

【T2】 Delphi チュートリアルセッション

「『Delphiはじめて奮戦記』で学ぶ Delphiチュートリアル」

株式会社フルネス
コーチング事業部 マネージャー
田原 孝

アジェンダ

- 株式会社フルネスについて
- Delphiプログラミングの基本
- 演習: 計算機のテンキーを作る
- 演習: 計算機の四則演算ボタンを作る
- 練習問題
- まとめ

株式会社フルネスについて

- 事業内容
 - ハンズオン教育サービス
 - コーチングサービス
 - スキル診断サービス
 - システム開発サービス
- 教育セミナーのコースは、オブジェクト指向、UML、Delphi/Java/.NETなど
- 詳しくは、<http://www.fullness.co.jp/>をご覧ください。

チュートリアルの目的と内容

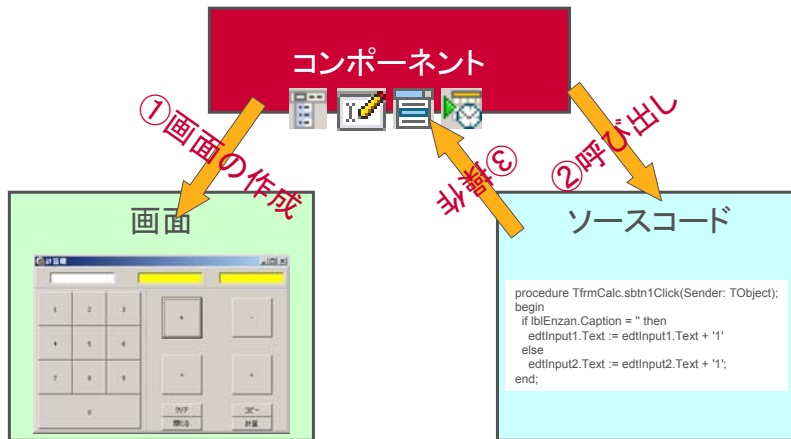
- 目的
 - Delphiのプログラミングを体験！
- 内容
 - 簡単な計算機(電卓)を作ってみる
- 演習の順序
 - テンキーを作る
 - 休憩
 - 四則演算ボタンを作る
 - 練習問題「計算結果が3の倍数なら赤になる」

演習がうまくいかないときは。。

- 手をあげてください！
 - アシスタントがフォローさせていただきます

Delphiプログラミングの基本

プログラミングの基本は「コンポーネント」



Delphiプログラミング手順

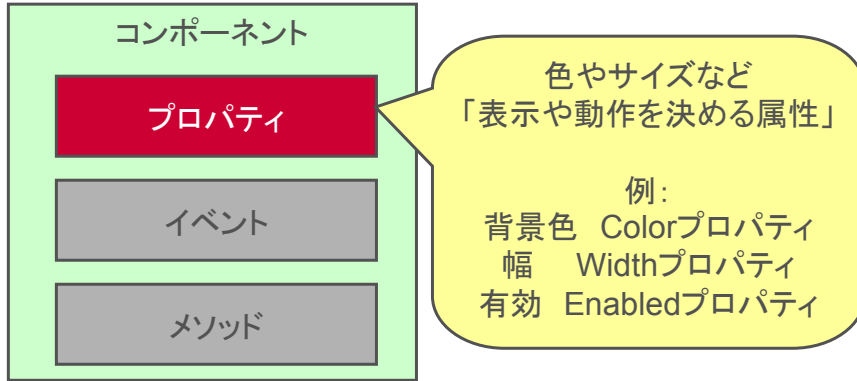
1. プロジェクトを作成
2. フォームに「コンポーネント」を配置
3. 「コンポーネント」のプロパティを設定
4. 「コンポーネント」のイベントハンドラにソースコードを記述
5. 実行して動作を確認
6. 完成！

