

松山大学 経営学部

# 情報処理論（応用）



## 第9回 テキストエディタ（2）



講師 檀 裕也

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

2006年10月24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>

# 今回の予定

- テキストエディタの作成（前回の続き）
  - 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



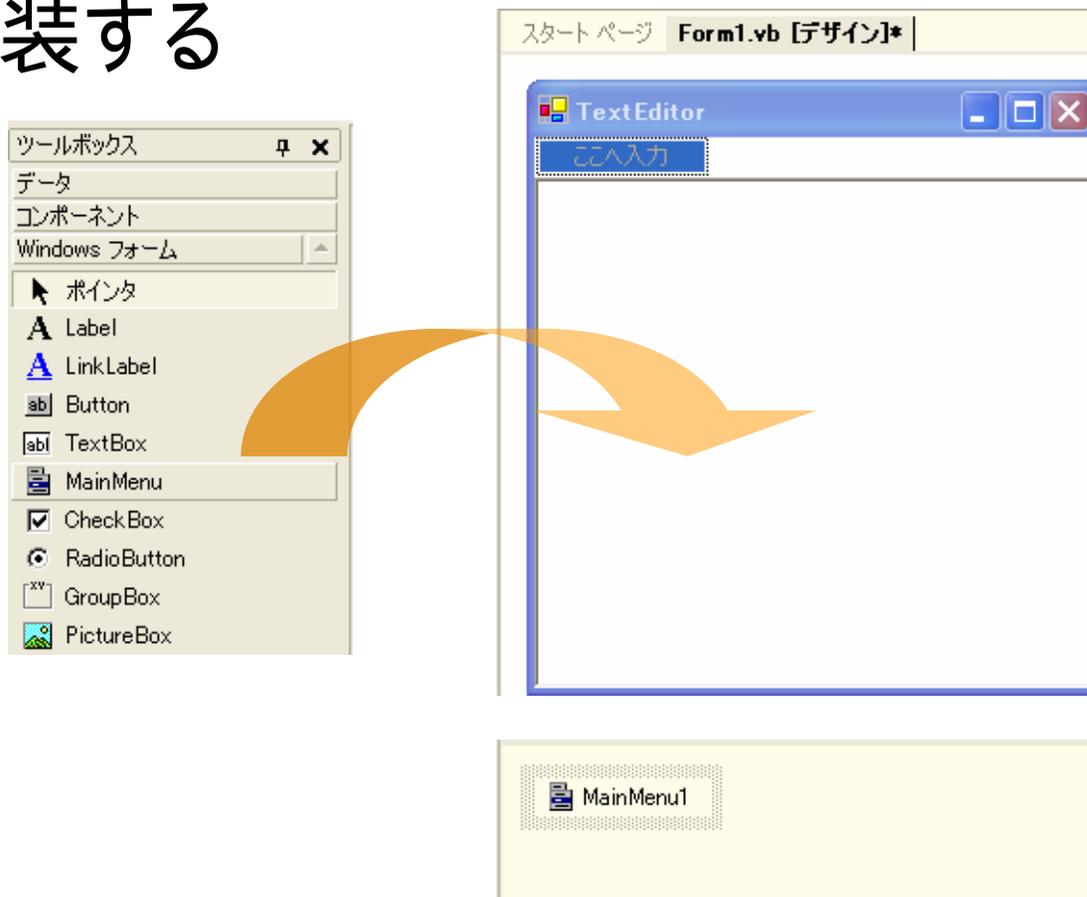
```
TextEditor
ファイル (F)
#include <stdio.h>

int main()
{
    printf("Hello, world!");
    return 0;
}
```

