実践編:複雑なアプリケーションを作ってみよう

実践編では、これまでより少し複雑なアプリケーションを構築します。より実用的なアプリケーションを 作成し、アプリケーション構築のコツを習得しましょう。

◆目次

実践編: 複雑なアプリケーションを作ってみよう.....1

LESSON. 11 ガントチャートを使ってみよう	2
Step.1 ガントチャートとは?	2
- Step.2 ガントチャートで使用するデータ	2
Step.3 テーブルデータとガントチャートを表示する	3
Step.4 テーブルとガントチャートを連動させる	. 16

Step.3 新しい ノレームの利用	31
Step.4 イメージビューワーの設定変更	37
Step.5 複合コンポーネントによる階層化	40
Step.6 複合コンポーネントの利用	44
Step.7 機能を追加する	58

Step.1	任息们的計算	74
Step.2	[0]ボタンと[. (小数点)]ボタンを追加	86
Step.3	四則演算	94
Step.4	複合コンポーネントによる階層化1	19
Step.5	複合コンポーネントの利用1	21

Lesson.11 ガントチャートを使ってみよう

MZ Platformの標準コンポーネントの「ガントチャート」を使ってみましょう。

Step.1 ガントチャートとは?

工場などで人や工程などの管理に用いられる帯状のグラフです。横軸では時間などの期間、縦軸では人・ 製造工程・作業工程等を表し、それぞれ開始日(時間)、完了日(時間)といった情報を帯状に示します。 MZ Platform では標準コンポーネントでガントチャートを提供しています。



<u>Step.2 ガントチャートで使用するデータ</u>

ガントチャートは[テーブル] コンポーネントのデータを元に作成します。最初にガントチャートに必要 なデータが入力されている[テーブル] コンポーネントを作成する必要があります。

1)ガントチャート用データの作成

ガントチャート用データをテーブルに入力します。

ガントチャートの帯の部分のデータは、「開始」データと「終了」データが必要になります。

つまり1つの帯に2列必要です。

ここには開始時刻と終了時刻が入力されています。



2) ガントチャートで使用できるデータ

列名	データの意味
1 列目	各行の項目名となる文字列データです。
	グラフにした場合、この項目が軸の作業名として表示されます。
2列目以降(2列1組)	実際にガントチャートで表示する開始・終了日時のデータです。
	データ型は日付と文字列が使えます。
	"年/月/日 時∶分∶秒"という形式で書きます。
	例えば、"2015/05/15 15:00:00(数字や記号、空白などすべて半角
	で入力)"となります。

Step.3 テーブルデータとガントチャートを表示する

実際に操作して、テーブルデータとガントチャートを表示しましょう。

1) ガントチャートに必要なデータを入力する

テーブルコンポーネントを準備してガントチャートに必要なデータを入力しましょう。

完成図 テーブルデータを用意します。

🍰 作業予定表	Ę						3
WorkName	神立		石岡		高萩		
見積り	2011/11/18 8:30:00	2011/11/19 12:00:00					
設計			2011/11/18 15:00:00	2011/11/25 9:00:00			
調達					2011/11/20 9:00:00	2011/11/25 17:00:00	

準備しここでは以下のコンポーネントを使用します。

コンポーネント名	必要数	
■アプリケーション	(1)	
■フレーム	1	[画面構成部品]-[ウィンドウ]-[フレーム]
<mark>■</mark> テーブル	1	[画面構成部品]-[テーブル]-[テーブル]



① 必要なコンポーネントを追加します。

作業領域で右クリックー[コンポーネント追加] – [画面構成部品] – [ウィンドウ] – [フレーム]、 作業領域で右クリックー[コンポーネント追加] – [画面構成部品] – [テーブル] – [テーブル] とクリックします。

MZ Platform アプリケーションドルダー		
ファイル 編集 アブリケーション オブション ヘルブ		コメント行検索 ▼
アプリケーション名称		
■ アプリケーション KEY:*** フレーム ID:1 KEY:**フレーム1*		
コンポーネント追加・	画面構成部品 ・ ウィンドウ ・	
コンポーネントー括追加	処理部品 メニュー →	
複合コンボーネント作成 🏮	入出力 リーパネル ト	
複合コンポーネント追加 り	クラス指定 テキスト ト	
貼り付け	ボタン	
オペイ雇用	7-70 7	
サインで用にろ		
9. (090.0		
コンポーネント検索	コンホホッンス ・ チェックギックフ ト	
GUIコンポーネント検索	テェッシネッシス ・	
コメント行追加		
コメント行検索	7-74174	
検索	グラフィックス・	
コノホーネントに構設定		
【アイコン凡例】 🛑 アプリケーション 🗐 画面構	成部品 🛑 ウィンドウ 🛑 メニュー 🗐 パネル 🧧	▶処理部品 🛑 複合(画面構成) 🛑 複合(処理) 🛑 リモート
実行 実行(設定可) 画面	編集 「「帳票編集」「ロード」「挿入」(保存	- 上書き保存 - クリア - 終7

接続確認

コンポーネント同士の接続を確認します。

開始

接続項目	接続関係
接続元コンポーネント	■マプリケ>,->,
(イベント発生コンポーネント)	
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■フレーム (ID:1)
起動メソッド	フレームを表示する()

終了

接続項目	接続関係
接続元コンポーネント	$\square \square \downarrow = I$ (ID:1)
(イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント	■ アプリケーション
起動メソッド	アプリケーションを終了する()

操 作

① [フレーム] コンポーネントと [アプリケーション] コンポーネントを接続します。

MZ Platform アプリケーミ	ションビルダー					
ファイル 編集 アブリケーシ	ィョン オブション ヘルブ			コメント行	検索	•
アブリケーション名称						
■ アプリケーション		プリケーション開始イベント	フレームを表:	示する	<u>■フレーム</u>	
KEY:"		•			ID:1 KEY:"フレーム1"	
ロ フレーム ID:1		アクションイベント	アプリケーション	を終了する	アブリケーション	
	$ \rightarrow $				(KEY:"	
ID:2 KEY:"テーブル2"						
[アイコン凡例] 📕	アブリケーション 🗊 画	画構成部品 🛑 ウィンドウ	■メニュー ■パネル	▶ 処理部品 🛑 複合(画面相	転) 🗐 複合(処理)	■リモート
実	行 実行(設定可)	画面編集	ロード 挿入 保存	こう 「上書き保存」 ク	リア 終了	

画面編集

ウィンドウ (フレーム) に [テーブル] を配置しましょう。

- ① 画面編集ツールボタンをクリックし、[画面編集] 画面に入ります。
- ② [テーブル] コンポーネントをフレームに貼り付けます。

[画面編集] 画面上で右クリック- [コンポーネント追加] - [テーブル] コンポーネントと クリックします。

₩ 画面編集	
編集	
₩ 画面編集 編集 - 『フレーム [ID:1] (KEY:"フレーム1")	フレーム (ID.1] (KEY:"フレーム1") コンボーネント通知 新規コンボーネントが追加 新規コンボーネントが追加 コンボーネントが追加 コンボーネントが追加 二コンボーネントが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーが追加 二コンボーネントーがし 二コンボーネントーが追加 二コンボーネントーが加 二コンボーネントーがに 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二コンボーネントーが 二日 二コンボーネントーが 二日 二日 二日 二日 二日 二日 二日 二日 二日 二
	記選 横方向誕列 ▼ 図 自動サイズ設定 グリッド開照 5 0 10 20 30 40 50 閉じる

③ 追加できたら閉じるをクリックし、ビルダー画面に戻ります。

二 集	
ー フレーム [ID:1] (KEY:"フレーム1") ー ゴ テーブル [ID:2] (KEY:"テーブル2")	フレーム (ID:1) (KEY."フレーム1")
- □ <i>Ţ</i> - ⊃Jµ (ID:2] (KEY: <i>Ţ</i> - ⊃Jµ2')	
	配置 横方向整列 ▼ 図 自動サイズ設定 グリッド間隔5 0 10 20 30 40 5 閉じる

④ テーブルの画面が完成したことを確認します。
 実行(設定可)で実行します。



⑤ テーブルを作成します。

項目名を入力する列を文字列型で作成します。

実行(設定可)で実行したテーブル上で右クリックします。

```
[テーブル]-[列]-[追加]-[一列]-[文字列]をクリックします。
```



- ⑥ 開始・終了日時を入力する列を日付型で作成します。
 (設定可)で実行したテーブル上で右クリックします。
 [テーブル] [列] [追加] [複数列] [日付...]をクリックします。
- ⑦ 列数を指定します。

ここでは「6」列にします。

追加列数の入力
<u>追加列</u> 数を入力してください。 6 適用 取消

⑧ テーブルの行数を指定します。

実行(設定可)で実行したテーブルの上で右クリックします。 [テーブル] - [行] - [追加] - [複数行...]をクリックします。



行数を指定します。

ここでは「3」行にします。

追	助行数	ወ入力	x
		適用 取消	



Ш

•

Þ

①タイト	・ノレ					
			③列幅			
_			<u> </u>			
▲ 作業予定表	長		\square	I		
WorkName	神立		石岡		高萩	
見積り	2011/11/18 8:30:00	2011/11/19 12:00:00				
設計			2011/11/18 15:00:00	2011/11/25 9:00:00		
調達					2011/11/20 9:00:00	2011/11/25 17:00:00
						
	④データン	入力			②ウ	ィンドウ枠の幅

ウィンドウの名前(タイトル)を入力します。

ビルダー上のフレームコンポーネント上で右クリックー [属性情報設定...]をクリックします。 [Title] に「作業予定表」と入力します。

設定をクリックします。

₩ コンポーネント属性情報					
ComponentKeys	日本語: 英語:	NULL 1			
AllowRemoteInvocation	🔘 true 🔘 false				
AllowPullTransfer	💿 true 💿 false				
AllowPushTransfer	🔘 true 🔘 false				
ExtendedState	0				
Resizable	● true ○ false	=			
Title	作業予定表	NULL			
ContainerLayout					
AutoResize	💿 true 🔘 false				
MultiLocaleToolTipText	日本語: 英語:	NULL			
GridLayoutRows	0				
GridLayoutColumns	0				
ContainerOrderedFocusTraverse	🔘 true 🔘 false				
FocusTraverseByEnterEnabled	🔘 true 🔘 false				
GridInterval	5				
MultiLocaleTitle	日本語: 作業予定表 英語:	NULL			
ConfirmDialooVisible	true @ falco	-			

⑪ テーブル全体の幅を広げます。

画面編集をクリックします。

テーブルの周りの赤い線をドラッグし幅を広げます。

閉じるをクリックします。



12 列幅を調整します。

実行(設定可)で実行します。

列名の列と列の間にマウスポインタを合わせてドラッグして列幅を調整します。



13 データを入力します。

セルをダブルクリックしてデータを入力し、以下のように完成してください。

入力する日時データは自由です。例えば開始日は本日の日付とし、終了日は数日先の日付とします。 時刻データは必ず「秒」まで入力してください。

④ 作業予定表							
WorkName	神立		石岡		高萩		
見積り	2011/11/18 8:30:00	2011/11/19 12:00:00					
設計			2011/11/18 15:00:00	2011/11/25 9:00:00			
調達					2011/11/20 9:00:00	2011/11/25 17:00:00	

2) ガントチャートを表示する

テーブルのデータをガントチャートに表示しましょう。





コンポーネント名	必要数	
■ガントチャート	1	[画面構成部品]-[グラフ]-[ガントチャート]

必要なコンポーネントを追加します。
 ここでは [ガントチャート] コンポーネントを追加します。
 作業領域で右クリックー [コンポーネント追加] - [画面構成部品] - [グラフ]
 - [ガントチャート] とクリックします。

接続確認

コンポーネント同士の接続を確認します。

ガントチャートのデータを設定する

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■ガントチャート(ID:3)
起動メソッド	ガントチャートのデータを設定する(PFObjectTable)
<引数>	説明:ガントチャートのデータ
	取得方法:メソッド戻り値
	コンポーネント:テーブル
	メソッド/値:テーブルデータを取得する

操作

計算処理を設定しましょう。

① イベントの接続先コンポーネントを選びます。
 [アプリケーション] コンポーネントの[アプリケーション開始イベント] に2つめの
 起動メソッドを追加します。
 左側の[アプリケーション] コンポーネントの[アプリケーション開始イベント] 上で
 右クリック-[起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリック-[接続コンポーネント選択] -

[ガントチャート(ID:3)]をクリックします。

👑 MZ Platform アプリケーションビル	ダー		
ファイル 編集 アプリケーション オ	ブション ヘルブ	コメント行	〒検索 ▼
アブリケーション名称			
■ アブリケーション KEY:™	アプリケーション開始イベント	フレームを表示する	1フレーム 1D:1 KEY: "フレーム1"
□ フレーム □ 21 □ 21 KEY:"フレーム1" □ テーブル □ 10:2 KEY:"デーブル2" □ ガントチャート □ 13 KEY:"ガントチャート3"	7993V1×VF	 アブリケーション [ID:0] (KEY:") フレール [ID:0] (KEY:") テーブル [ID:2] (KEY:"テーブル2") ガントチャート [ID:3] (KEY:"ガントチャート3") 	 接続コンボーネント選択… 接続コンボーネント選択 起動メソッド設定… イベント番号設定 起動モード 起動モード 起動メソッド追加 コピー 貼り付け 切り取り 削除
			コメント編集 接続コンボーネント宣言位置検索 ブレークポイント設定/解除
【アイコン凡例】 🛑 アブリケ 実行 📑	ーション 画面構成部品 ウィンドウ 影行(設定可) 画面編集 帳票編集	●メニュー ■バネル ■処理部品 ■複合価面 ロード 挿入 保存 上書き保存 /	構成) 💼 複合(処理) 🛑 リモート ツリア 終了

- ② 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド (処理)を選びます。
 [メソッド]の
 をクリックします。
 [ガントチャートのデータを設定する (PF0b jectTable)]をクリックします。
 引数を設定します。
 説明:ガントチャートのデータ
 取得方法:メソッド戻り値
 - コンポーネント:テーブル
 - メソッド/値:テーブルデータを取得する

設定後、了解ボタンをクリックします。

記動メソッド情報							
メソッド ガントチャートのデータを設定する(PFObjedTable)							
NO 型	記印	取得方法	コンポーネント	メソッド/値			
0 PFObjectTable	ガントチャートのデータ	メソッド戻り値	テーブル [ID:2] (KEY:"テーブル2")	テーブルデータを取得する			
				了解 取消し			

