパティの設定

- テキストボックスの (Name) プロパティの値を `EditTextBox` に変更する

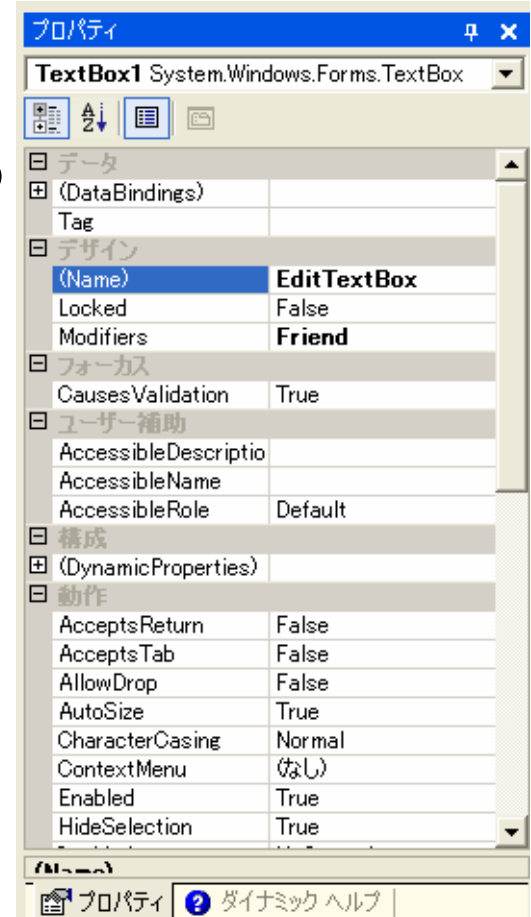
- Multiline の値を `True` に変更する



- Dock の値を `Fill` に変更する