- 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

# メニューをつける

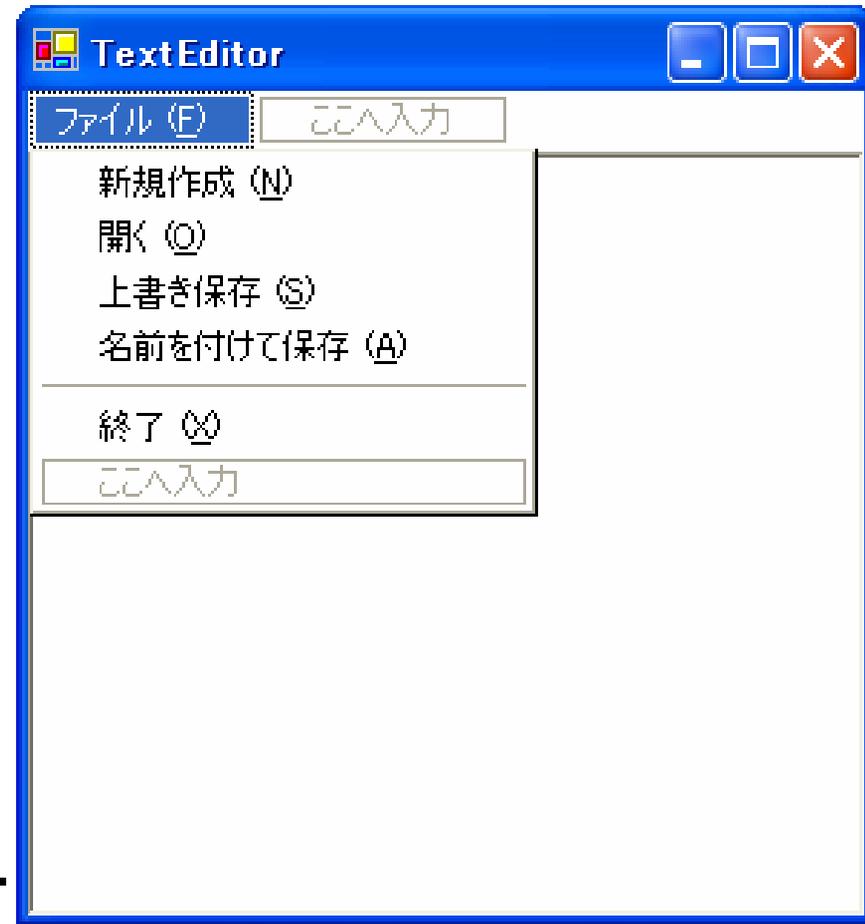
- MainMenu コンポーネントを使ってメニューを実装する



# 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)

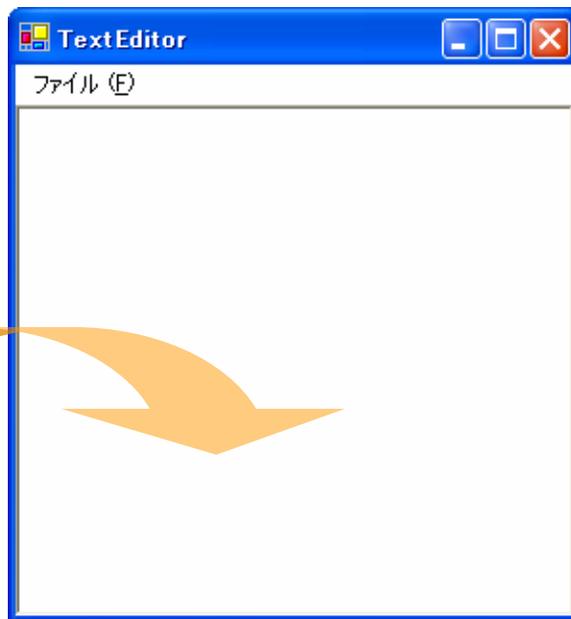
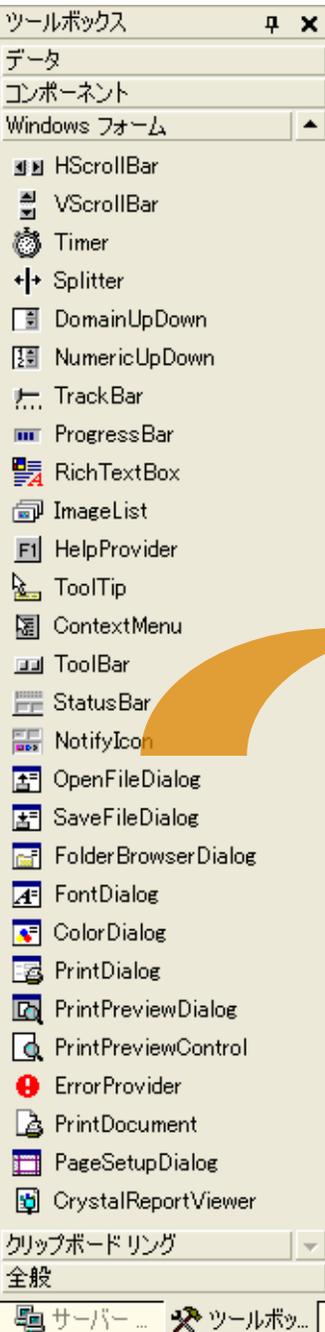
区切り線のところは「-」を入力またはコンテキストメニューから区切り線を挿入する