③ [アプリケーション] コンポーネントと [ガントチャート] コンポーネントが接続されます。 フレームはデータを設定してから開くように変更します。

[ガントチャート] コンポーネントと [フレーム] コンポーネントを入れ替えます。

[ガントチャート] コンポーネントをドラッグして [フレーム] コンポーネントの上に移動します。

MZ Platform アプリケーションビルダー	
ファイル 編集 アブリケーション オブション ヘルブ	コメント行検索 🔹
アプリケーション名称	
■ アブリケーション KEY: [™]	フレームを表示する フレーム ブレームを表示する ロ:1 ガントチャートのデータを設定する ガントチャート リントチャートのデータを設定する ロ:1
フレーム アクションイベント ID:1 ID:1 IF-ブル ID:1 ID:2 IF-ブル2 IIガントチャート ID:2 IIガントチャート ID:2 IIガントチャート3 ID:2	<u>アブリケーションを終了する</u> KEY: "
【アイコン凡例】 🗐 アプリケーション ■1画面構成部品 🛑 ウィンドウ	●メニュー ●パネル ●処理部品 ●複合(画面構成) ●複合(処理) ●リモート
実行 実行(設定可) 画面編集 棟票編集	ロード 挿入 保存 上書き保存 クリア 終了

- ④ 画面を作成します。
 <u>画面編集</u>をクリックします。
- ⑤ [ガントチャート] コンポーネントをフレームに追加します。
 画面上で右クリック- [コンポーネント追加] [ガントチャート] コンポーネントとクリックします。
 横方向に配置されるので [手動配置] にして上下にバランスよく配置しましょう。
 追加できたら閉じるをクリックします。
- ⑥ ガントチャートの画面が追加できたことを確認します。
 実行(設定可)で実行します。

⑦以下を参考にガントチャート表示の調整をします。 グラフエリアで右クリックし、

: [ガントチャート] - [作業項目軸] - [表示項目数...] 1) グラフ表示の項目数 2) グラフ表示の開始日時 : [ガントチャート] - [時間軸] - [表示開始日時...] 3) グラフの表示期間 : [ガントチャート] - [時間軸] - [表示期間...] :「ガントチャート]-「時間軸]-「大目盛間隔...] 4) 数値軸の目盛り間隔 : [ガントチャート] - [時間軸] - [小目盛間隔...] 5)時間軸ラベルの表示形式 : [ガントチャート] - [時間軸] - [大項目ラベル表示形式...] - [設定する] :「ガントチャート]-「時間軸]-「小項目ラベル表示形式...] - [設定する] *yyyy/MM/dd(年/月/日)、H:mm:ss(時:分:秒)等で設定 6)余白の設定 :[パネル] - [余白] :「ガントチャート]-「作業項目軸]-「ラベル表示幅] 7) ラベル位置の調整

: [ガントチャート] - [作業項日軸] - [フベル表示吶 : [ガントチャート] - [時間軸] - [ラベル表示高さ]



知っていると便利!

[ガントチャート] へは Step2 で紹介した基本のデータ形式だけではなく、順に「項目名」、「系列名」、「開始日時」、「終了日時」、「タスク名」となる列データを持つ、5 列ないし4列(「タスク名」の設定は任意)の テーブルデータを設定できます。このメソッドでデータを設定した時は、次の Step.4 で紹介する「テーブル とガントチャートの連動」もこのチュートリアルとは違った設定方法になります。

例)

テーブルデータを用意します。

				////
\sim				
工程	受注番号	開始時刻	終了時刻	工程
- 鍛造 - /	A03	2011/11/11 9:00:00	2011/11/11 15:00:00	鍛造
切削	A02	2011/11/12 9:00:00	2011/11/12 12:45:00	切削
溶接	A01	2011/11/13 9:30:00	2011/11/13 14:15:00	溶接
研磨 /	A02	2011/11/15 7:15:00	2011/11/15 8:30:00	研磨
検査	A02	2011/11/17 8:15:00	2011/11/17 8:45:00	検査
- 鍛造 - /	A04	2011/11/17 9:00:00	2011/11/17 12:50:00	鍛造
切削 /	A02	2011/11/18 9:00:00	2011/11/18 11:00:00	切削
放電 /	A04	2011/11/19 9:00:00	2011/11/19 16:12:00	放電
研磨 /	A04	2011/11/13 9:45:00	2011/11/13 13:37:00	研磨
めっき /	A03	2011/11/17 13:15:00	2011/11/17 17:15:00	めっき
旋盤	A01	2011/11/19 17:15:00	2011/11/19 18:30:00	旋盤

② [ガントチャート] にデータを設定します。

起動メソッド設定の際に「全メソッド対象」にチェックを入れ、[setRecordTable (PFObjectTale)]を 選びます。

<mark>™</mark> _ #	2 起動メソッド情報							
	メソッド setRecordTable(PFObjectTable)							
NO	型	說明	取得方法	コンポーネント	メソッド/値			
(PFObjectTable		メソッド戻り値	テーブル [ID:2] (KEY:"テーブル2")	テーブルデータを取得する			
					了解 取消し			

③ [ガントチャート] が表示されます。



<u>Step.4</u> テーブルとガントチャートを連動させる

テーブルのデータを変更したらガントチャートに反映されるように、またはガントチャートのデータを変 更したらテーブルに反映されるように設定を変更しましょう。

1)テーブルのイベント番号

コンポーネントとコンポーネントは「イベント」で接続しています。

ここでは [テーブル] コンポーネントと [ガントチャート] コンポーネントを接続します。

一方のデータを変更したときにもう一方のデータも変更したいので [データ更新イベント]を利用します。

テーブルデータの更新の種類は複数のケースがあります。キーボードから値を修正した場合や、行や列の 追加/削除のようにデータ構造を変更する場合も[データ更新イベント]が発生します。このように同じイ ベント(ここではデータ更新イベント)でも複数の意味を持つ場合があります。

こうした複数のイベントの内容を識別するために『イベント番号』を使用します。

『イベント番号』とは同じイベントが複数の意味を持つ場合、それぞれのイベントに番号が振ってあり、ど のイベントなのか識別することができるというものです。

イベント番号	内容
0	セルの値が更新されたケース
1	行が更新されたケース
2	列が更新されたケース
10	行が追加されたケース
11	複数行が追加されたケース
12	列が追加されたケース
13	複数列が追加されたケース
20	行が削除されたケース
21	全行が削除されたケース
22	列が削除されたケース
23	全列が削除されたケース
24	全行列が削除されたケース

[データ更新] イベントのイベント番号は以下のとおりです。

ここでは、上記すべての場合にデータを更新したいので『イベント番号』を設定するのではなく、 『定常起動』(イベント発生時は常に起動する)を利用します。

2) ガントチャートのイベント番号

ガントチャートのデータ更新は「マウスでドラッグ」した場合のみイベントが発生しますので、複数のケー スはありません。イベントをイベント番号で区別する必要がない場合、接続されているメソッドをイベント 発生時に常に起動するように指定します。

3)テーブルデータとガントチャートを連動

[テーブル] コンポーネントと [ガントチャート] コンポーネントを連動させましょう。



完成図 テーブルとガントチャートを連動させます。

テーブルのデータをガントチャートへ設定する

コンポーネント同士の接続を確認します。

接続確認

接続項目	接続関係
接続元コンポーネント	■テーブル (ID:2)
(イベント発生コンポーネント)	= (10.2)
発生イベント	データ更新イベント
イベント番号	定常起動
接続先コンポーネント	■ガントチャート(ID:3)
起動メソッド	ガントチャートのデータを設定する(PFObjectTable)
<引数>	説明:ガントチャートのデータ
	取得方法:イベント内包
	メソッド/値:イベント対象データ

ガントチャートのデータをテーブルに設定する

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	
発生イベント	データ更新イベント
イベント番号	定常起動
接続先コンポーネント	■テーブル(ID:2)
起動メソッド	テーブルデータを設定する(PFObjectTable)
<引数>	説明:テーブルデータ
	取得方法:イベント内包
	メソッド/値:イベント対象データ

はじめに、テーブルのデータをガントチャートへ設定するよう接続します。

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[テーブル(ID:2)] コンポーネント上で右クリックー[イベント処理追加]
 [データ更新イベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。
 左側の[テーブル] コンポーネントの[データ更新イベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 [ガントチャート(ID:3)] をクリックします。

③ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド (処理)を選びます。
 [メソッド]の をクリックします。
 [ガントチャートのデータを設定する (PFOb ject Table)]をクリックします。
 引数を設定します。
 説明:ガントチャートのデータ
 取得方法:イベント内包
 メソッド/値:イベント対象データ

設定後、了解ボタンをクリックします。

🔛 起動メソッド情報				×
	メソッド ガントチャートのデー:	タを設定する(PFObjectTa	ble)	▼ □ 全メソッド対象
NO 型	記印	取得方法	コンポーネント	メソッド/値
0 PFObjectTable	ガントチャートのデータ	イベント内包	-	イベント対象データ
				了解 取消し

次にガントチャートのデータをテーブルに設定するように接続します。

- ④ 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ガントチャート(ID:3)] コンポーネント上で右クリックー[イベント処理追加]
 [データ更新イベント] とクリックします。
- 「イベントの接続先コンポーネントを選びます。
 左側の[ガントチャート(ID:3)] コンポーネントの[データ更新イベント]上で
 右クリック-[起動メソッド追加] とクリックします。
 薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリック-[接続コンポーネント選択] [テーブル(ID:2)] をクリックします。
- ⑥ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...] をクリックします。
 起動メソッド (処理) を選びます。
 [メソッド] の ・ をクリックします。
 [テーブルデータを設定する (PF0b ject Table)] をクリックします。
 引数を設定します。
 説明:テーブルデータ
 取得方法:イベント内包
 メソッド/値:イベント対象データ

設定後、了解ボタンをクリックします。

▶ 起動メソッド情報				×
	メソッド テーブル	データを設定する(PFObjectT	able) 👻	■ 全メソッド対象
NO 型	ii 兑8月	取得方法	コンポーネント	メソッド/値
0 PFObjectTable	テーブルデータ	イベント内包	-	イベント対象データ
				「ア解」「取消し

⑦ イベント番号を確認します。

今回は常に起動するので『定常起動』になっていることを確認します。 接続先コンポーネント(ここではテーブルコンポーネント)上で右クリック−[イベント番号設定] - [定常起動]をクリックします。

W MZ Platform アブリケーションビルダー	
ー ファイル 編集 アブリケーション オブション ヘルブ	
アプリケーション名称	
■ アプリケーション アプリケーション KEY: "	ガントチャート ID:3 KEY:"ガントチャート3"
フレームを表示する	ロンレーム ID:1 KEY: "フレーム1"
	■ アブリケーション KEY:"
「 <u>「テーブル</u> 「 <u>テーブル</u> 「シーマー」」の データ変動イベント ガントチャートのデータを設定する	ガントチャート D:3 ケントチャート
(KEY:" <u>5) > F + - F 3"</u>	接続コンボーネント選択 接続コンボーネント選択
	1980にコンホーキシー 2013
定常起	が イベント番号設定
	ト番号設定。起動モード
	起動メソッド追加
	コピー
	貼り付け
	UIGHX0 BIIR÷
	接続コンポーネント宣言位置検索
	ブレークポイント設定/解除
「「イコンパ(例」 ■)アブリケーション 画面構成部品 ■)ワインドワ メニュー ■/パネル 処理部品	■ 後首(画面構//// ■) (2日(火*生) ■) ソート・
実行 実行感定可) 画面編集 帳票編集 ロード 挿入 保存 上書き	保存 クリア 終了

デーブルとガントチャートが連動します。
 デーブルのデータやガントチャートのデータを変更しましょう。



ここまで進めるとビルダー上では以下のようになります。

■アブリケーション	アプリケーション開始イベント	ガントチャートのデータを設定する	ガントチャート
KEY:""			ID:3 KEY:"ガントチャート3"
		フレームを表示する	
			ID:1 КЕҮ:"フレーム1"
■フレーム	アクションイベント	アプリケーションを終了する	アプリケーション
ID:1 KEY:"フレーム1"			KEY:"
<u> </u> テーブル	データ更新イベント	ガントチャートのデータを設定する	ガントチャート
ID : 2 KEY : "テーブル2"			ID:3 KEY:"ガントチャート3"
ガントチャート	データ更新イベント	テーブルデータを設定する	□ テーブル
ID:3 KEY:"ガントチャート3"			ID:2 KEY:"テーブル2"

知っていると便利!