演習： 計算機のテンキーを作る

学習内容

- コンポーネントで画面を作る方法
- コンポーネントのプロパティ
- コンポーネントのイベントとイベントハンドラ
- オブジェクトインスペクタの使い方

コンポーネントのプロパティとは



オブジェクトインスペクタでプロパティの値を設定

値を設定する
コンポーネント名

プロパティ名

プロパティの値

オブジェクトインスペクタ

frmCalc TfrmCalc

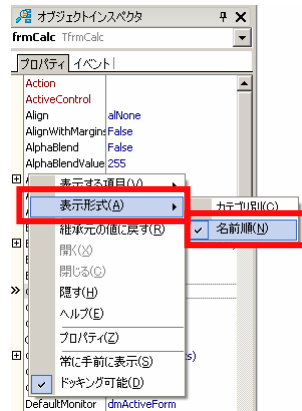
プロパティ | イベント

BorderIcons	bdLeftToRight
BorderStyle	[bsSystemMenu, biMinimize, biMaximize]
BorderWidth	0
Caption	引数指定
ClientHeight	308
ClientWidth	472
Color	[clBtnFace]
Constraints	(TSizeConstraints)
Ctl3D	True
Cursor	crDefault
DefaultMonitor	dniActiveForm
DockSite	False
DragKind	dkDrag
DragMode	dniManual

すべての項目が表示されています

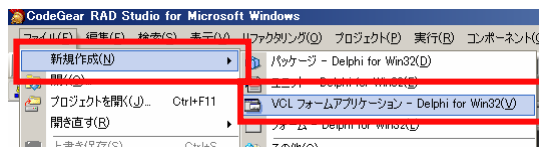
オブジェクトインスペクタの表示形式

- [表示形式 | 名前順]ポップアップメニューを選択



[演習] プロジェクトの作成

- [ファイル | 新規作成 | VCLフォームアプリケーション]メニューを選択



[演習] 計算機フォームのプロパティを設定

• オブジェクトインスペクタで以下のプロパティを設定してください

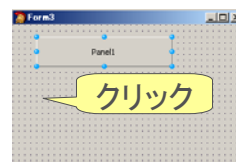
プロパティ名	設定する値
Caption	計算機
Height	380
Width	500

フォームにコンポーネントを配置する方法

1. ツールパレットのコンポーネントをクリック




2. フォーム上の配置したい場所をクリック



[演習] TPanelコンポーネントを配置

- 以下のコンポーネントを配置してください


カテゴリ	コンポーネント	配置先
Standard	TPanel 	フォーム

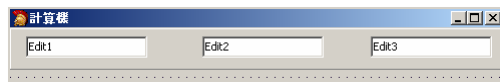
- 以下のプロパティを設定してください

コンポーネント名	プロパティ名	設定する値
Panel1	Align	alTop
	Caption	(空白)
	Enabled	False

[演習] TEditコンポーネントを配置

- 以下のコンポーネントを配置してください


カテゴリ	コンポーネント	個数	配置先
Standard	TEdit 	3個	Panel 1



[演習] TEditコンポーネントを配置(つづき)

コンポーネント名	プロパティ名	設定する値
Edit1	Color	clWhite
	Name	edtInput 1
	Text	(空白)
Edit2	Color	clYellow
	Name	edtInput2
	Text	(空白)
Edit3	Color	clYellow
	Name	edtKekka
	Text	(空白)

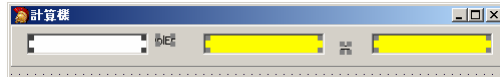
[演習] TLabelコンポーネントを配置

カテゴリ	コンポーネント	個数	配置先
Standard	TLabel 	2個	Panel1

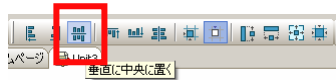
コンポーネント名	プロパティ名	設定する値
Label1	AutoSize	False
	Name	lblEnzan
	Width	20
Label2	Caption	=
	Name	lblEqual
	Visible	False

[演習]位置あわせ

1. TEditコンポーネントとTLabelコンポーネントを選択

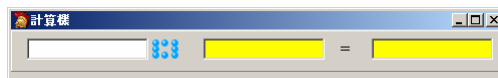


2. [垂直に中央に置く]ボタン押下



[演習] TLabelコンポーネントの設定

コンポーネント名	プロパティ名	設定する値
Label1	Caption	(空白)