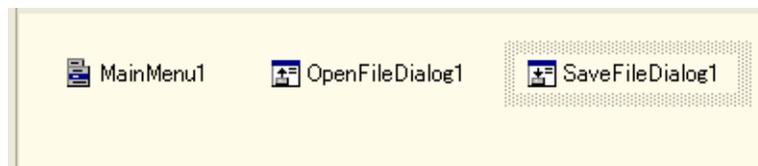
# オブジェクトの名前と設定値

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



# 「新規作成」のコード

テキストボックスの値を空にする

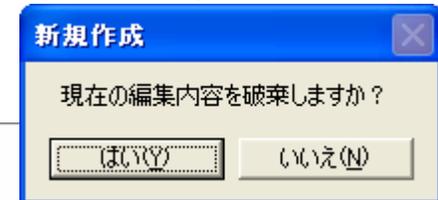
```
EditTextBox.Text = ""
```

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
```

# 「新規作成」コードの改良

- いきなりコマンドを実行するのではなく、ユーザーの確認を得てから新規作成する



Windows フォーム デザイナで生成されたコード

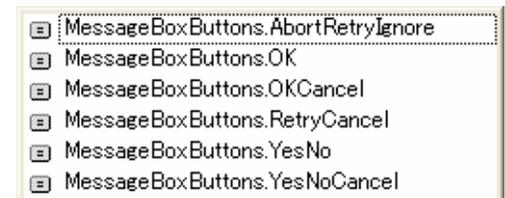
```
Private Sub Menu_New_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Menu_New.Click
    Dim n As Integer