ガントチャートには「縦ドラッグ」操作や「コネクタ」、「アイコン」なども設定することができます。

縦ドラッグ操作

実行(設定可)で実行し、グラフエリアで右クリックー[ガントチャート]ー [縦ドラッグ]ー [有効]とします。マウスでタスクをドラッグします。



② コネクタ設定

コネクタを開始したいタスクを右クリックして、ドラッグしたまま接続したいタスク上に カーソルを移動し、離します。

コネクタを削除するときは、マウスで選択した後 Delete キーを押してください。



③ アイコン設定

<u>実行(設定可)</u>で実行し、グラフエリアで右クリックー[ガントチャート]-[アイコン]-[表示する]とします。

タスク上で右クリックー [ガントチャート] - [アイコン] - [設定する...]を選びます。アイコン設定 の画面が出るので、表示したい画像のファイルを参照ボタンを押し、選択します。



Lesson.12 いろいろな画像ファイルを表示する

ここでは「ファイル入出力」コンポーネントを利用して、ファイルを呼び出す方法をご紹介します。 これまでフレームを1つしか使用していませんでしたが、ここでは2つのフレームを使用します。 また「複合コンポーネント」という MZ Platform の考え方を使った効率の良いアプリケーション構築方法を ご紹介します。

Step.1 利用できる画像ファイルの種類

MZ Platform では画像ファイルを扱うことが可能です。取り扱い可能な主なファイル形式は以下のとおりです。

- GIF
- $\boldsymbol{\cdot} \text{ JPEG}$
- PNG
- BMP

<u>Step.2</u> 画像ファイルの入力

画像ファイルを取り込んで表示しましょう。



ここでは以下のコンポーネントを使用します。

コンポーネント名	必要数	
■アプリケーション	(1)	
■フレーム	1	[画面構成部品]-[ウィンドウ]-[フレーム]
■ボタン	1	[画面構成部品]-[ボタン]-[ボタン]
■ファイル選択	1	[画面構成部品]-[ダイアログ]-[ファイル選択]
■画像ファイル入力	1	[入出力]-[ファイル]-[画像ファイル入力]

操作

① 必要なコンポーネントを追加します。
 作業領域で右クリック-[コンポーネント追加] - [画面構成部品] - [ウィンドウ] - [フレーム]、
 作業領域で右クリック-[コンポーネント追加] - [画面構成部品] - [ボタン] - [ボタン]、
 作業領域で右クリック-[コンポーネント追加] - [画面構成部品] - [ダイアログ]
 - [ファイル選択]、
 作業領域で右クリック-[コンポーネント追加] - [入出力] - [ファイル] - [画像ファイル入力]
 とクリックします。

₩ MZ Platform アプリケーションビルダー	
ファイル 編集 アブリケーション オブション ヘルブ	コメント行検索
アプリケーション名称	
■ アブリケーション <u>KEY: ***</u> ■ フレーム 10:1 (EY: ** 大クレーム)* ■ ボタン 10:2 (EY: *ボタン2** 「ファイル違訳 TO:3 KEY: **ファイル違訳3*	
	 ファイル CSV入力 CSVH力
	マキントコーイルスカ デキストファイルスカ テキストファイルスカ 画像ファイルスカ 画像ファイルスカ ゴロバティファイル入出力 営声再生 音声録音 マログ
【アイコン凡例】 ● アブリケーション ●●画面構成部品 ●●ウィンドウ ●●メニ	■ XML変換 うべル付きしストXML変換 複合(画面構成) ■複合(処理) ■ リモート
実行 実行 (設定可) 画面編集 「帳票編集」 ロー	Excelファイルアクセス 評 クリア 終了

準 備

画面編集

画面を作成します。

 画面編集をクリックします。
 [ボタン] コンポーネントをフレームに追加します。
 [画面編集]画面上で右クリックー[コンポーネント追加] - [ボタン] コンポーネントと クリックします。

***** 	フレーム (ID:1) (KEY."フレーム1")
	コンボーネント-新造加加 新規コンボーネント新造加加 新規コンボーネント新造加加 コンボーネント新造加加 オコンボーネント
	配置 横方向整列 ▼ 図 自動サイズ設定 グリッド間隔5 pm 20 30 40 50 閉じる

④ 追加できたら閉じるをクリックし、ビルダー画面に戻ります。

🤟 画面編集	
≣ بي	
■ ■ 【フレーム [[D:1] (KEY:"フレーム1") └─↓ ボタン [ID:2] (KEY:"ボタン2")	フレーム (ID.1) (KEY."フレーム1") (ボタン>
	記畫 模方向整列 ▼ 図 自動サイズ設定 グリッド間隔5 0 10 20 30 40 閉じる

接続確認

コンポーネント同士の接続を確認します。

開始

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■フレーム (ID:1)
起動メソッド	フレームを表示する()

終了

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント	■アプリケーション
起動メソッド	アプリケーションを終了する()

ボタンをクリックしたらファイル選択画面が表示される

接続項目	接続関係
接続元コンポーネント	ボタン (ID:2)
(イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント	■ファイル選択(ID:3)
起動メソッド	単数 Open 用ファイル選択ダイアログを表示する (Component)
<引数>	説明:親コンポーネント
	取得方法:コンポーネント
	コンポーネント : フレーム(ID:1)

ファイル選択画面からファイル名を指定して画像を読み込む

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	
発生イベント	データ選択イベント
接続先コンポーネント	■画像ファイル入力(ID:4)
起動メソッド	ファイル名を指定して画像を読み込む(String)
<引数>	説明:読み込むファイル名
	取得方法:イベント内包
	コンポーネント:選択データ
イベント番号	1

操作

① [フレーム] コンポーネントと [アプリケーション] コンポーネントを接続します。

MZ Platform アプリケーションビルタ	1-		
 ファイル 編集 アブリケーション オブ	ション ヘルブ		コメント行検索 🔻
アプリケーション名称			
■ アプリケーション	アプリケーション開始イベント	フレームを表示する	
	アクションイベント		KEY: "フレーム1"
ID:1 KEY:"フレーム1"	•	アプリケーションを終了する	KEY: "
<u> ボタン</u> D:2 /(TV:"ボカン:2"			
マアイル選択			
U:3 KEY:"ファイル選択3"			
回家ファイル入力 ID:4 KEY:"画像ファイル入力4"			
【アイコン凡例】 🛑 アブリケー	ション 📄 画面構成部品 🗐 ウィンドウ	●メニュー ■バネル ●処理部品 (■複合(画面構成) 🛑 複合(処理) 🛑 リモート
実行実行	う(設定可) 画面編集 帳票編集		保存 クリア 終了

- ボタンをクリックしたらファイル選択画面が表示されるように接続します。
- ② 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ボタン(ID:2)] コンポーネント上で右クリックー[イベント処理追加]
 [アクションイベント] とクリックします。
- ③ イベントの接続先コンポーネントを選びます。
 左側の[ボタン(ID:2)] コンポーネントの[アクションイベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 [ファイル選択(ID:3)] コンポーネントをクリックします。

(
₩ MZ Platform アプリケーションビルダー			
ファイル 編集 アブリケーション オブション	ヘルプ		コメント行検索▼
アプリケーション名称			
アプリケーション	アプリケーション開始イベント		
	•	フレームを表示する	
ID:1	P99894X2F	アブリケーションを終了する	7500-982
KEY:"70-41"			KEY:"
	アクションイベント		
KEY:"ボタン2"	•		接続コンボーネント選択
● ファイル選択		アプリケーション [ID:0] (KEY:**)	接続コンポーネント選択
ID:3 KEY:"ファイル選択3"		■ フレーム [ID:1] (KEY:"フレーム1")	起動メソッド設定
「画像ファイル入力 」		ホタン [ID:2] (KEY:"ボタン2")	イベント番号設定)
		ファイル選択 [ID:3] (KEY:"ファイル選択3")	
		■ 画像ファイル入力 [ID:4] (KEY:"画像ファイ.	ル入力4")
			記動メソッド追加
			コピー
			貼り付け)
			切り取り
			削烱余
			コメント編集
			接続コンポーネント宣言位置検索
			ブレークポイント設定/解除
【アイコン凡例】 🛑 アブリケーション	■画面構成部品 ■ウィンドウ	フ 📑 メニュー 🗐 バネル 📑 処理部品 📑	(●回面構成) ■複合(処理) ■リモート
実行と実行の設定可)		そ クリア 終了

④ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...] をクリックします。
 起動メソッド (処理)を選びます。
 [メソッド]の ・ をクリックします。
 [単数 0pen 用ファイル選択ダイアログを表示する (Component)] をクリックします。
 引数を設定します。
 説明:親コンポーネント
 取得方法:コンポーネント

コンポーネント:フレーム(ID:1)

設定後、了解ボタンをクリックします。

M t	己動メソッド情報				X
		メソッド 単数O	ipen用ファイル選択ダイアログを表示	≂する(Component) ▼] 🔲 全メソッド対象
NO	型	I.光B月	取得方法	コンポーネント	メソッド/値
(Component	親コンポーネント	コンポーネント	フレーム [ID:1] (KEY:"フレーム1")	-
					了解 取消し

ファイルが選択されたら画像ファイルが読み込まれるように接続します。

- ⑤ 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ファイル選択(ID:3)] コンポーネント上で右クリックー[イベント処理追加]
 [データ選択イベント] とクリックします。
- ⑥ イベントの接続先コンポーネントを選びます。
 左側の[ファイル選択(ID:3)] コンポーネントの[データ選択イベント]上で

右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] – [画像ファイル入力(ID:4)] コンポーネントをクリックします。

⑦ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。 起動メソッド設定画面が表示されます。 起動メソッド(処理)を選びます。 [メソッド]の ←をクリックします。 [ファイル名を指定して画像を読み込む(String)]をクリックします。

引数を設定します。

説明:読み込むファイル名

取得方法:イベント内包 メソッド/値:選択データ

設定後、了解ボタンをクリックします。

🛂 起動メソッド情報				X
	メソッド ファ	■イル名を指定して画像を誘	ほみ込む(String)	ノッド対象
NO 型	記 兑8月	取得方法	コンポーネント	メソッド/値
0 String	読み込むファイル名	イベント内包	-	選択データ
				了解 取消し

⑧ イベント番号を設定します。

[画像ファイル入力(ID:4)] コンポーネントの上で右クリックー [イベント番号設定] - [イベント番号設定] をクリックします。

定常起動のチェックをオフにして [N0:1] をチェックします。

2010日 イベント番号設定	×
「定常起動」(イ	ベント番号追加
NO イベント発生	起動
0キャンセルボタンを押したとき	「お勧する」
1showOpenSingleFile(), showSaveSingleFile()でファイル選択にOKを作	. 📝 起動する 🌒
2showOpenFile(), showSaveFile()でファイル選択後にOKボタンを押した	直通動する
設定 キャンセル	

⑨ ボタンが表示され、ボタンをクリックするとファイル選択の画面が表示することを確認します。
 実行(設定可)で実行します。

ボタン名を [画像ファイル入力] に変更しましょう。



※画像ファイルはまだ表示されません。

<u>Step.3</u>新しいフレームの利用

ファイル選択画面で選んだファイルを新しいフレームに表示します。

これまでのLessonでは1つのフレームを利用していました。ここでは2つ目のフレームを利用します。 複数のフレームを使用するときには、[画面編集] 画面の左側の領域を利用します。この領域には、そのアプ リケーションに準備してあるフレームが表示されます。フレームのIDを確認しながら使います。

ここまでで[フレーム(ID:1)] コンポーネントに [ボタン] コンポーネントが追加されています。ここで は新たに [フレーム] コンポーネントを追加して [イメージビューワー] コンポーネントを追加します。

	🔥 画面編集		
	編集		
	■ フレーム [ID:1]	(KEY:"フレーム1")	フレーム [ID:1] (KEY:"フレーム1")
	L=]イメージビュ-	-ワー [ID:6] (KEY:"イメージビー 画像ファイル人力	
・ *<		📴 画面編集	
Image: Section of the secti		編集	
	_		フレーム [ID:5] (KEY."フレーム5")
		ー ボタン [ID:2] (KEY:"画像ファイル入力ア	
配置 換方向控列 → 図 自動サイズ設定 グリッド開展5	×	ー - ジビューワー (D.6) (KEY*イメージ)	
配置 横方向腔列 → 図 自動サイズ設定 グリッド間隔5			
			配置 横方向盛列 → 図 自動サイズ設定 グリッド間隔5

考え方

1. ボタンとは別のウィンドウ(フレーム)を追加しそのウィンドウに画像を表示する。

準 備	ここでは以下のコンポーネントを追加します。
-----	-----------------------

コンポーネント名	必要数	
■フレーム	1	[画面構成部品]-[ウィンドウ]-[フレーム]
■イメージビューワー	1	[画面構成部品]-[グラフィックス]- [イメージビューワー]

操作

必要なコンポーネントを追加します。
 作業領域で右クリック-[コンポーネント追加]-[画面構成部品]-[ウィンドウ]-[フレーム]、
 作業領域で右クリック-[コンポーネント追加]-[画面構成部品]-[グラフィックス]
 -[イメージビューワー]とクリックします。

画面編集

① 画面を作成します。

画面編集をクリックします。

左側の領域から [フレーム(ID:5)] をクリックします。

ここに [イメージビューワー(ID:6)] コンポーネントを追加します。

[画面編集] 画面上で右クリックー [コンポーネント追加] - [イメージビューワー(ID:6)] と クリックします。



② 追加できたら閉じるをクリックし、ビルダー画面に戻ります。

接続確認

コンポーネント同士の接続を確認します。

ファイルが選択されたら新しいフレームが起動する

接続項目	接続関係
接続元コンポーネント	■ファイル,選択 (ID:2)
(イベント発生コンポーネント)	
発生イベント	データ選択イベント
接続先コンポーネント	■フレーム (ID:5)
起動メソッド	フレームを表示する()
イベント番号	1

ファイルが読み込まれたら新しいフレームに画像が表示される

接続項目	接続関係	
接続元コンポーネント	■画クファイルみも (ID:4)	
(イベント発生コンポーネント)		
発生イベント	データ生成イベント	
接続先コンポーネント	■イメージビューワー(ID:6)	
起動メソッド	イメージデータを設定する(Image)	
<引数>	説明:イメージ	
	取得方法:イベント内包	
	メソッド/値:イベント対象データ	

[フレーム(ID:1)] が閉じる時、イメージデータをクリアする

接続項目	接続関係	
接続元コンポーネント		
(イベント発生コンポーネント)		
発生イベント	アクションイベント	
接続先コンポーネント	■イメージビューワー(ID:6)	
起動メソッド	イメージをクリアする()	

[フレーム(ID:1)]が閉じる時 [フレーム(ID:5)]も同時に閉じる

接続先コンポーネント	■フレーム (ID:5)
起動メソッド	フレームを閉じる()

操作 ファイルが読み込まれたら新しいフレームに画像が表示されるようにしましょう。

――ファイルが選択されたら新しいフレームが起動する―――

- ① イベントの接続先コンポーネントを選びます。
 左側の[ファイル選択(ID:3)] コンポーネントの[データ選択イベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] –
 「フレーム(ID:5)]をクリックします。
- ② 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド設定画面が表示されます。
 起動メソッド(処理)を選びます。
 [メソッド]の
 をクリックします。
 [フレームを表示する()]をクリックします。
 設定後、「ア解ボタンをクリックします。

₩_ 走	己動メソッド情報						x
		メソッド フレームを	〔表示する()		▼ ■ 全メソッド対象		
NO	型	说明	取得方法	コンポーネント		メソッド/値	
						了解 取消	iL

③ イベント番号を設定します。

[フレーム(ID:5)] コンポーネントの上で右クリックー [イベント番号設定] - [イベント番号設定] をクリックします。 定常起動のチェックをオフにして [N0:1] をチェックし設定をクリックします。

-----ファイルが読み込まれたら新しいフレームに画像が表示される------

- ④ 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[画像ファイル入力(ID:4)] コンポーネント上で右クリック- [イベント処理追加]
 「データ生成イベント]とクリックします。
- ⑤ イベントの接続先コンポーネントを選びます。 左側の[画像ファイル入力(ID:4)] コンポーネントの[データ生成イベント]上で 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] – [イメージビューワー(ID:6)]をクリックします。