[演習] TFlowPanelコンポーネントを配置

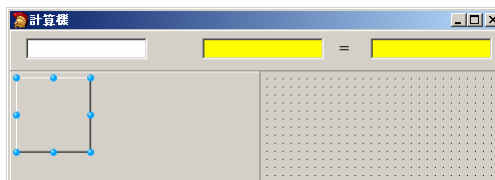
カテゴリ	コンポーネント	個数	配置先
Additional	TFlowPanel 	1個	フォーム

コンポーネント名	プロパティ名	設定する値	
FlowPanel1	Align	alLeft	
	Caption	(空白)	
	Padding	Bottom	5
		Left	5
		Right	5
		Top	5
Width	250		

[演習] TSpeedButtonコンポーネントを配置

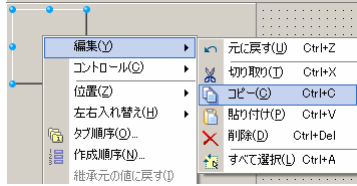
カテゴリ	コンポーネント	個数	配置先
Additional	TSpeedButton 	1個	FlowPanel1

コンポーネント名	プロパティ名	設定する値
SpeedButton1	Height	75
	Width	75



[演習] TSpeedButtonコンポーネントをコピー

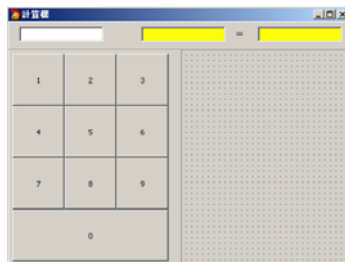
1. SpeedButton1コンポーネントを右クリック
2. [編集 | コピー]メニューを選択



3. SpeedButton1コンポーネントを右クリック
4. [編集 | 貼り付け]メニューを選択
5. 「3.」と「4.」を繰り返して、SpeedButtonを10個にする

[演習] TSpeedButtonコンポーネントを設定

コンポーネント名	プロパティ名	設定する値
SpeedButton1～9	Caption	1～9
SpeedButton10	Caption	0
	Width	225