    n = MessageBox.Show("現在の編集内容を破棄しますか?", "新規作成", MessageBoxButtons.YesNo)
    If n = DialogResult.Yes Then
        EditTextBox.Text = ""
    End If
End Sub
```

– MessageBox.Show(表示列, タイトル, ボタン)

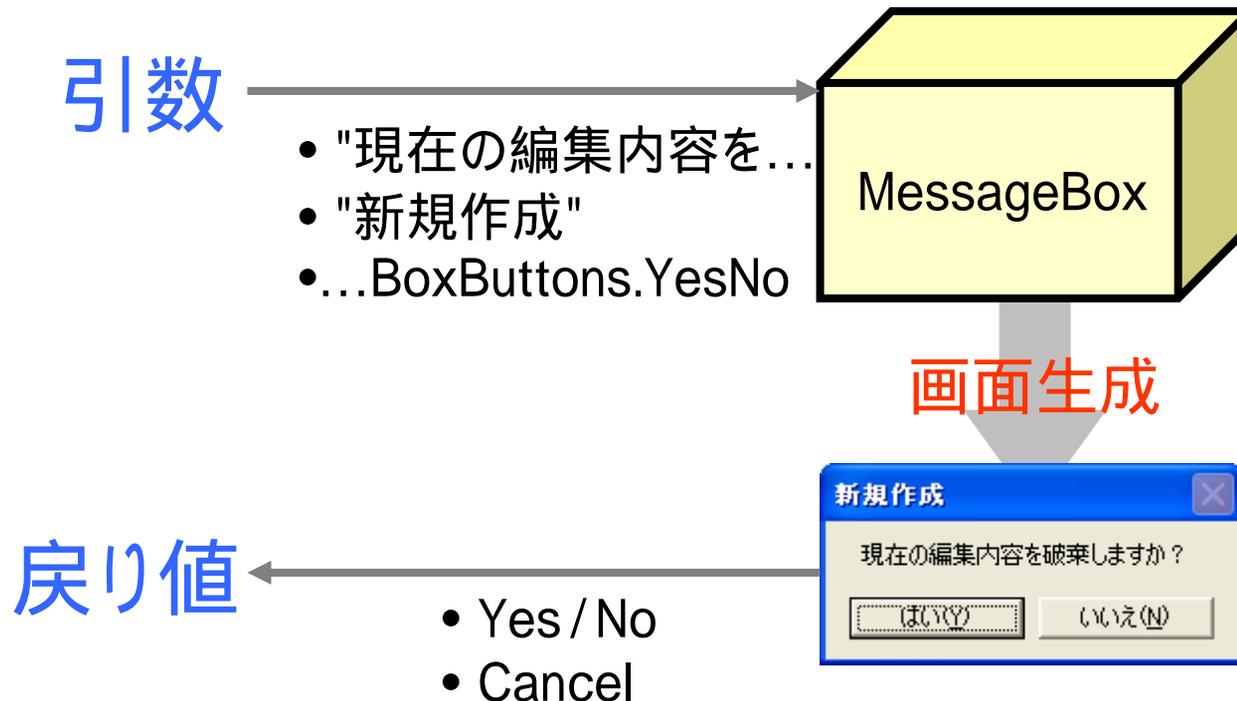
– MessageBoxButtons クラス

- OKCancel [OK] および [Cancel]
- YesNo [Yes] および [No]



# 引数と戻り値

- 一般に、**引数**を与えて機能を実行し、その結果を**戻り値**として返す



# 「開く」のコード (1)

## 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 プロパティ
  - OpenFileDialog.Filter = "説明|フィルタ条件"
  - フィルタ条件に拡張子 (\*.txt) などを指定する
  - 複数の説明およびフィルタ条件を指定可能
  - 複数のフィルタ条件は;で区切って指定する

使用例: "JPEG画像|\*.jpeg; \*.jpg"

# 「開く」のコード (2)

## 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.RestoreDirectory** プロパティ
  - **OpenFileDialog.Filter = True**  
次にファイルを開くときに備え、現在のファイルの場所（フォルダ/ディレクトリ）を覚えておく
  - **OpenFileDialog.Filter = False**  
現在のファイルの場所を記憶しない

# 「開く」のコード (3)

## テキスト文書を開き、画面に表示する

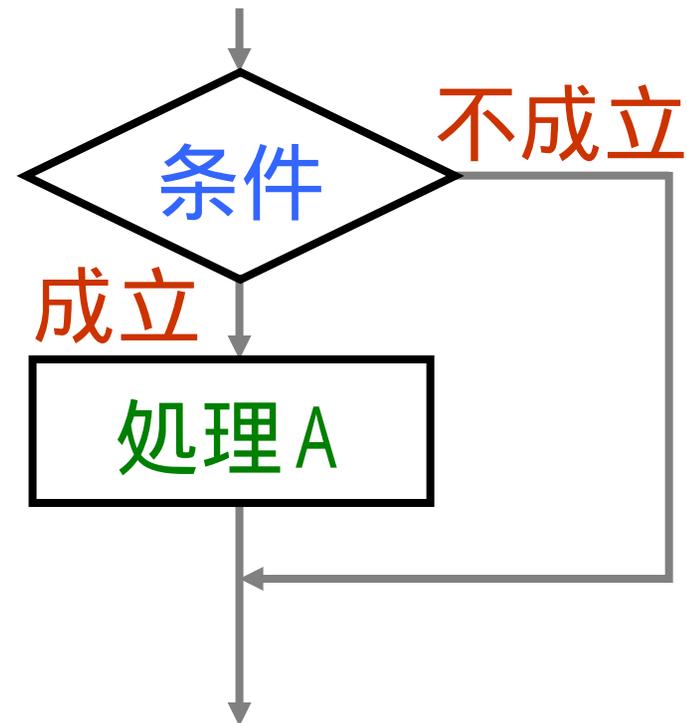
```
Private Sub Menu_OpenFile_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Menu_OpenFile.Click
    OpenFileDialog1.Filter = "テキスト文書 (*.txt)|*.txt|すべてのファイル|*.*"
    OpenFileDialog1.RestoreDirectory = True
    If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
        Dim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

- **OpenFileDialog.ShowDialog()** メソッド
  - 「ファイルを開く」ダイアログを表示し、ユーザにファイルを選択させる
  - 正常終了すれば **DialogResult.OK** を返し、**OpenFileDialog.FileName** にファイル名を与える
  - キャンセルされれば **DialogResult.Cancel** を返す

# If ステートメント (Elseなし)

- 条件が成立するとき処理 A を実行する。成立しないときは何も実行しない。

```
If 条件 Then  
    処理 A  
End If
```



# 「開く」のコード (4)

## ファイルの入出力をストリームで実現する

```
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
```

- **System.IO.StreamReader** クラス
  - System.IO.StreamReader(**ファイル名**)  
指定したファイルのストリームリーダーを生成する
  - ... (**ファイル名**, System.Text.Encoding.Default)  
文字のエンコードを Shift-JIS に設定する

# 「開く」のコード (5)

## ファイルの入出力をストリームで実現する

```
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
```

- **System.IO.StreamReader** クラス
  - sr.ReadToEnd() メソッド  
ファイルの先頭から末尾までのデータを読み込む
  - sr.Close() メソッド  
ストリームリーダーを閉じる