⑥ 接続したコンポーネントの処理を選びます。
接続したコンポーネントの上で右クリック- [起動メソッド設定...]をクリックします。
起動メソッド(処理)を選びます。
[メソッド]の ◆をクリックします。
[イメージデータを設定する(Image)]をクリックします。
引数を設定します。
説明:イメージ
取得方法:イベント内包
メソッド/値:イベント対象データ

設定後、了解ボタンをクリックします。

💾 起動メソッ	ド情報			X
メソッド イメージデータを設定する(Image)				
NO 型	記 兌8月	取得方法	コンポーネント	メソッド/値
0 Image	イメージ	イベント内包	-	イベント対象データ
	· · · · · · · · · · · · · · · · · · ·	·		
				了解取消し

――ボタンが付いているフレームが閉じたら、[イメージビューアー]に表示されている画像を クリアし、フレームも閉じる――

⑦ イベントの接続先コンポーネントを選びます。
 左側の[フレーム(ID:1)] コンポーネントの[アクションイベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] ー
 [イメージビューワー(ID:6)] をクリックします。

⑧ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド設定画面が表示されます。
 起動メソッド(処理)を選びます。
 [メソッド]の をクリックします。
 [イメージをクリアする()]をクリックします。
 設定後、「「解ボタンをクリックします。

🌇 起	記動メソッド情報			×
		メソッド イメージをクリアする	() 🔹 🔹 全メソッド対象	ŧ
NO	型 説明	取得方法	コンポーネント	メソッド/値
				7輝 取消」

- ④ イベントの接続先コンポーネントを選びます。
 左側の[フレーム(ID:1)] コンポーネントの[アクションイベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 [フレーム(ID:5)] をクリックします。
- ① 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド (処理)を選びます。
 [メソッド]の ・ をクリックします。
 [フレームを閉じる()]をクリックします。
 設定後、「解ボタンをクリックします。
- ① [イメージビューワー(ID:6)] 上の画像をクリアし、[フレーム(ID:5)] コンポーネントを閉じてから アプリケーションを終了するように変更します。
 [イメージビューワー(ID:6)]、[フレーム(ID:5)] コンポーネントと
 [アプリケーション] コンポーネントを入れ替えます。
 [アプリケーション] コンポーネントをドラッグして一番下に移動します。
- ② ここまでの動きを確認します。
 実行(設定可)で実行します。
<u>Step.4 イメージビューワーの設定変更</u>

イメージビューワーの設定を変更しましょう。

- 操作 ウィンドウのサイズ変更に伴って、イメージビューワーのサイズも変更するようにしましょう (画像のサイズは変わりません)。
- ① 画面編集をクリックします。
- ② 左側の領域の [フレーム(ID:5)] をクリックして選択します。
- ③ [フレーム(ID:5)]をクリックし、[配置] [領域配置]に変更します。

	🔐 画面編集	X
Image: Spin Spin Spin Spin Spin Spin Spin Spin	編集	
• • </td <td>B-■ フレーム [ID:1] (KEY:"フレーム1")</td> <td>フレーム (ID:5) (KEY:"フレーム5")</td>	B-■ フレーム [ID:1] (KEY:"フレーム1")	フレーム (ID:5) (KEY:"フレーム5")
	a <u>1 フレーム 10051 (KEY "フレーム57</u> └ <u>1 フレーム 1055 (KEY "フ</u> レー <u>365</u>)	
18./J[Pitht?']	x P	記歴 様方施証列 ・ 2 1 8 8 9 サイズ設定 グリッド間隔5 0 10 20 30 40 50 翻じる (対方施証)
版式方向脉列 体验改变 电影分割放置		182方の起列 <u>峰の起こ</u> 地形分割配置

- 閉じるをクリックします。
- (5) 実行をクリックし、動作確認します。
 ここでは画像ファイル「mz_logo_long.gif」を使用して確認します。

参照:	\mu Tutorial		
	▶ 購買管理	🖻 Lesson12(イメージ表示).mzax	プレビュー
最近使った項	esson04(ウィンドウ)-1.mzax	i Lesson13(拡張版電卓)-1.mzax Lesson13(拡張版電卓)-1a.mzax	
B	💌 Lesson04(ウィンドウ)-2.mzax	Nesson13(拡張版電卓)-2.mzax	
	■ Lesson05(電卓).mzax	📴 Lesson13(拡張版電卓)-3.mzax 🔤 Lesson14(抗理版ガントチャット)-1 mzax	
デスクトップ	Nesson06(住所錄)-2.mzax	■ Lesson14(拡張版ガントチャート)-2.mzax	
	■ Lesson07(フォルダ構成).mzax	■ Lesson15(テーブル機能拡張)-1.mzax	MI M7 Diotform
	■ Lesson07(組織構成).mzax ■ Lesson08(気温グラフ).mzax	🔤 Lesson15(テーブル機能拡張)-2.mzax 醒 Lesson15(テーブル機能拡張)-3.mzax	🛃 IN Z FIALIVII
マイドキュー	Nesson08(気象情報).mzax	醒 Lesson15(テーブル機能拡張)-4.mzax	
	■ Lesson09(人口分布).mzax	■ Lesson15(テーブル機能拡張)-5.mzax	
	Network Construction (デジタル時計).mzax	mz_logo.gif	
コノヒュー ター	🖳 Lesson11(ガントチャート).mzax	mz_logo_long.gif	
	(III	4	
	ファイル名: mz_logo_long.gif		開く

操作 ウィンドウの中で画像をドラックできるようにしましょう。

① [イメージビューワー] コンポーネントの上で右クリックー [属性情報設定...]をクリックします。

WZ Platform アプリケーションビルダー			
ファイル 編集 アブリケーション オブション	ヘルブ		メント行検索 💌
アプリケーション名称			e • • •
■ アプリケーション	アプリケーション開始イベント	フレームを表示する	
KEY : ""	•		10:1 KEY:"フレーム1"
	アクションイベント	イメージをクリアする	
(KET.) D-41		フレームを閉じる	
			ID:5 KEY:"フレーム5"
		アブリケーションを終了する	■ アブリケーション
	アクションイベント		KEY:**
ID:2 KEY:"画像ファイル入力"	•	単数Open用ファイル運転ダイアロジを表示す。	◎ ID:3 ID:3 KEY:"ファイル選択3"
ファイル選択	データ選択イベント	ファイル名を指定して画像を読み込む	 画像ファイル入力 D:4
KEY:"ファイル選択3"			NO:1] KEY: "画像ファイル入力4"
		フレームを表示する	N0:1] ID : 5 KEY: "7 L= /55"
画像ファイル入力	データ生成イベント	イメージデータを設定する	
ID:4 KEY:"画像ファイル入力4"	•		1D:6 KEY:"イメージビューワー6"
ID:6 KEY:"イ イベント処理追加 ♪			
コピー			
CT-	画面構成部品 🗐 ウィンド	ウ 🧐メニュー 🗐パネル 🗐処理部品 🗐複	合(画面構成) 🛑 複合(処理) 🛑 リモート
ペロ111首和REX定 コンポーネント情報表示	可) 画面編集 帳票編集	ロード 挿入 保存 上書き保存	クリア 終了
いしのド記動位置絵奏	/		
インデント			

② [DragEnabled] を [true] にします。

🛂 コンポーネント属性情報		×
ComponentPublicName	V NULL	- ^
ComponentID	6	
ComponentKey	イメージビューワー6 📃 NULL	-
ComponentKeys	日本語: 英語: □ NULL	- 1
AllowRemoteInvocation	🔘 true 🔘 false	
AllowPullTransfer	🔘 true 🔘 false	
AllowPushTransfer	🔘 true 💿 false	E
AutoResize	🔘 true 💿 false	
DragEnabled	Intrue	
MultiLocaleToolTipText	日本語: 英語: INULL	-
HorizontalScrollBarPolicy	30	
VerticalScrollBarPolicy	20	
MaxUnitIncrement	5	
VerticalScale	100	
HorizontalScale	100	
Visible	● true ○ false	
AlianmentX	0.5	-
	設定	

③ [実行] ボタンをクリックし、ウィンドウの中で画像がドラッグできることを確認します。 ここでは画像ファイル「mz_logo_long.gif」を使用して確認します。 ここまで進めるとビルダー上では以下のようになります。

■アプリケーション	アプリケーション開始イベント	フレームを表示する	<u> </u>
KEY:"			ID : 1 KEY : "フレーム1"
<u> </u>	アクションイベント	¬ イメージをクリアする	<u> </u>
ID : 1 KEY : "フレーム1"		ſ	ID : 6 KEY : "イメージビューワー6"
		フレームを閉じる	<u> </u>
			ID:5 KEY:"フレーム5"
		アプリケーションを終了する	■アプリケーション
			KEY:""
ボタン	アクションイベント	単数Open用ファイル選択ダイアログを表示する	ファイル選択
ID:2 KEY:"画像ファイル入力"			ID:3 KEY:"ファイル選択3"
ファイル選択	データ選択イベント	- ファイル名を指定して画像を読み込む	■画像ファイル入力
ID:3 KEY:"ファイル選択3"		[NO:1]	ID:4 KEY:"画像ファイル入力4"
		フレームを表示する	<u> </u>
		[NO:1]	ID : 5 KEY : "フレーム5"
■画像ファイル入力	データ生成イベント	イメージデータを設定する	<u> </u>
ID:4 KEY:"画像ファイル入力4"			ID : 6 KEY : "イメージビューワー6"
<u> </u>			
ID:5 KEY:"フレーム5"			
<u> </u>			
ID:6 KEY:"イメージビューワー6"			

<u>Step.5 複合コンポーネントによる階層化</u>

複合コンポーネントとは、いくつかのコンポーネントとひとまとめにした新たなコンポーネントです。 以下のような場合に便利です。

- 1. 繰り返し使われる操作をまとめておく
- 2. ビルダー上の記述が長くなってしまった場合に整理する
- 3. 機能単位にまとめておき開発作業の効率を上げたい
- 4. 後のメンテナンス時に見やすくしたい

複合コンポーネントは、違う階層にコンポーネントをまとめておく方法です。

🔀 MZ Platform アプリケーションビル	ダー			
ファイル 編集 アプリケーション オコ	プション ヘルプ	コメント	行検索 ▼	
アブリケーション名称			🖻 💼 🔄 😤	
■ アブリケーション	アプリケーション開始イベント	フレームを表示する	1 70-4	
KEY: "	•		ID:1 KEY:"フレーム1"	
■ フレーム	アクションイベント	イメージをクリアする	画像表示コンポーネント	
ID:1 KEY:"フレーム1"	_		ID:7 KEY:**	
		フレームを閉じる	画像表示コンポーネント	
		<u> </u>	ID:7 KEY:**	
		アブリケーションを終了する	アプリケーション	
	L		KEY:"	
ボ タン	アクションイベント	単数Open用ファイル選択ダイアログを表示する	🗐 ファイル選択	
ID:2 KEY:"画像ファイル入力"			ID:3 KEY:"ファイル選択3"	
- ファイル選択	データ選択イベント	ファイル名を指定して画像を読み込む	● 画像表示コンポーネント	
ID:3 KEY:"ファイル違択3"	Ť	[NO:	11 ID:7 KEY:	
		フレームを表示する	● 画像表示コンポーネント	
		[NO:	1] D:7 KEY:**	
画像表示コンポーネント				
KEY:"				
1	r			
	MZ Platform アプリケーションビルタ	-		
	ファイル 編集 アプリケーション オプシ	/ョン ヘルブ	コメント行机	* ****
	コンボーネント名称 画像表示コンボーネ	ント		È 🗟 🖄
	■画像表示コンポーネント			
	ID:7 KEY:***			
[大イコン凡例] 🗐 アプリケ				
	ID:7-1 KEY:"フレーム1"			
	「画像ファイル入力 」 □ □	データ生成イベント	イメージデータを設定する	ゴイメージビューワー
	UE7-2 KEY:"画像ファイル入力2"	•	l	IU:7-3 KEY:"イメージビューワー3"
	(KEY:"イメージビューワー3"			
\				
\				
\				
\				
\backslash				
\				
\				
\backslash	[アイコン凡例] 🛑 アプリケー	ション 📄 画面構成部品 🗐 ウィンドウ 🧾 メニュー	■パネル ■処理部品 ■複合(画面構	成) 🗐 複合(処理) 🛑 リモート
\		(設定可) 画面編集 「雌悪編集」 「ロード」		7 終7
1				·

考え方

[画像ファイル] コンポーネント、[フレーム] コンポーネント、[イメージビューワー] コンポーネント を、以下の手順で複合コンポーネント内にまとめます:

1. [複合コンポーネント] を追加する

- 2. [画像ファイル] コンポーネント、[フレーム] コンポーネント、 [イメージビューワー] コンポーネントを複合コンポーネントに追加する
- 3. 元の階層から [画像ファイル] コンポーネント、[フレーム] コンポーネント、 [イメージビューワー] コンポーネントの3つのコンポーネントを削除する

準備 ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■複合コンポーネント	1	

操作

┣━_」複合コンポーネントを追加しましょう。

- 必要なコンポーネントを追加します。
 作業領域で右クリックー [複合コンポーネント作成] [コンポーネント]を追加します。
- ② 追加した [複合コンポーネント] をダブルクリックします。



 M2 Platform アブリケーション オブション ヘルブ
 コメント行後末...

 コメーキント名称 (No Name>)
 国家

 「い Name>)
 国家

 「「 Name>)
 国家

 「 Name>)
 I Name>)

 [T/1 コンパル州)
 「 T/1 リゾワーション)

 「 Name>)
 I Name>)

 [T/1 コンパル州)
 「 T/1 リゾワーション)

 「 Name>)
 I Name>)

 [T/1 コンパル州)
 「 T/1 リゾワーション)

 「 Name>)
 I Name>)

 [T/1 コンパル州)
 I Name>)

 [T/1 コンパー)</td

③ 複合コンポーネントに名前を付けます。

[コンポーネント名称]の「<No Name>」を消して「画像表示コンポーネント」と入力します。

MZ Platform アノリケーションビルター	
ファイル 編集 ラウケーション オフション マーフ	
10:7	
(KEY: ""	
	- 副パネル 副処理部品 副雑合(画面構成) 副親合(処理) 同リモート
(実行)(実行(設定可)) 画面編集 帳票編集 ロード	挿入 保存 上書き保存 クリア 終了





知っていると便利!