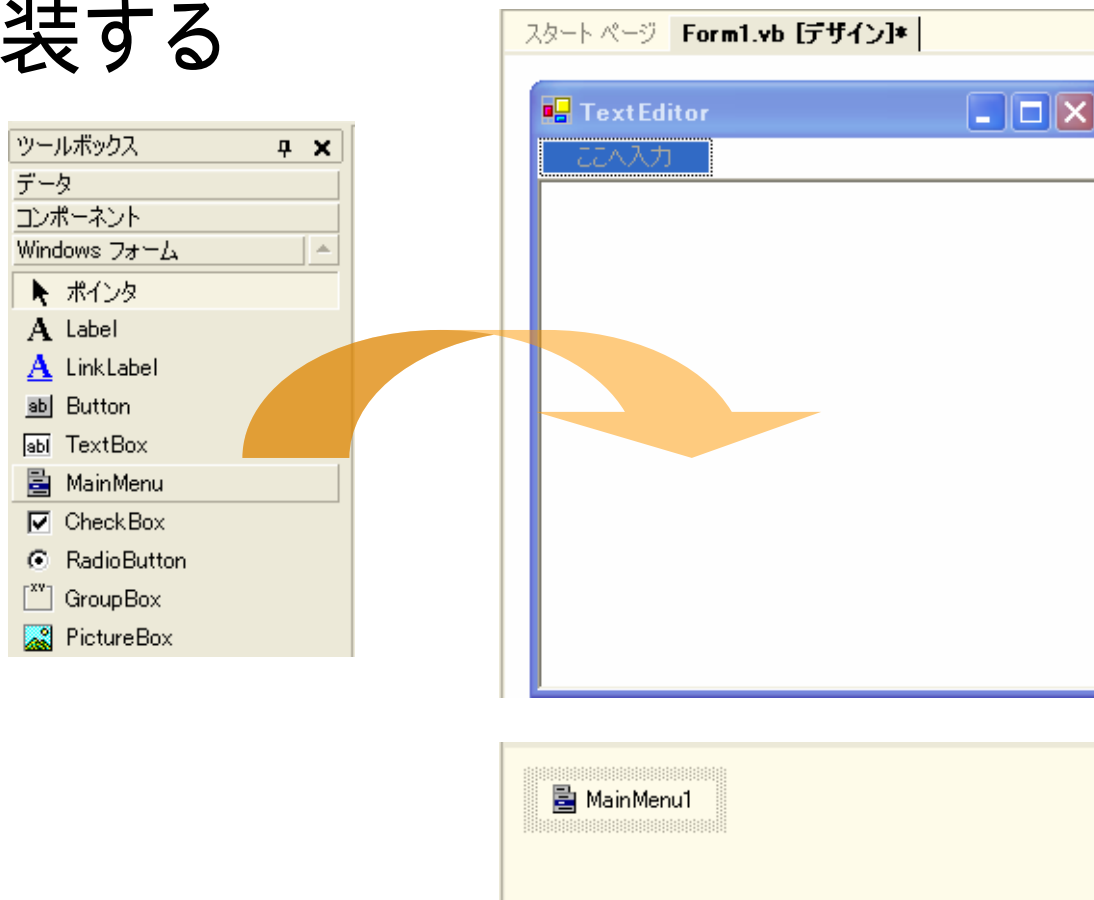


- Text の値を空にする



メニューをつける

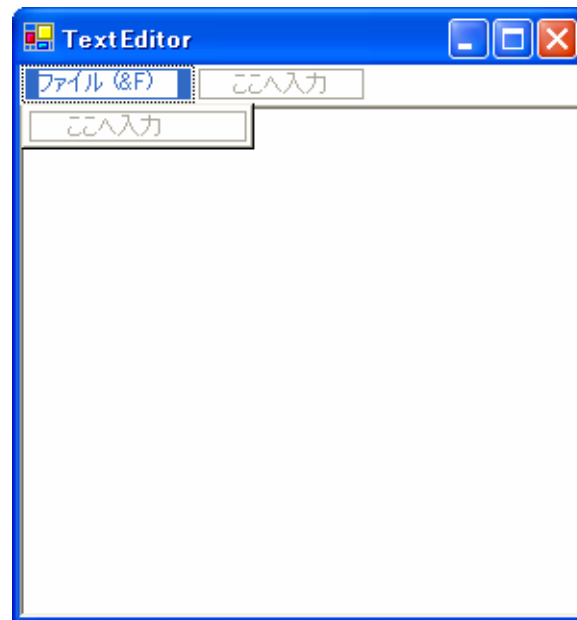
- MainMenu コンポーネントを使ってメニューを実装する



メニューの設定

- メニューの [ここへ入力] のところに
ファイル (&F)