[演習]ファイルに保存

1. [ファイル | すべて保存]メニューを選択
2. 以下のとおり入力し、[保存]ボタン押下

ファイル名	CalcForm
-------	----------

3. 以下のとおり入力し、[保存]ボタン押下

ファイル名	Calclator
-------	-----------

設計モードと実行モード



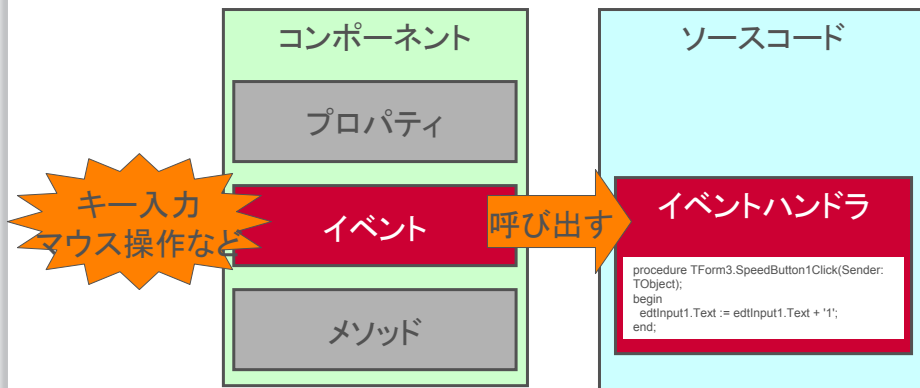
- プログラムの変更は、設計モードしか出来ない

[演習]プログラムを実行

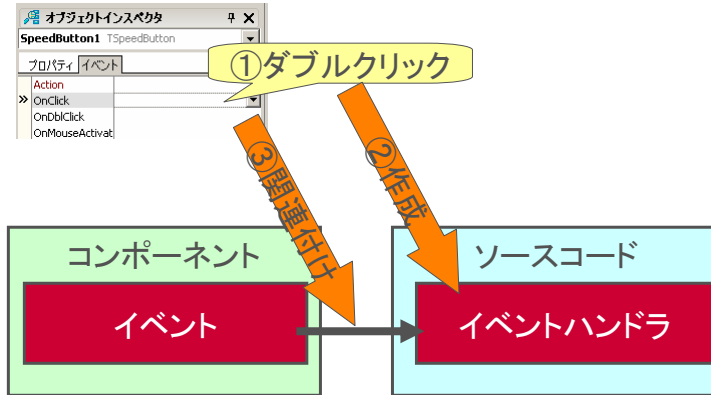
1. [実行 | 実行]メニューを選択
 - プログラムが実行され、実行モードになる

2. 動作を確認したら、計算機ウィンドウの[X]ボタン押下
 - プログラムが終了し、設計モードに戻る

イベントとイベントハンドラ



イベントハンドラの作り方

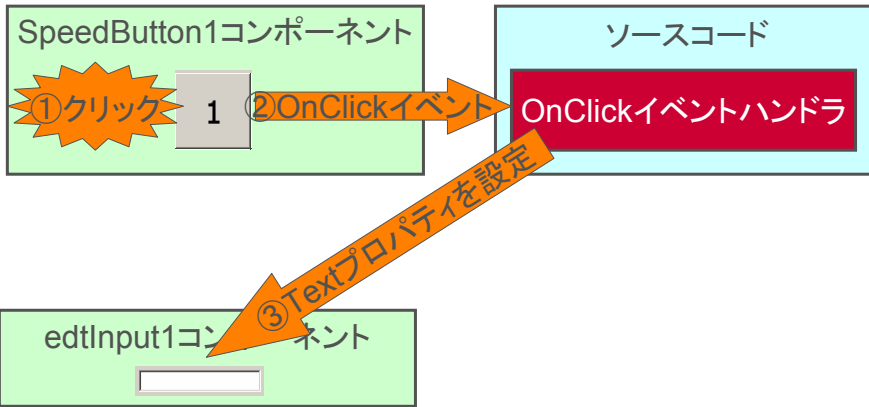


フォームとソースの切り替え

[コード] デザイン / 履歴

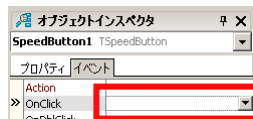
- [コード]タブ
 - ソースコードを表示
- [デザイン]タブ
 - フォームを表示

テンキーのOnClickイベントハンドラの概要



[演習]テンキーのOnClickイベントハンドラを作成

1. オブジェクトインスペクタでSpeedButton1のOnClickイベントをダブルクリック



2. 以下のソースコードを入力


```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    edtInput1.Text := edtInput1.Text + '1';
end;
```

[演習]テンキーのOnClickイベントハンドラを作成

1. SpeedButton2～10のOnClickイベントハンドラを作成
2. SpeedButton1のOnClickイベントハンドラのbegin～endのコードをコピー
3. SpeedButton2～10のOnClickイベントハンドラにペースト
4. 「1」の部分を各ボタンの数字(2～9,0)に変更

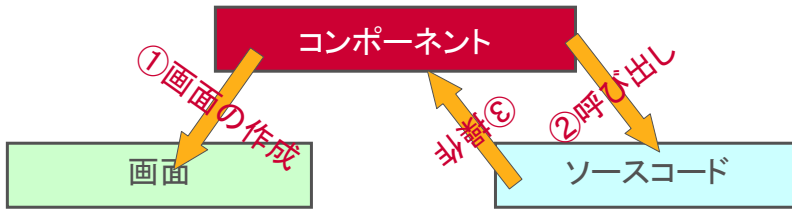
```
procedure TForm3.SpeedButton2Click(Sender: TObject);
begin
  edtInput1.Text := edtInput1.Text + '2';
end;
```