複合コンポーネントの中に入っている時、保存ボタンを押して表示されるダイアログでこの階層のみ ボタンを押すと、複合コンポーネントだけをアプリケーションとは別に保存することができます。 デフォルトでは XML 形式(拡張子:.mzcx)とバイナリ形式(.mzcs)の2種類で保存されます。 この時 MZPlatform のインストールフォルダー¥AP_DATA_COMB フォルダ(デフォルトでは C:¥MZPlatform¥3.0¥AP_DATA_COMB)の下にファイルを保存すると、[複合コンポーネント追加]の際、 ファイルが一覧表示されるようになります。



 ② 保存フォルダに「AP_DATA_COMB」を選択します。フォルダ内にさらに新規フォルダを作成しても 構いません。

🖖 保存		×
保存:	: 🕞 AP_DATA_COMB 🔹 🦻	📂 💷 📰
最近使った項 目	 ▲ グラフ ● チュートリアル ● テーブル ③ 合成モジュール ■ 稼働日時 ● 金型履歴 	
デスクトップ		
コンピュー ター		
	ファイル名: 画像表示コンポーネント	保存
ネットワーク	ファイルタイブ: MZ 複合コンボーネント XML (.mzox) ・	取消し

③ [複合コンポーネント追加]時に内容が一覧表示されます。

「AP_DATA_COMB」フォルダ以外に保存してあるファイルを追加する際には、[ロード...]をクリック します。

	_		
コンボーネント追加 コンボーネントー括追加… 複合コンポーネント作成	•		
複合コンポーネント追加	ſ	グラフ	Þ
貼り付け		チュートリアル テーブル	+
すべて展開 すべて閉じる		プロセス 合成モジュール	+
コンボーネント検索 GUIコンボーネント検索		稼働日時 金型履歴	+ +
コメント行追加 コメント行検索		□- <i>\</i> " ♪	
検索			
コンボーネントID再設定			

<u>Step.6 複合コンポーネントの利用</u>

複合コンポーネントを利用しましょう。

1) 複合コンポーネントの作成

複合コンポーネントの中を作ります。これまで使用していた階層と同じ方法で作成します。

準 備

___ここでは以下のコンポーネントを使用します。

コンポーネント名	必要数	
■複合コンポーネント (画像表示コンポーネント)	(1)	
■フレーム	1	[画面構成部品]-[ウィンドウ]-[フレーム]
■画像ファイル入力	1	[入出力]-[ファイル]-[画像ファイル入力]
■イメージビューワー	1	[画面構成部品]-[グラフィックス]- [イメージビューワー]

操作

① 必要なコンポーネントを追加します。
 作業領域で右クリック-[コンポーネント追加] - [画面構成部品] - [ウィンドウ] - [フレーム]、
 作業領域で右クリック-[コンポーネント追加] - [入出力] - [ファイル] - [画像ファイル入力]、
 作業領域で右クリック-[コンポーネント追加] - [画面構成部品] - [グラフィックス]
 - [イメージビューワー] とクリックします。

画面編集

画面を作成します。

画面編集をクリックします。 [イメージビューワー(ID:7-3)] コンポーネントを複合コンポーネントのフレームに追加します。 [画面編集] 画面上で右クリックー [イメージビューワー(ID:7-3)] コンポーネントと クリックします。

追加できたら閉じるをクリックし、ビルダー画面に戻ります。

接続確認

コンポーネント同士の接続を確認します。

接続は複合コンポーネントの上の階層(前の Step)と同じです。

ファイルが読み込まれたら新しいフレームに画像が表示される

接続項目	接続関係
接続元コンポーネント	■画クファイルみカ (ID:7-2)
(イベント発生コンポーネント)	
発生イベント	データ生成イベント
接続先コンポーネント	■イメージビューワー (ID:7-3)
起動メソッド	イメージデータを設定する(Image)
<引数>	説明:イメージデータ
	取得方法:イベント内包
	メソッド/値:イベント対象データ

操作

」複合コンポーネントの中を作りましょう。

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[画像ファイル入力(ID:7-2)] コンポーネント上で
 右クリックー[イベント処理追加] [データ生成イベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。

左側の [画像ファイル入力(ID:7-2)] コンポーネントの [データ生成イベント] 上で 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] – [イメージビューワー(ID:7-3)] をクリックします。

③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。 起動メソッド設定画面が表示されます。 起動メソッド(処理)を選びます。

[メソッド]の 🚽 をクリックします。

[イメージデータを設定する(Image)] をクリックします。

引数を設定します。

説明 : イメージ

取得方法:イベント内包

メソッド/値:イベント対象データ

設定後、了解ボタンをクリックします。

😼 起動メソッド情報				×	
メソッド イメージデータを設定する(Image)					
NO 型	記印月	取得方法	コンポーネント	メソッド/値	
0 Image	イメージ	イベント内包	-	イベント対象データ	
				了解 取消し	

知っていると便利!

複合コンポーネントの中を作成する際、コンポーネントの[コピー]及び[貼り付け]を使用すると効率 良く作成できます。

- ① 複合コンポーネントにするコンポーネント上で右クリック-[コピー]とクリックします。
- ② [複合コンポーネント]をダブルクリックし、複合コンポーネント内に入ります。
- ③ 作業領域で右クリックー[貼り付け]とクリックします。

2) 複合コンポーネントの中のメソッドを公開する

複合コンポーネントの中に設定したメソッドを上の階層から呼び出します。 上の階層からメソッドを呼び出すには複合コンポーネントの中のメソッドを上の階層に「公開」して使える ようにする必要があります。

「公開」するメソッドは上の階層で必要なものだけを公開します。

ここで必要なのは以下の4つのメソッドであることがわかります。これらを公開します。

MZ Platform アプリケーションビルダー				
ファイル 編集 アプリケーション オプション	・ヘルブ	=×:	ノト行検索 ▼	
アブリケーション名称			<u> </u>	
■ アブリケーション	アプリケーション開始イベント	フレームを表示する	■フレーム ID:1	
	70932442F	イメージをクリアする		
(KET. JD-AI				公開メンツト
		ノレームを開しる	ID:5 KEY:"フレーム5"	
		アプリケーションを終了する	■ アブリケーション	
			KEY:"	
■ ボタン ID:2	709324421	単数Open用ファイル選択ダイアログを表示する	ファイル選択 ID:3	
KEY:"画像ファイル人力"	デーな遅根イベント		KEY:"ファイル違択3"	
ID:3 KEY:"ファイル選択3"		ファイル名を指定して画像を読み込む	NO:11 ID:4 KEY: "画像ファイル入力4"	
		フレームを表示する	 ■ フレーム 	╈ 公開メソッド
			NO:1] D:5 KEY: "フレーム5"	
	データ生成イベント	イメージデータを設定する		
KEY: "画像ファイル入力4"			KEY:"イメージビューワー6"	
10:5 KEY:"71/~45"				
- イメージビューワー				
ID:6 KEY:"イメージビューワー6"				
画像表示コンポーネント				
KEY:"				
【アイコン凡例】 🛑 アブリケーション	ン 📄画面構成部品 🗐 ウィンドウ 🧧	メニュー ■バネル ■処理部品 ■複合	画面構成) 🛑 雑合(処理) 🛑 リモート	
実行 実行 設定	(可) 画面編集 帳票編集 (ロード 挿入 保存 上書き保存	クリア 終了	
L				

操作 複合コンポーネントのメソッドを公開しましょう。

- ① [画面表示コンポーネント]をダブルクリックして複合コンポーネントに入ります。
- ② [画面表示コンポーネント]の複合コンポーネントで[右クリック] [公開メソッド設定]を クリックします。

W MZ Platform アプリケーションビルダー	
ファイル 編集 アプリケーション オプション ヘルプ	コメント行検索 🔹
コンボーネント名称 画像表示コンボーネント	🗈 🖻 🔿 😤
画像な 第二キント 公開メジッド設定 第世情報設定 10.71 第七情報設定 10.71 コンボーネント情報表示 10.73 コンボーネント情報表示 10.73 ソンッド起動位置検索 10.73 ビビー・インージビューワーズ	イメージデータを設定する ロイメージビューワー DP.73 ビマン・イメージビューワープ
【アイコン凡例】 🛑 アブリケーション 🔤 画面構成部品 🛑 ウィンドウ 🚽 メ	ニュー 🗐 バネル 🗐 処理部品 🗐 複合(画面構成) 🗐 複合(処理) 🧐 リモート
実行 実行(設定可) 画面編集 「帳票編集」 ロー	- ド 挿入 保存 上書き保存 クリア 終了

③ 公開メソッド設定の窓が表示されます。

公開するメソッドを選びます。

[フレーム]から選びます。

[フレーム(ID:7-1)]で右クリックー[公開メソッド設定...]をクリックします。



④ [フレームを表示する()] [フレームを閉じる()] を公開します。
 [フレームを表示する()] をクリック、[フレームを閉じる()] を【Shift】+クリックします。
 追加(公開化) をクリックし、設定をクリックします。

19 メソッド選択	×
	□ 全メソッド対象
GUIコンポーネントを追加する(PEGUIComponent)	
タイトルバー文字列を設定する(String)	
フレームを表示する()	
フレームを表示する(Component)	
ノレームを同しる() 前面に表示する()	
背面に表示する()	
追加(公開化) 削余(非公開化)	
公開メソッド一覧	
設定	キャンセル

- ⑤ 公開するメソッドを選びます。
 [画像ファイル入力(ID:7-2)]から選びます。
 [画像ファイル入力(ID:7-2)]で右クリックー[公開メソッド設定...]をクリックします。
- ⑥ [ファイル名を指定して画像を読み込む(String)]をクリックします。
 追加(公開化)をクリックし、設定をクリックします。

🙀 メソッド選択	X
	🔲 全メソッド対象
メソッド一覧	
ファイル名を指定して画像を読み込む(String) 入力可能な画像ファイルの拡張子のリストを取得する()	
入力可能な画像ファイル形式のリストを取得する()	
抽画するコンホーネントを登録する(Component) 最適な画像スケール(%)を計算する()	
現在の画像スケール(%)を取得する()	
画像スケール(%)を設定する(int)	
追加(公開化)	
公開メソッド一覧	
設定	キャンセル

⑦ 公開するメソッドを選びます。

[イメージビューワー(ID:7-3)]から選びます。 [イメージビューワー(ID:7-3)]で右クリックー[公開メソッド設定...]をクリックします。 ⑧ [イメージをクリアする()] をクリックします。
 追加(公開化)をクリックし、設定をクリックします。

₩ メソッド選択	×
	□ 全メソッド対象
メソッド一覧	
イメージをクリアする()	
イメージをフィットする()	
1メーンを取得する() イメージアイコンを設定する(Icon)	
イメージデータを設定する(Image)	
単直方向のイメージ表示比率を設定する() 垂直方向のイメージ表示比率を設定する(int)	
水平方向のイメージ表示比率を取得する()	
水平方向のイメージ表示比率を設定する(int)	
追加(公開化) 削除(非公開化)	
設定	キャンセル





ここまで進めるとビルダー上では以下のようになります。



3) 公開してあるメソッドを上の階層から使用する

複合コンポーネントで公開したメソッドを上の階層から使用します。 現在設定されているメソッドを削除して複合コンポーネントのメソッドに置き換えます。

操作 現在設定されているメソッドを削除しましょう。

右上の[編集サポートボタン]をクリックして1階層上に上がります。



確認

元の階層に戻ります。

₩ MZ Platform アプリケーションビルダー	
	コメント行検索
アプリケーション名称	💼 💼 🙆 🖄
アプリケーション アプリケーション開始イベント	フレームを表示する
	- イメージをクリアする
	フレームを閉じる フレーム
	KEY:"
ボタン アクションイベント D:2 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	単数Open用ファイル選択ダイアログを表示する
KEY: 「国際ファイル人力」" 「ファイル選択 データ選択イベント	
ID:3 KEY:"ファイル選択3"	[N0:1] [N0:1] [KEY: "画像ファイル入力4"
	フレームを表示する [N0:1] [D:5 ⁻ [10:1] [D:5 ⁻ [10:1] [D:5 ⁻ [10:1] [D:5 ⁻
- 画像ファイル入力 データ生成イベント	イメージデータを設定する イメージビューワー
ID:4 KEY: 画像ファイル入力4*	し、6 KEY:"イメージビューワー6"
D:5 KEY:"7U-45"	
(花EY): "イメージビューワー6" 画画像表示コンポーネント	
ID:7:"	
【アイコン凡例】 ■アブリケーション ■画面構成部品 ■ウィ:	ンドウ 🧻メニュー 🗐パネル 🍯 処理部品 🗊 複合(画面構成) 🗐 複合(処理) 🛑 リモート
実行 実行感定可) 画面編集 候票編	集 ロード 挿入 保存 上書き保存 クリア 終了

2 複合コンポーネントになっている接続コンポーネントを変更します。
 [フレーム(ID:1)]と接続されている[イメージビューワー(ID:6)]を
 [画像表示コンポーネント(ID:7)]に変更します。

右側の [イメージビューワー(ID:6)] の上で右クリックー [接続コンポーネント選択] – [画像表示コンポーネント(ID:7)] をクリックします。

W MZ Platform アプリケーションビルダー		
ファイル 編集 アプリケーション オプション ヘルプ	コメント行	〒検索 ▼
アプリケーション名称		n 🖻 🖄 📥
アプリケーション アプリケーション 開始イベント アプリケーション 開始イベント	フレームを表示する	<u> </u>
KEY:"		U:1 КЕҮ:"フレーム1"
■フレーム アクションイベント	イメージをクリアする	
КЕÝ:"70-41"	<u> </u>	接続コンポーネント選択
	アプリケーション [ID:0] (KEY:=)	接続コンポーネント選択
	🧐 フレーム [ID:1] (KEY:"フレーム1")	起動メワッド設定
	ボタン [ID:2] (KEY:"画像ファイル入力")	イベント番号設定)
	🧐 ファイル選択 [ID:3] (KEY:"ファイル選択3")	記動モード
	🧐 画像ファイル入力 [ID:4] (KEY:"画像ファイル入力4")	
	- ジフレーム(ID:51/KEY:"フレーム5")	起動メソッド追加 ・
	イメージビューワー [ID:6] (KEY:"イメージビューワース	=Ľ-
	_ ■ ■像表示コンポーネント [ID:7] (KEY:"")	- 」貼り付け ・
KEY:"ファイル選択3"	[no.	切り取り
	JU 4000190	肖明余
	[NO:1] コメント編集
	イメージデータを設定する	接続コンポーネント宣言位置検索
		ブレークポイント設定/解除
ID:5 KEY:"フレーム5"		
(KEY:"		
【アイコン凡例】 🛑 アプリケーション 📄 画面構成部品 🛑 ウィン	ッドウ 🧐メニュー 🗐バネル 🗐処理部品 🗐複合(画面	構成) 🛑 複合(処理) 🛑 リモート
実行 実行(設定可) 画面編集 帳票編)	集 ロード 挿入 保存 上書き保存 / /	7リア 終了