と入力する

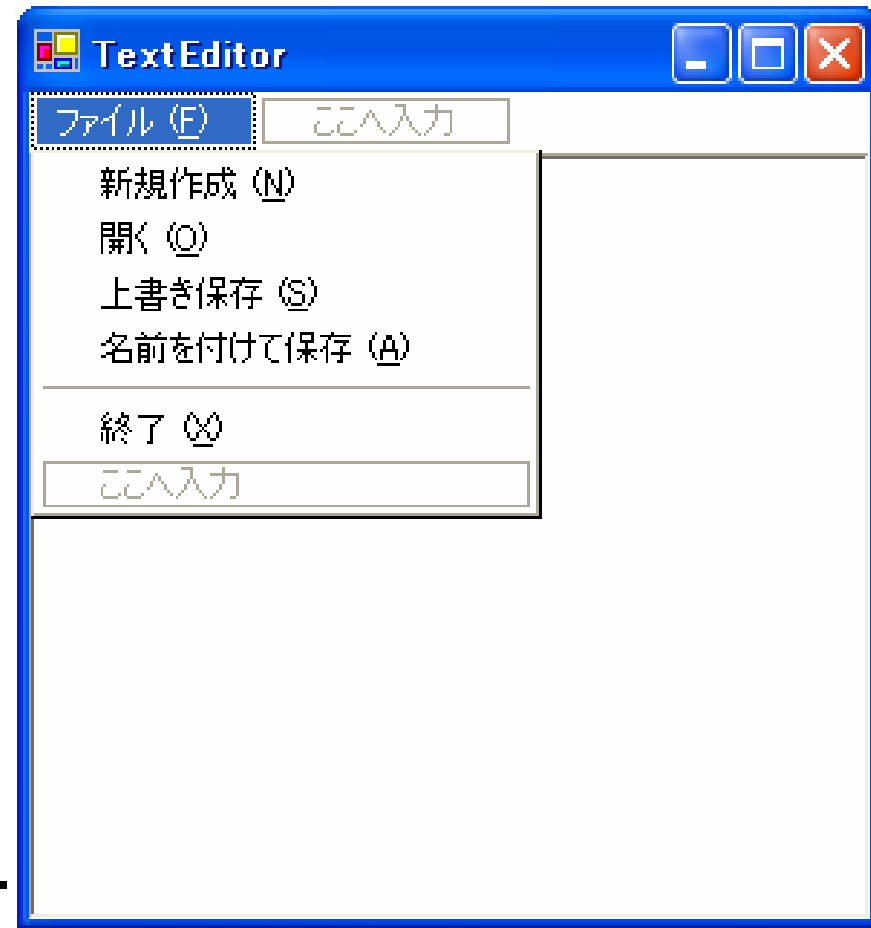


&F は [F] がアクセスキーになることを示す

メニュー項目の設定

- ファイルメニューの項目
 - 新規作成 (&N)
 - 開く (&O)
 - 上書き保存 (&S)
 - 名前を付けて保存 (&A)
 - - 区切り線
 - 終了 (&X)

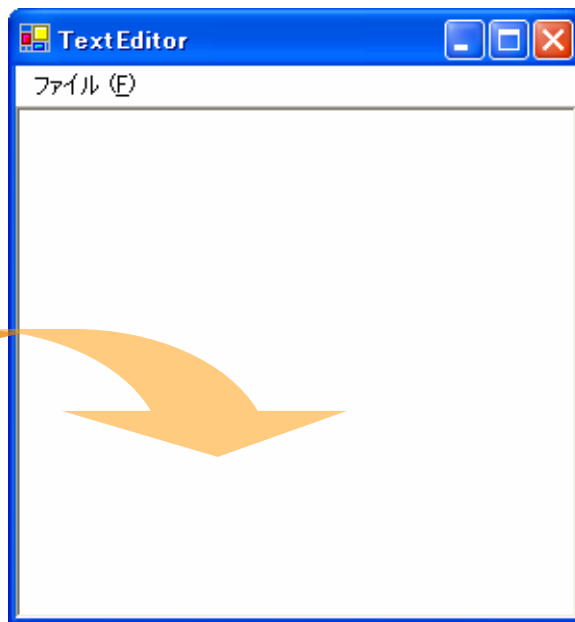
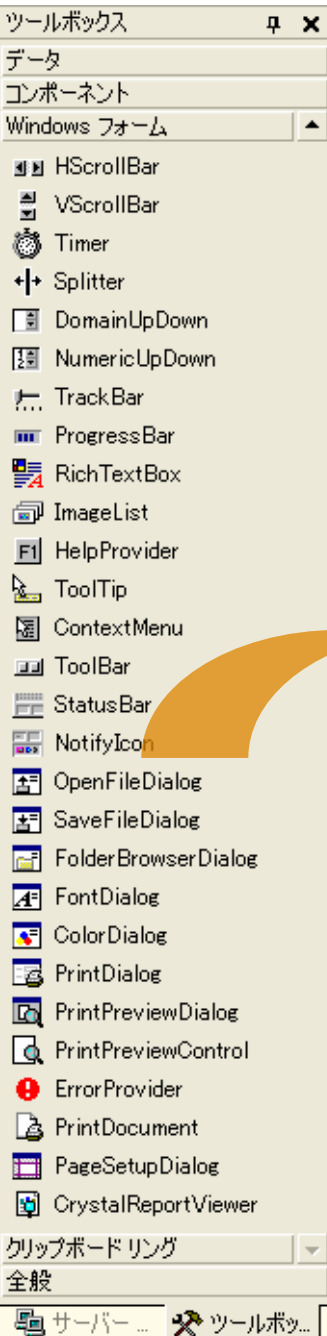
区切り線のところは「-」を入力またはコンテキストメニューから区切り線を挿入する