# 「名前を付けて保存」のコード

## 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 sw As New System.IO.StreamWriter(SaveFileDialog1.FileName)
        sw.Write(EditTextBox.Text)
        sw.Close()
    End If
End Sub
```

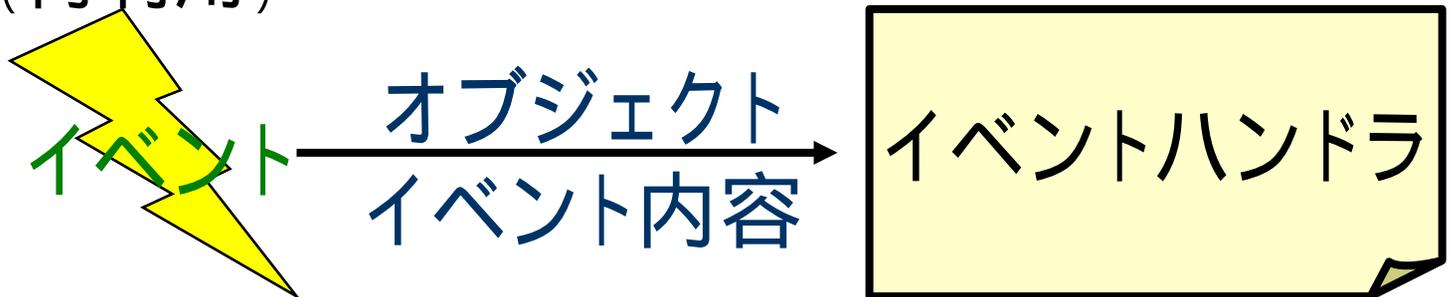
- **SaveFileDialog.Show()**
  - 「名前を付けて保存する」ダイアログを表示し、ファイルの保存場所および名前を入力させる
- **sr.Write(文字列)**
  - 文字列をファイルに書き込む

# 「上書き保存」のコード

## 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 sw As New System.IO.StreamWriter(SaveFileDialog1.FileName)
        sw.Write(EditTextBox.Text)
        sw.Close()
    Else
        Menu_SaveAs_Click(sender, e)
    End If
End Sub
```

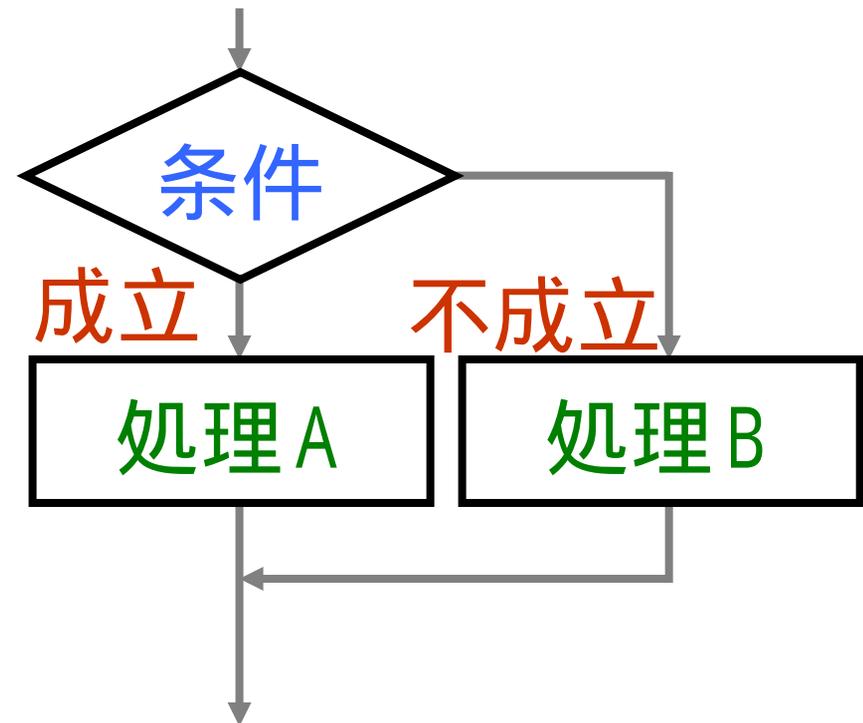
- Menu\_SaveAs\_Click(sender, e)
  - 「名前を付けて保存」のイベントハンドラを呼び出す（再利用）



# If ステートメント

- 条件が成立するとき処理 A を実行し、成立しないとき処理 B を実行する

```
If 条件 Then
    処理 A
Else
    処理 B
End If
```



# 「終了」のコード

プログラムの実行を終了する

Me.Dispose()

```
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する過程で、コンポーネントの利用やファイルの入出力について学んだ。
  - ストリームによるファイルの入出力

# 次回予定

- 乱数
  - コンピュータ上で偶然性を実現するには？