③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定...] をクリックします。 起動メソッド設定画面が表示されます。 起動メソッド(処理)を選びます。 [メソッド]の ◆ をクリックします。 [イメージをクリアする()] をクリックします。 設定後、 了解ボタンをクリックします。

2011年1月1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日		×
NO型 詳明	メソッド イメージをクリアする() ファイル名を指定して画像を読み込む(String) フレームを表示する() フレームを閉じる()	メソッド/値 7解 取消し

 ④ [フレーム(ID:1)] と接続されている[フレーム(ID:5)]を [画像表示コンポーネント(ID:7)] に変更します。 右側の[フレーム(ID:5)]の上で右クリックー[接続コンポーネント選択] – [画像表示コンポーネント(ID:7)] をクリックします。 ⑤ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド設定画面が表示されます。
 起動メソッド(処理)を選びます。
 [メソッド]の → をクリックします。
 [フレームを閉じる()]をクリックします。
 設定後、「了解ボタンをクリックします。

📙 起動メソッド情報	服		X
NO 型	1%0月	メソッド イメージをクリアする() ファイル名を指定して画像を読み込む(String) フレームを表示する() フレームを閉じる()	メソッド/値
			了解 取消し

- ⑥ [ファイル選択(ID:3)] と接続されている [画像ファイル入力(ID:4)] を [画像表示コンポーネント(ID:7)] に変更します。
 右側の [画像ファイル入力(ID:4)] の上で右クリックー [接続コンポーネント選択] – [画像表示コンポーネント(ID:7)] をクリックします。
- ⑦ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド設定画面が表示されます。
 起動メソッド(処理)を選びます。
 [メソッド]の をクリックします。
 [ファイル名を指定して画像を読み込む(String)]をクリックします。
 説明:読み込むファイル名
 取得方法:イベント内包
 メソッド/値:選択データ
 設定後、了解ボタンをクリックします。

2010年1月1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日		×
NO型 說明	メソッド イメージをクリアする() ファイル名を指定して画像を読み込む(String) フレームを示する() フレームを閉じる()	メソッド/値
		了解 取消し

 ⑧ [ファイル選択(ID:3)] と接続されている[フレーム(ID:5)] を [画像表示コンポーネント(ID:7)] に変更します。 右側の[画像ファイル入力(ID:4)] の上で右クリックー[接続コンポーネント選択] – [画像表示コンポーネント(ID:7)] をクリックします。 ⑨ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。
 起動メソッド設定画面が表示されます。
 起動メソッド(処理)を選びます。
 [メソッド]の ◆ をクリックします。
 [フレームを表示する()]をクリックします。
 設定後、「「解ボタンをクリックします。

🛂 起動メソッド情報		x
NO型 説明	メソッド イメージをクリアする() ファイル名を指定して画像を読み込む(String) フレームを表示する() フレームを開じる()	メソッド/値
		了解 取消し

⑩ 確認します。

実行(設定可)で実行します。

複合コンポーネントを作成する前と同じ動作ができることを確認します。

確認 以下のようになります。

■アプリケーション	アプリケーション開始イベント	フレームを表示する	<u>■フレーム</u>
KEY:""			ID : 1 KEY : "フレーム1"
<u> </u>	アクションイベント	¬ イメージをクリアする	● 画像表示コンポーネント
ID : 1 KEY : "フレーム1"		ſ	ID : 7 KEY : ""
		フレームを閉じる	● 画像表示コンポーネント
			ID:7 KEY:""
		アプリケーションを終了する	■アプリケーション
			KEY:"
ボタン	アクションイベント	単数Open用ファイル選択ダイアログを表示する	ファイル選択
ID:2 KEY:"画像ファイル入力"			ID:3 KEY:"ファイル選択3"
ファイル選択	データ選択イベント	- ファイル名を指定して画像を読み込む	■ 画像表示コンポーネント
ID:3 KEY:"ファイル選択3"		[NO:1]	ID:7 KEY:""
		フレームを表示する	● 画像表示コンポーネント
		[NO:1]	ID:7 KEY:""
画像ファイル入力	データ生成イベント	イメージデータを設定する	<u> </u>
ID:4 KEY:"画像ファイル入力4"			ID : 6 KEY : "イメージビューワー6"
ID:5 KEY:"フレーム5"			
<u> </u> イメージビューワー			
ID : 6 KEY : "イメージビューワー6"			
画像表示コンポーネント			
ID:7 KEY:""			

不要なコンポーネントを削除しましょう。

① コンポーネントを削除します。

操作

[画像ファイル入力(ID:4)]を削除します。

[画像ファイル入力(ID:4)]の上で右クリックー [削除]をクリックします。

MZ Platform アプリケーションビルダー			
ファイル 編集 アブリケーション オブショ:	ン ヘルブ		コメント行検索 🔹
アブリケーション名称			🖻 💼 🚞
■ アプリケーション	アプリケーション開始イベント	フレームを表示する	■ フレーム
KEY:"	•		ID:1 KEY:"フレーム1"
<u>■フレーム</u>	アクションイベント	イメージをクリアする	画像表示コンポーネント
D:1 KEY:"フレーム1"	• T		ID:7 KEY:"
		フレームを開じる	画像表示コンポーネント
			KEY:"
	L	アブリケーションを終了する	■ アブリケーション
			KEY:"
<u> ボタン</u> D:2	アクションイベント	単数Open用ファイル選択ダイアログを表示	Ref a D : 3
KEY: "画像ファイル入力"			KEY:"ファイル選択3"
		ファイル名を指定して画像を読み込む	
(KET:: ファイル選択3:			REY:-
	L	フレームを表示する	[N0:1] D_7
「画像ファイル入力	データ生感イベント	イマニジデニタを設定する	■イメージビューワー
ID:4 KEY: "画像- イベント処理追加	•	1 × 77 × 200E9 0	ID:6 KEY:"イメージビューワー6"
1 70-4			
ID:5 KEY:シレー切り取り			
「イメージ」 削除			
KE 属性情報服改定			
画像表示: コンホーホンド情報表示.			
KEY:== メソッド起動位置検索			
[アイ: インデント	▶ ■画面構成部品 ■ ● ウィンド!	う 🗐メニュー 🗐バネル 🗐処理部品 🥤	■複合(画面構成) 🔳複合(処理) 🛑 リモート
実行実行(設)	定可) 画面編集 帳票編集	ロード 挿入 保存 上書き供	解存 クリア 終了

② [コンポーネントを削除します。よろしいですか?]のメッセージが表示されるので [はい]をクリックします。

WZ Platform アプリケーションビルダー				
ファイル 編集 アプリケーション オプション ヘルプ		コメント行	検索 ▼	
アブリケーション名称				
■ アプリケーション アプリケーション 開始イベント	フレームを表示する		<u>■フレーム</u>	
KEY:"			ID:1 КЕҮ:"フレーム1"	
	□ イメージをクリアする		■ 画像表示コンポーネント ID:7	
<u>[KĒÝ:"フレ−ム1"</u>				
	フレームを閉じる			
	7415	+ 7	RET.	
	「 アプリケーションを除了	10	KEY:"	
ボタン アクションイベント		▼を表示する	🗐 ファイル選択	
D:2 KEY:"画像ファイル入力" 構築			ID:3 KEY:"ファイル選択3"	
	-ネントを削除します。よろしいですか?	み込む	■画像表示コンポーネント ID:7	
KEY:"ファイル選択3"		[NO:1]		
		INO:11	 画像表示コンホーネント ID:7 	
「画像ファイル入力」	/		KEY:**	
10:4 KEY: "画像ファイル入力4"	1 メーンナージを設定す	<u>ə</u>	1D:6 KEY:"イメージビューワー6"	
ID:5 KEY:フレーム5"				
■ 画像表示コンポーネント ID:7				
(KEY:"				
【アイコン凡例】 🛑 アブリケーション 📄 画面構成部品 🛑 ウィン	[アイコン凡州] 🛑 アブリケーション 📄 画面構成部品 📄 ウィンドウ 🦳 メニュー 🗐 パネル 🛑 処理部品 🛑 複合(画面構成) 🛑 複合(処理) 🛑 リモート			
実行 実行(設定可) 画面編集 帳票編集	ロード 挿入 保存	上書き保存 2	リア 終了	

- ③ ①~②を繰り返して
 - [フレーム(ID:5)]、[イメージビューワー(ID:6)]を削除します。

ここまで進めるとビルダー上では以下のようになります。

■アプリケーション	アプリケーション開始イベント	フレームを表示する	■ フレーム
KEY:"			ID:1 KEY:"フレーム1"
■フレーム	アクションイベント	- イメージをクリアする	一 画像表示コンポーネント
ID : 1 KEY : "フレーム1"			ID:7 KEY:""
		フレームを閉じる	一 画像表示コンボーネント
		<u> </u>	ID:7 KEY:""
		アブリケーションを終了する	<mark>■</mark> アプリケーション
			KEY : ""
ボタン	アクションイベント	単数Open用ファイル選択ダイアログを表示する	- ファイル選択
ID:2 KEY:"画像ファイル入力"			ID:3 KEY:"ファイル選択3"
- ファイル選択	データ選択イベント	- ファイル名を指定して画像を読み込む	一 画像表示コンポーネント
ID:3 KEY:"ファイル選択3"	t	[NO:1]	ID:7 KEY:""
		フレームを表示する	画像表示コンポーネント
		[NO:1]	ID:7 KEY:""
■画像表示コンポーネント ID:7 KEY:""			

<u>Step.7</u>機能を追加する

ここまで作成してきたアプリケーションに機能を追加します。

1) 画像をウィンドウの領域表示の大きさに合わせる

画像を表示領域の大きさに合わせることができます。

画像の大きさと表示領域の大きさを比較して画像のスケールを計算し、そのスケールを指定して画像を再表示することができます。

画像のスケールについては [イメージビューワー] コンポーネントが計算の機能を持っています。

完成図
画像をウィンドウの表示領域の大きさに合わせて再表示します。



準 備

ここでは以下のコンポーネントを複合コンポーネントに追加します。

コンポーネント名	必要数	
■パネル	1	[画面構成部品]-[パネル]-[パネル]
■ボタン	1	[画面構成部品]-[ボタン]-[ボタン]

操作

必要なコンポーネントを複合コンポーネントに追加します。
 作業領域(緑)で右クリック- [コンポーネント追加] - [画面構成部品] - [パネル] - [パネル]
 作業領域(緑)で右クリック- [コンポーネント追加] - [画面構成部品] - [ボタン] - [ボタン]
 とクリックします。

画面編集

画面を作成します。
 画面編集をクリックします。
 [パネル] コンポーネントをフレームに追加します。

[配置]を[領域配置]に変更します。

画面編集画面上で右クリックー [コンポーネント追加] - [パネル(ID:7-4)] - [South] とクリックします。

₩ 画面編集	
編集 	フレーム ID:7-11 (KEY:"フレーム1")
ー イメージビューワー [ID:7-3] (KEY:"イメージt	
	(285,280)
	コンポーネント追加 🗐 パネル [[D:7-4] (KEY:"パネル4") North
	コンボーネントー括道加 📄 ボタン [ID:7-5] (KEY:"ボタンの) South 💦
	新規コンボーネントつ話追加 ・ Last
	コンポーネント貼り付け Center
	子コンボーネントー括削除
	周性情報服奠定
	元に戻す コンボーネント削除
	PARO -
	記置 領域配置 → 1 自動サイズ設定 グリッド間隔:5
۰ III + I	0 10 20 30 40 50 閉じる

 ② [ボタン(ID:7-5)] コンポーネントを [パネル(ID:7-4)] コンポーネントに追加します。
 左側の領域を見ると [フレーム(ID:7-1)] コンポーネントに [イメージビューワー(ID:7-3)] コンポ ーネントと [パネル(ID:7-4)] コンポーネントが並列に追加されています。このうち、[パネル(ID:7-4)] コンポーネントに [ボタン(ID:7-5)] コンポーネントを追加します。[パネル(ID:7-4)] コンポーネン ト上で右クリックして、[ボタン(ID:7-5)] コンポーネントを追加します。

隆 画面編集		
編集		
🖃 🗐 フレーム [ID:7-1]	(KEY:"フレーム1")	パネル [ID:7-4] (KEY:"パネル4")
- 1.1110:7-41-4	VEV-1271.4"	
	コンボーネント追加	■ ボタン [ID:7-5] (KEY:"ボタン5")
	コノホーネントー括追加.	
	新規コンポーネント追加	
	新規コンポーネント一括	的
	コンボーネント挿入	•
	コンポーネント削除	
	子コンポーネント一括削り	λ.
	属性情報設定	
		配置 横方向整列 ▼ 図 自動サイズ設定 グリッド問題:5
•	+	0 10 20 30 40 50 閉じる

確認	パラルのトにボタンが	白加されます
(2000000)		
	□ □ フレーム [ID:7-1] (KEY:"フレーム1") □ イメージビューワー [ID:7-3] (KEY:"イメージt □ □ パネル [ID:7-4] (KEY:"パネル4")	バネル/[D:7-4] (KEY:') パネル4")
	ボタン [ID:7-5] (KEY:"ボタン5")	
	4	配置 横方向整列 ▼ 図 自動サイズ設定 グリッド間隔5 0 10 20 30 40 50 閉じる
· 催認	「フレーム(ID:7-1)]を	クリックすろと「イメージビューワー(ID:7-3)]と「パネル(ID:7-4)]
	[x y y (ID: 7-5)] O(⑦アアアア のこしてアーマンニューアー (11) のうこし 「アイロントリント」 位置が確認できます。
	E# Β-■1 フレーム [ID:7-1] (KEY:"フレーム1")	フレーム ID:7-11 (KEY:"フレーム1")
	■ <u>イメージビューワー [ID:7-3] (KEV-</u> (ルージ) ■ <mark>1</mark> パネル [ID:7-4] (KEY:"パネル4")	
		<ポタン>
	۲	配置 補助配置 ▼ 2 自動サイズ設定 グリッド問題5 0 10 20 30 40 50 閉じる