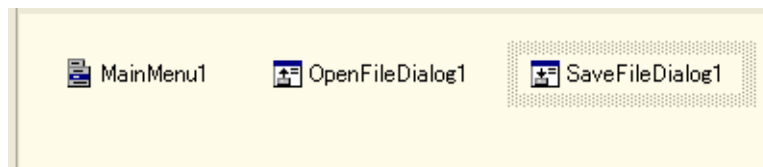
オブジェクトの名前と設定値

オブジェクト名	Text プロパティ
MenuItem1	ファイル (&F)
Menu_New	新規作成 (&N)
Menu_OpenFile	開く (&O)
Menu_SaveFile	上書き保存 (&S)
Menu_SaveAs	名前を付けて保存 (&A)
MenuItem6	-
Menu_Exit	終了 (&X)

ファイルを開く / 保存する



- 次のコンポーネントを配置する
 - OpenFileDialog
 - SaveFileDialog



コードの記述（新規作成）

テキストボックスの値を空にする

```
EditTextBox.Text = ""
```

Windows フォーム デザイナで生成されたコード

```
Private Sub Menu_New_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Menu_New.Click  
    EditTextBox.Text = ""  
End Sub
```

コードの記述（開く）

テキスト文書を開き、画面に表示する

```
Private Sub Menu_OpenFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Menu_OpenFile.Click
    OpenFileDialog1.Filter = "テキスト文書 (*.txt)|*.txt|すべてのファイル|*.*"
    OpenFileDialog1.RestoreDirectory = True
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
        Dim sr As New System.IO.StreamReader(OpenFileDialog1.FileName)
        EditTextBox.Text = sr.ReadToEnd()
        sr.Close()
    End If
End Sub
```

- OpenFileDialog.Filter
 - フィルタ条件（拡張子など）
- OpenFileDialog.ShowDialog()
 - 「ファイルを開く」ダイアログを表示し、ユーザからファイルの場所および名前を入力・選択させる

コードの記述（上書き保存）

編集したファイルを記憶媒体に保存する

```
Private Sub Menu_SaveFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Menu_SaveFile.Click
    If SaveFileDialog1.FileName <> "" Then
        Dim sr As New System.IO.StreamWriter(SaveFileDialog1.FileName)
        sr.Write(EditTextBox.Text)
        sr.Close()
    Else
        Menu_SaveAs_Click(sender, e)
    End If
End Sub
```

- SaveFileDialog.FileName
 - 以前保存したファイルの名前
- sr.Write(EditTextBox)
 - テキストボックスの内容をファイルに書き出す
- Menu_SaveAs_Click(sender, e)
 - 名前をつけて保存する

コードの記述（名前を付けて保存）

名前を付けてファイルを保存する

```
Private Sub Menu_SaveAs_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Menu_SaveAs.Click
    SaveFileDialog1.Filter = "テキスト文書 (*.txt)|*.txt|すべてのファイル|*.*"
    SaveFileDialog1.RestoreDirectory = True
    If SaveFileDialog1.ShowDialog() = DialogResult.OK Then
        Dim sr As New System.IO.StreamWriter(SaveFileDialog1.FileName)
        sr.Write(EditTextBox.Text)
        sr.Close()
    End If
End Sub
```

- SaveFileDialog1.RestoreDirectory
 - ファイルの保存場所を記憶する
- SaveFileDialog1.ShowDialog()
 - 「ファイルを保存する」ダイアログを表示し、ファイルの保存場所および名前を入力させる

コードの記述 (終了)

プログラムの実行を終了する

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

まとめ

- テキストエディタを作成し、メニューの設定、ファイルの選択といった処理について学んだ。
 - MainMenu コンポーネント
 - OpenFileDialog コンポーネント
 - SaveFileDialog コンポーネント

次回予定

- テキストエディタの作成（２）
 - 今回の続き...