[演習]プログラムを実行

1. [ファイル | すべて保存]メニューを選択
 2. [実行 | 実行]メニューを選択
 3. テンキーをクリック
 - 押したキーの値がedtInput1に表示される
- 
4. 動作を確認したら、計算機ウィンドウの[X]ボタン押下
 - プログラムが終了し、設計モードに戻る

ここまでのまとめ

- プログラミングの基本は「コンポーネント」



- プロパティとイベントはオブジェクトインスペクタで設定
- イベントハンドラにソースコードを記述


休憩

演習： 計算機の四則演算ボタンを作る

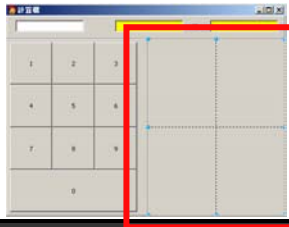
学習内容

- if文
- データ型

[演習] TGridPanelコンポーネントを配置

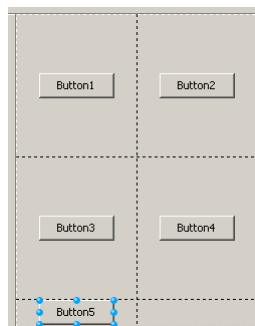
カテゴリ	コンポーネント	個数	配置先
Additional	TGridPanel 	1個	フォーム

コンポーネント名	プロパティ名	設定する値
GridPanel1	Align	alClient
	Caption	(空白)



[演習] TButtonコンポーネントを配置

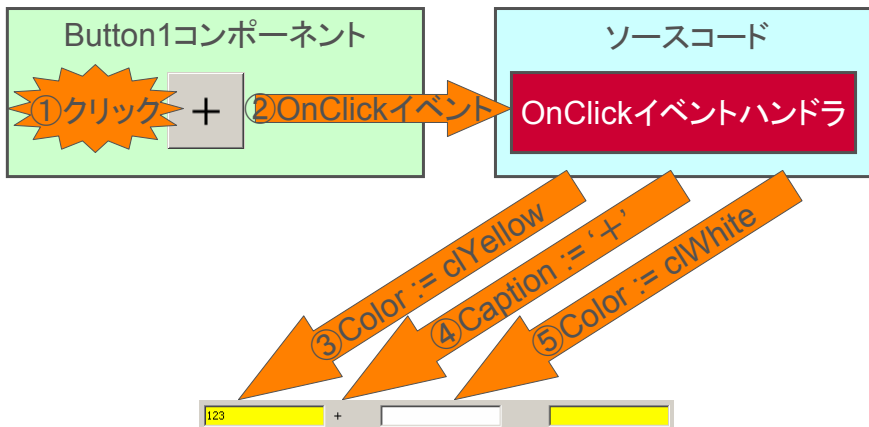
カテゴリ	コンポーネント	個数	配置先
Standard	TButton 	5個	GridPanel1



[演習] TButtonコンポーネントを配置(つづき)

コンポーネント名	プロパティ名	設定する値
Button 1	Caption	+ (全角の+)
	Height	75
Button2	Caption	- (全角の-)
	Height	75
Button3	Caption	× (全角の×)
	Height	75
Button4	Caption	÷ (全角の÷)
	Height	75
Button5	Caption	= (全角の=)

四則演算ボタンの概要




[演習] [+]ボタンのOnClickイベントハンドラ

1. 「+」Button1のOnClickイベントハンドラを作成
2. テキスト P.152の中段のソースコードを入力

[演習] 「-」「×」「÷」ボタンのOnClickイベントハンドラ



1. 「-」ボタンのOnClickイベントハンドラを作成
2. テキスト P.153のソースコードを入力
 - 「+」ボタンとほぼ同じコードなのでコピペしてください
3. 同じように「×」「÷」ボタンのOnClickイベントハンドラを作成
4. テキスト P.153のソースコードを入力
 - 「+」ボタンとほぼ同じコードなのでコピペしてください

[演習]プログラムを実行

1. [ファイル | すべて保存]メニューを選択
 2. [実行 | 実行]メニューを選択
 3. テンキーをクリック
 4. 四則演算ボタンをクリック
 - 演算子が表示される
- 
5. 動作を確認したら、計算機ウィンドウの[X]ボタン押下
 - プログラムが終了し、設計モードに戻る