③ [ボタン(ID:7-5)] に「フィット」の文字列を設定します。

[ボタン(ID:7-5)] 上でマウス右クリックし[属性情報設定...]を選択します。コンポーネント情報 設定画面内の[TEXT]の欄に「フィット」と設定します。

設定ボタンを押し設定を確定し、閉じるをクリックし画面編集を終了します。

接続確認

コンポーネント同士の接続を確認します。

画像をイメージビューワーにフィットさせる

接続項目	接続関係	
接続元コンポーネント	$= \pi^2 a \cdot (10.7-5)$	
(イベント発生コンポーネント)	$-\pi \varphi \mathcal{Y} (ID; 7-5)$	
発生イベント	アクションイベント	
接続先コンポーネント	■イメージビューワー (ID:7-3)	
起動メソッド	イメージをフィットする()	

操作

」[フィット] ボタンに機能を割り当てましょう。

――画像をイメージビューワーにフィットさせる―――

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ボタン(ID:7-5)] コンポーネント上で
 右クリック- [イベント処理追加] [アクションイベント] とクリックします。
- 2 イベントの接続先コンポーネントを選びます。
 左側の[ボタン(ID:7-5)] コンポーネントの[アクションイベント]上で
 右クリック-[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリック-[接続コンポーネント選択] [イメージビューワ-(ID:7-3)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリック- [起動メソッド設定...]をクリックします。
 起動メソッド設定画面が表示されます。
 起動メソッド(処理)を選びます。
 [メソッド]の ◆をクリックします。
 [イメージをフィットする()]をクリックします。
 設定後、「「解ボタンをクリックします。

≥	起動メソッド	情報			×
		メンシ	・ド イメージをフィットする	0 全メソッ	ド対象
N	D 型	11.11 11.11	取得方法	コンポーネント	メソッド/値
					「解」取消し

ここまで進めるとビルダー上では以下のようになります。

■ 画像表示コンポーネント ID:7 KEY:""			
<mark> フレーム</mark> D:7-1 KEY:"フレーム1"			
■画像ファイル入力 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-2 ID:7-1 ID:7-2 ID:7-1 ID:7-2 ID:7-1 ID:7-2 ID:7-7-7-7 ID:7-7 ID:7-7 I	データ生成イベント	イメージデータを設定する	ゴイメージビューワー ID:7-3 (1) 2010 - 2010
KEY: "画像ファイル入力2" 「イメージビューワー			(KEY: "1 X = 2 E = - 9 = 3"
ID: 7-3 KEY: "イメージビューワー3"			
<u> バネル</u> D : 7-4 KEY : "パネル4"			
「ボタン 」D: 7-5	アクションイベント	イメージをフィットする	■イメージビューワー ID:7-3
(KEY:"フィット"			KEY:"イメージビューワー3"

2) 画像を拡大・縮小する

画像を拡大・縮小する機能を設定します。10%ずつサイズが変更するようにします。

完成図 画像を拡大・縮小する機能を設定します。



準備 ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ボタン	2	[画面構成部品]-[ボタン]-[ボタン]
■加算(+)	1	[処理部品]-[演算制御]-[加算(+)]
■比較演算(>)	1	[処理部品]-[条件制御]-[比較演算(>)]

操作

複合コンポーネント「画像表示コンポーネント」に必要なコンポーネントを追加します。

① 複合コンポーネントに入ります。
 作業領域(緑)で右クリック-[コンポーネント追加] - [画面構成部品] - [ボタン] - [ボタン]
 と、クリックします。(2回繰り返します)
 作業領域(緑)で右クリック-[コンポーネント追加] - [処理部品] - [演算制御] - [加算(+)]、
 作業領域(緑)で右クリック-[コンポーネント追加] - [処理部品] - [条件制御]
 - [比較演算(>)] とクリックします。

追加した2つのボタンは以下で[+10%]ボタン、[-10%]ボタンになります。

画面編集

画面を作成します。
 画面編集をクリックします。

[パネル] コンポーネントに [ボタン] コンポーネントを2つ追加します。
 [パネル(ID:7-5)] 上で右クリック- [コンポーネント追加] - [ボタン(ID:7-6)]、
 [パネル(ID:7-5)] 上で右クリック- [コンポーネント追加] - [ボタン(ID:7-7)]、
 とクリックします。



接続確認

コンポーネント同士の接続を確認します。

初期状態を登録する

接続項目	接続関係
接続元コンポーネント	- イメージビューロー (ID:7-2)
(イベント発生コンポーネント)	
発生イベント	データ設定イベント
接続先コンポーネント	■加算(+) (ID:7-8)
起動メソッド	数値に変換後、左右オペランドを設定する(String, String)
<引数0>	説明:左オペランド
	取得方法:固定值
	メソッド/値:100
<引数1>	説明:右オペランド
	取得方法:固定值
	メソッド/値:0
イベント番号	1

フィット後の状態を登録する

接続項目	接続関係	
接続元コンポーネント	■ボタン(ID:7-5)	
(イベント発生コンポーネント)		
発生イベント	アクションイベント	
接続先コンポーネント	■加算(+) (ID:7-8)	
起動メソッド	数値に変換後、左右オペランドを設定する(String, String)	
<引数0>	説明:左オペランド	
	取得方法:メソッド戻り値	
	コンポーネント:イメージビューワー(ID:7-3)	
	メソッド/値:垂直方向のイメージ表示比率を取得する	
<引数1>	説明:右オペランド	
	取得方法:固定值	
	メソッド/値:0	

画像を 10%ずつ大きくする

接続項目	接続関係
接続元コンポーネント	$- \pi z (10.7-6)$
(イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント	■加算(+)(ID:7-8)
起動メソッド	数値変換/左右オペランド設定後、演算を行う
	(String, String)
<引数0>	説明:左オペランド
	取得方法:メソッド戻り値
	コンポーネント : 加算(+)(ID:7-8)
	メソッド/値:演算結果(左オペランド+
	右オペランド)を取得する
<引数1>	説明:右オペランド
	取得方法:固定值
	メソッド/値:10

画像を10%ずつ小さくする

接続項目	接続関係
接続元コンポーネント	ーギタン (ID:7_7)
(イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント	■加算(+)(ID:7-8)
起動メソッド	数値変換/左右オペランド設定後、演算を行う
	(String, String)
<引数0>	説明:左オペランド
	取得方法:メソッド戻り値
	コンポーネント : 加算(+) (ID:7-8)
	メソッド/値:演算結果(左オペランド+
	右オペランド)を取得する
<引数1>	説明:右オペランド
	取得方法:固定值
	メソッド/値:-10

左右オペランドを計算する

接続項目	接続関係
接続元コンポーネント (イベント発生コンポーネント)	■加算(+)(ID:7-8)
発生イベント	処理完了イベント
接続先コンポーネント	■比較演算(>) (ID:7-9)
起動メソッド	数値変換/左右オペランド設定後、演算を行う
	(String, String)
<引数0>	説明:左オペランド
	取得方法:イベント内包
	メソッド/値:処理結果データ
<引数1>	説明:右オペランド
	取得方法:固定值
	メソッド/値:0
イベント番号	0

画像スケールを設定する

接続項目	接続関係
接続元コンポーネント	■ 比較演算(\) (ID·7_0)
(イベント発生コンポーネント)	■ L1牧,與身(/) (10.7-9)
発生イベント	処理完了イベント
接続先コンポーネント	■イメージビューワー (ID:7-3)
起動メソッド	setScale (int)
<引数>	説明:スケール(%)
	取得方法:メソッド戻り値
	コンポーネント:比較演算(>)(ID:7-9)
	メソッド/値:左オペランドを取得する
イベント番号	1

操 作

___[+10%] ボタン、[−10%] の機能を設定しましょう。

――初期状態を登録する―――

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[イメージビューワー(ID:7-3)] コンポーネント上で
 右クリックー[イベント処理追加] [データ設定イベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。
 左側の[イメージビューワー(ID:7-3)] コンポーネントの[データ設定イベント]上で
 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] –

[加算(+) (ID:7-8)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。 接続したコンポーネントの上で右クリックー「起動メソッド設定...]をクリックします。 起動メソッド設定画面が表示されます。 起動メソッド(処理)を選びます。 [メソッド]の ▼ をクリックします。 「数値変換後、左右オペランドを設定する(String, String)」をクリックします。 引数を設定します。 <引数0> 説明: 左オペランド 取得方法:固定值 メソッド/値:100 <引数1> 説明:右オペランド 取得方法:固定值 メソッド/値:0 設定後、了解ボタンをクリックします。 ④ イベント番号を設定します。 「加算(+)(ID:7-8)] コンポーネントの上で右クリックー「イベント番号設定]

- [イベント番号設定] をクリックします。

定常起動のチェックをオフにして [N0:1] をチェックし設定をクリックします。

――フィット後の状態を登録する―――

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ボタン(ID:7-5)] コンポーネント上で
 右クリックー[イベント処理追加] [アクションイベント] とクリックします。
- 2 イベントの接続先コンポーネントを選びます。
 左側の[ボタン(ID:7-5)] コンポーネントの[アクションイベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 [加算(+)(ID:7-8)] コンポーネントをクリックします。
- ③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。 起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

[メソッド] の **▼**をクリックします。

[数値変換後、左右オペランドを設定する(String, String)]をクリックします。

引数を設定します。

<引数0>

説明 : 左オペランド

取得方法:メソッド戻り値

コンポーネント:イメージビューワー(ID:7-3)

メソッド/値:垂直方向のイメージ表示比率を取得する()

<引数1>

説明:右オペランド 取得方法:固定値 メソッド/値:0

設定後、了解ボタンをクリックします。

――画像を10%ずつ大きくする―――

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ボタン(ID:7-6)] コンポーネント上で
 右クリックー[イベント処理追加] [アクションイベント] とクリックします。
- イベントの接続先コンポーネントを選びます。
 - 左側の [ボタン(ID:7-6)] コンポーネントの [アクションイベント] 上で 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] – [加算(+) (ID:7-8)] コンポーネントをクリックします。
- ③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。 起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

[メソッド]の ▼ をクリックします。

[数値変換/左右オペランド設定後、演算を行う(String, String)]をクリックします。

引数を設定します。

<引数0>

説明 : 左オペランド

取得方法:メソッド戻り値

コンポーネント:加算(+) (ID:7-8)

メソッド/値:演算結果(左オペランド+右オペランド)を取得する()

<引数1>

説明:右オペランド

取得方法:固定值

メソッド/値:10

設定後、了解ボタンをクリックします。

🛂 起動メソッド情報				×	
	メソッド 数値	変換/左右オペランド設定後、	演算を行う(String,String) ▼ 🗌 全メソッド	対象	
NO 型	記印月	取得方法	コンポーネント	メソッド/値	
0 String	左オペランド	メソッド戻り値	加算(+)[ID:7-8](KEY:"加算(+)8")	演算結果(左オペランド+右オペ…	
1 String	右オペランド	固定値	-	10	
1 String 右オペランド 固定値 - 10					
				了解 取消し	

―画像を10%ずつ小さくする――

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ボタン(ID:7-7)] コンポーネント上で
 右クリックー[イベント処理追加] [アクションイベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。
 左側の[ボタン(ID:7-7)] コンポーネントの[アクションイベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 [加算(+)(ID:7-8)] コンポーネントをクリックします。
- ③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。 起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

[メソッド]の ▼ をクリックします。

[数値変換/左右オペランド設定後、演算を行う(String, String)]をクリックします。

引数を設定します。

<引数0>

説明 : 左オペランド

取得方法:メソッド戻り値

コンポーネント:加算(+) (ID:7-8)

メソッド/値:演算結果(左オペランド+右オペランド)を取得する()

<引数1>

説明:右オペランド

取得方法:固定值

メソッド/値:-10

設定後、了解ボタンをクリックします。

<mark>₩</mark> 2 ‡	己動メソッド情報				×		
	メソッド 数値変換/左右才ペランド設定後、演算を行う(String,String) 👻 🗌 全メソッド対象						
NO	型	i兑8月	取得方法	コンポーネント	メソッド/値		
0	String	左オペランド	メソッド戻り値	加算(+)[ID:7-8](KEY:"加算(+)8")	演算結果(左オペランド+右オペ…		
1	String	右オペランド	固定値	-	-10		
					了解 取消し		

―左右オペランドを計算する―――

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[加算(+)(ID:7-8)] コンポーネント上で
 右クリックー[イベント処理追加] [処理完了イベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。
 左側の[加算(+)(ID:7-8)] コンポーネントの[処理完了イベント]上で
 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。

右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択]-[比較演算(>)(ID:7-9)]コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定...]をクリックします。 起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

[メソッド]の ▼ をクリックします。

[数値変換/左右オペランド設定後、演算を行う(String, String)]をクリックします。

引数を設定します。

<引数0>

説明 : 左オペランド

取得方法:イベント内包

メソッド/値:処理結果データ

<引数1>

説明:右オペランド

取得方法:固定值

メソッド/値:0

設定後、了解ボタンをクリックします。

₩_ 起	副メソッド情報				×
		メソッド 数値変換/左右	5オペランド設定後、氵	演算を行う(String,String) 👻 🗌 全メソッド対象	
NO	型	i说 ⁰ 月	取得方法	コンポーネント	メソッド/値
0	String	左オペランド	イベント内包	-	処理結果データ
1	String	右オペランド	固定値	-	0
					了解 取消し

④ イベント番号を設定します。

[比較演算(>)(ID:7-9)] コンポーネントの上で右クリックー [イベント番号設定] - [イベント番号設定] をクリックします。 定常起動のチェックをオフにして [N0:0] をチェックし設定をクリックします。

――比較結果を受け取る―――

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[比較演算(>)(ID:7-9)] コンポーネント上で
 右クリックー[イベント処理追加] [処理完了イベント] とクリックします。
- 2 イベントの接続先コンポーネントを選びます。
 左側の[比較演算(>)(ID:7-9)] コンポーネントの[処理完了イベント]上で
 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] –
 [イメージビューワー(ID:7-3)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリック- [起動メソッド設定...]をクリックします。
 起動メソッド(処理)を選びます。
 [メソッド]の ◆をクリックします。
 [setSale (int)]をクリックします。
 引数を設定します。
 説明:スケール(%)
 取得方法:メソッド戻り値
 コンポーネント:比較演算(>)(ID:7-9)
 メソッド/値:左オペランドを取得する
 設定後、「了解ボタンをクリックします。