テンキーの動作を修正

- 四則演算ボタン押下前は、edtInput1に表示
- ボタン押下後は、edtInput2に表示

四則演算ボタン	画面
押下前	
押下後	

- 四則演算ボタン押下の判定は「lblEnzan.Captionプロパティ」で可能

if文

```
if 条件式 then
```

条件式が**成立**したときに実行する文 

```
else
```

条件式が**不成立**のときに実行する文 ;

```
if lblEnzan.Caption = " then
```

edtInput1.Text := edtInput1.Text + '1' 

```
else
```

edtInput2.Text := edtInput2.Text + '1';

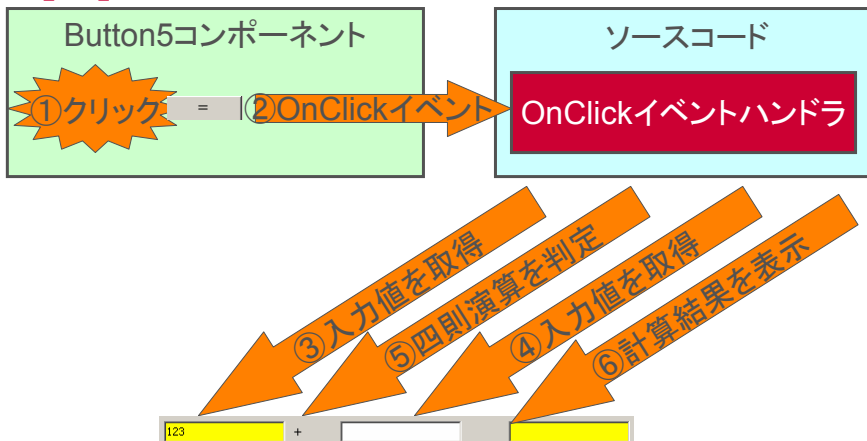
[演習]テンキーのOnClickイベントハンドラを修正

1. SpeedButton1のOnClickイベントハンドラを表示
2. テキスト P.154のとおり変更
3. SpeedButton2～10も同様に変更
 - SpeedButton1とほぼ同じコードなのでコピペしてください

[演習]プログラムを実行

1. [ファイル | すべて保存]メニューを選択
2. [実行 | 実行]メニューを選択
3. テンキーをクリック
4. 四則演算ボタンをクリック
5. テンキーをクリック
 - edtInput2に表示される
6. 動作を確認したら、計算機ウィンドウの[X]ボタン押下
 - プログラムが終了し、設計モードに戻る


[=]ボタンの概要



[演習] [=]ボタンのOnClickイベントハンドラ

1. 「=」ボタンのOnClickイベントハンドラを作成
2. テキスト P.163下段～P.164のソースコードを入力
 - 行端の「;」の有無を確認しながら入力してください

[演習]プログラムを実行

1. [ファイル | すべて保存]メニューを選択
2. [実行 | 実行]メニューを選択
3. テンキーをクリック
4. 四則演算ボタンをクリック
5. テンキーをクリック
6. [=]ボタンをクリック
 - 計算結果が表示される 
7. 動作を確認したら、計算機ウィンドウの[X]ボタン押下
 - プログラムが終了し、設計モードに戻る

練習問題

[練習問題]

- 計算結果が3の倍数なら、edtKekkaコンポーネントを赤色にする

• ヒント

- 3の倍数か判定する方法
if $\text{Frac}(\text{Ans} / 3) = 0.0$ then

- 赤色
clRed

Frac関数は、小数部の値を返す
例: $\text{Frac}(1.3) = 0.3$

まとめ

Delphiチュートリアルまとめ

- Delphiプログラミングの基本は「コンポーネント」
 - 「コンポーネント」で画面を作成
 - 「コンポーネント」のイベントにソースコードを記述
 - ソースコードで「コンポーネント」を操作
- Delphiはコンポーネントを活用して、簡単にプログラムが作れる！

ありがとうございました