💾 起動メソッド情報	報			×
	עא	ノッド setScale(int)	▼ 全メソッド対象	
NO 型		取得方法	コンポーネント	メソッド/値
0 int		メソッド戻り値	比較演算(>)[ID:7-9](KEY:"比較演算(>)9")	左オペランドを取得する
				「解」「収消し」

④ イベント番号を設定します。

[画像ファイル入力(ID:7-2)] コンポーネントの上で右クリックー[イベント番号設定] - [イベント番号設定]をクリックします。 定常起動のチェックをオフにして [N0:1] をチェックし設定をクリックします。

⑤ [実行(設定可)]で実行し確認します。さらにボタン名を「+10%」「-10%」にそれぞれ変更します。
ここまで進めるとビルダー上では以下のようになります。

■ 画像表示コンボーネント ID:7			
KEY': "			
■ フレーム			
ID : 7-1 KEY : "フレーム1"			
■画像ファイル入力	データ生成イベント	イメージデータを設定する	コイメージビューワー
ID:7-2 KEY:"画像ファイル入力2"			ID : 7-3 KEY : "イメージビューワー3"
「イメージビューワー」	データ設定イベント	数値に変換後、左右オペランドに設定する	🗐 加算(+)
ID : 7-3 KEY : "イメージビューワー3"		[NC):1] ID : 7-8 KEY : "加算(+)8"
「 」バネル			
ID : 7-4 KEY : "パネル4"			
ボタン	アクションイベント	ニー イメージをフィットする	「イメージビューワー」
ID:7-5 KEY:"フィット"		T	ID : 7-3 KEY : "イメージビューワー3"
		 数値に変換後、左右オペランドに設定する	🗐 加算(+)
			ID:7-8 KEY:"加算(+)8"
ボタン	アクションイベント	数値変換/左右オペランド設定後、演算を行う	🗐 加算(+)
ID : 7-6 KEY : "+10%"			ID:7-8 KEY:"加算(+)8"
ボタン	アクションイベント	数値変換/左右オペランド設定後、演算を行う	🗐 加算(+)
ID : 7-7 KEY : "-10%"			ID:7-8 KEY:"加算(+)8"
1 加算(+)	処理完了イベント	数値変換/左右オペランド設定後、演算を行う	🗾 比較演算(>)
ID:7-8 KEY:"加算(+)8"		[NC	2:0] [D:7-9 KEY:"比較演算(>)9"
■比較演算(>)	処理完了イベント	setScale	☐イメージビューワー
ID:7-9 KEY:"比較演算(>)9"		[NC):1] ID : 7-3 KEY : "イメージビューワー3"

Lesson.13 電卓の機能を拡張してみよう

ここではLesson.5で作成した電卓の機能を次のように拡張していきます。

- 1. 2桁以上の桁が扱える
- 2. 四則演算できる

完成図

Lesson.5と比較して電卓の完成図を確認しましょう。



Lesson.5 完成図

🏂 電卓			
		()
7	8	9	÷
4	5	6	×
1	2	3	-
0	•		+
С			=

Lesson. 13 完成図

Step.1 任意桁の計算

1 桁同士の計算ではなく、任意桁同士の計算ができるようにします。

通常の電卓では、任意桁の数値を入力するには数字ボタンを連続して押すと、後から押した数字が表示され ている数字の右側に追加されて複数の桁になっていきます。その後、演算子ボタン(+や-など)やイコー ルボタンを押すと表示されている数値を使って計算します。

このような任意桁を利用するには以下のことが必要になります。

- 1. [数字]ボタンからの一連の入力を文字列の連結として処理する
- 2. 連結された文字列を数値に変換する

これらの処理を可能にするために『変数』コンポーネントを利用します。

1)変数コンポーネント

[変数] コンポーネントとは各種のデータを一時的に保持して(格納して)、保持したままそのデータに対 する操作を行うコンポーネントの総称です。文字データが格納できる変数や、数値データが格納できる変数 など多くの種類があります。

ここでは、① [数字] ボタンから連続して入力された数字を文字列として連結する処理を変数コンポーネ ントで行います。また②連結された文字列を数値に変換する処理も変数コンポーネントで行います。 Lesson. 5 と Lesson. 13 の数値入力フィールドへの数値の表示の違いを確認します。



Lesson.5 では [数字] ボタンを押したら、直接数値が [数値入力フィールド] に表示されました。

Lesson. 13 では [数字] ボタンが押されたら、変数コンポーネントを用いて数字を文字列として連結して (1) 複数の桁にして、(2) 文字列から数値に変換して、その後、数値入力フィールドに表示します。



準 備

考え方

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■文字列格納変数	1	[処理部品]-[変数]-[文字列格納変数]
■ 任 意 精 度 実 数 (BigDecimal)格納変数	1	[処理部品] ー [変数] ー [任意精度実数(BigDecimal)格納変数]

操作

」変数コンポーネントを追加し、変数コンポーネントの名称を変更します。 (完成している Lesson05(電卓). mzax の続きに追加していきます)

必要なコンポーネントを追加します。
 ここでは[文字列格納変数] コンポーネントと[任意精度実数(BigDecimal)格納変数] コンポーネン

トを追加します。 作業領域で右クリックー[コンポーネント追加] - [処理部品] - [変数] - [文字列格納変数]、 作業領域で右クリック- [コンポーネント追加] - [処理部品] - [変数] - [任意精度実数(BigDecimal)格納変数] とクリックします。

② わかりやすいように変数コンポーネントの名前を変更します(名前の変更はしなくても構いません)。
 追加した[文字列格納変数(ID:16)]の上で右クリックー[属性情報設定...]をクリックします。

🖖 MZ Platform アブ	リケーションビルダー			
ファイル 編集 アブ	リケーション オブション へ	レプ	コメン	ト行検索 💌
アプリケーション名科	ホ 電卓アプリケーション			
KEY : " 7"		-		KEY: "数値入力フィールド2" ^
ボ タン		アクションイベント	表示したい文字列を設定する	1数値入力フィールド
LID:10 KEY:"8"				ID:2 KEY: "敏値入力フィールド2"
ボタン		アクションイベント	表示したい文字列を設定する	■数値入力フィールド
ID:11 KEY:"9"				ID:2 KEY: "藪値入力フィールド2"
ボタン		アクションイベント	教値変換/左右オペランド設定後、演算を行う	1 加算(+)
ID:12 KEY:"C"				ID:15 KEY:"加算(+)15"
		アクションイベント	- 難値に変換後 右オペランドに設定する	1 加算(+)
ID:13 KEY:"+"				ID:15 KEY:"加算(+)15"
			· "你你太行马	1加算(+)
			///////////////////////////////////////	ID:15 KEY:"加管(+)15"
				1加算(+)
				ID:15 KEY:"加算(+)15"
「ボタン		アクションイベント		
ID:14 KEY:"="				ID:15 KEY:"加算(+)15"
			Selection and the Sec. St.	
			演算を打つ	ID:15
(一加資(+)		処理案でイベント		
ID:15				
(KEI: /// g(+//5				
			数値に変換後、左右オペランドに設定する	
一方字制物体态制				(KEY:"Jug(+)15"
ID:16	イベント処理追加 ・			
KEY:"文字列				
ID:17	Հուլանու			
(KEY:"任意藉	削除			
1747	属性情期研究	「画面様は部品」(目白マンド	(白 - マニュー - パネル、- 如神部島 - 一雄本語	▼ ■面構成) ■220-(処理) ■11エート
	コンポーネント情報表示			
		画面編集 帳票編集	ロード 挿入 保存 上書き保存	クリア 終了
	メファー については「見糸…			
	インデント			

ComponentKey を「入力文字列格納変数」に変更します。
 入力後[設定]をクリックします。

2000 コンポーネント属性情報		×
Index	0	
String		NULL
Name		NULL
ComponentPublicName		NULL
ComponentID	18	
ComponentKey	入力文字列格納変数	NULL
ComponentKeys	口太語: 英語:	NULL
AllowRemoteInvocation	💿 true 💿 false	
AllowPullTransfer	💿 true 💿 false	
AllowPushTransfer	🔘 true 💿 false	
	設定	

④ ②~③を繰り返して[任意精度実数(BigDecimal)格納変数]も「内部数値格納変数」に変更します。

2) 起動時の変数の初期化

電卓アプリケーションを起動したときに「文字列格納変数(Key:入力文字列格納変数)」に0を設定して初 期化しておきます。

接続確認 コンポーネント同士の接続を確認します。

開始

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■文字列格納変数(ID:16,Key:入力文字列格納変数)
起動メソッド	文字列を設定する(String)
<引数>	説明:文字列
	取得方法:固定值
	メソッド/値:0

操作 アプリケーション開始時に [入力文字列格納変数]を0に初期化します。

- ① イベントの接続先コンポーネントを選びます。
 左側の[アプリケーション] コンポーネントの[アプリケーション開始イベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] ー
 [文字列格納変数(ID:16 Key:入力文字列格納変数)] をクリックします。
- ② 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。
 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
 [メソッド]の
 をクリックします。
 [文字列を設定する(String)] をクリックします。
 引数を設定します。

説明:設定する文字列 取得方法:固定値 メソッド/値:0

設定後、了解ボタンをクリックします。

🌄 趙	副メソッド情報					×
		メソッド 文字列を設定する(Strin	ng)	•	🔲 全メソッド対象	
NO	型	記問	取得方法	コンポーネント	メンシ	・ド/値
0	String	文字列	固定値	-	0	
						了解 取消し

3) 数字ボタンの起動メソッドの変更

Lesson.5では[数字]ボタンを押すと数字が直接[数値入力フィールド]に表示されていましたが、 [数字]ボタンを押すと[文字列格納変数(Key:入力文字列格納変数)]に格納するように変更します。

接続確認 コンポーネント同士の接続を確認します。

数字ボタンを押したら文字列格納変数に格納し、次に押された数字と連結する

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント	■文字列格納変数(ID:16,Key:入力文字列格納変数)
起動メソッド	指定した文字列と連結して置き換える(String)
<引数>	説明:連結する文字列
	取得方法:固定值
	メソッド/値:1

操作

[数字]ボタンの起動メソッドを変更します。

数字ボタンを押したら文字列格納変数に格納し、その変数内で、次に押された数字と連結して いきます。

① イベントの接続先コンポーネントを変更します。

左側の [ボタン(1, ID:3)] コンポーネントから接続されている [数値入力フィールド(ID:2)] メソッド上で右クリックー [接続コンポーネント選択] – [文字列格納変数 (ID:16, Key:入力文字列格納変数]

とクリックします。

② 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定]をクリックします。

起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

[メソッド]の ▼をクリックします。

[指定した文字列と連結して置き換える(String)]をクリックします。

引数を設定します。

説明:連結する文字列

取得方法:固定值

メソッド/値:1

設定後、了解ボタンをクリックします。

型起	記動メソッド情報				×
		メソッド 指定した文字列と連結!	して置き換える(String)	▼ 全メソッド対象	a K
NO	型	記明	取得方法	コンポーネント	メソッド/値
0	String	連結する文字列	固定値	-	1
					了解 取消し

③ ①~②の操作を繰り返して数字ボタン2~9まで設定をします。

4) 文字列データを数値データに変換

[文字列格納変数(Key:入力文字列格納変数)]に格納してあるデータを数値データに変更するため [任意精度実数(BigDecimal)格納変数(Key:内部数値格納変数)]にセットします。

接続確認コンポーネント同士の接続を確認します。

文字列格納変数に格納してあるデータを内部数値格納変数に格納する(数値に変換される)

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	■又于列格衲复致(ID:IO,NEY:八刀又于列格衲复数)
発生イベント	データ設定イベント
接続先コンポーネント	■任意精度実数(BigDecimal)格納変数(ID:17,
	Key:内部数值格納変数)
起動メソッド	数値を文字列で設定する(String)
<引数>	説明:数値の文字列表現
	取得方法:イベント内包
	メソッド/値:イベント対象データ

操作

J [文字列格納変数(Key:入力文字列格納変数)]に格納してあるデータを[任意精度実数 (BigDecimal)格納変数(Key:内部数値格納変数)]にセットします。

- ① 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[文字列格納変数(Key:入力文字列格納変数)(ID:16)] コンポーネント上で右クリックー
 [イベント処理追加] [データ設定イベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。

左側の [文字列格納変数(Key:入力文字列格納変数)(ID:16)] コンポーネントの [データ設定イベント] 上で右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] – [内部数値格納変数(ID:17)] をクリックします。

③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定]をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド(処理)を選びます。
[メソッド]の ◆をクリックします。
[数値を文字列で設定する(String)]をクリックします。
引数を設定します。
説明:数値の文字列表現
取得方法:イベント内包

メソッド/値:イベント対象データ

設定後、了解ボタンをクリックします。

₩_起	動メソッド情報				×
		メソッド 数	値を文字列で設定する(Strin	g) 🔹 🔽 全メソッド対象	
NO	型	i兑0月	取得方法	コンポーネント	メソッド/値
0	String	数値の文字列表現	イベント内包	-	イベント対象データ
		·			
					了解 取消し

5) 数値データを [数値入力フィールド] に表示

[任意精度実数(BigDecimal)格納変数(Key:内部数値格納変数)] にセットされているデータを [数値入力フィールド] に表示します。

接続確認 コンポーネント同士の接続を確認します。

内部数値格納変数に値がセットされたら数値入力フィールドに表示する

接続項目	接続関係
接続元コンポーネント	■任意精度実数(BigDecimal)格納変数(ID:17,
(イベント発生コンポーネント)	Key:内部数值格納変数)
発生イベント	データ設定イベント
接続先コンポーネント	■数値入力フィールド(ID:2)
起動メソッド	表示したい文字列を設定する(String)
<引数>	説明:文字列
	取得方法:イベント内包
	メソッド/値:イベント対象データ

操作

[内部数値格納変数] にセットされているデータを [数値入力フィールド] に表示します。

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[任意精度実数(BigDecimal)格納変数(ID:17, Key:内部数値格納変数)] コンポーネント上で
 右クリック- [イベント処理追加] [データ設定イベント] とクリックします。
- イベントの接続先コンポーネントを選びます。

左側の [任意精度実数(BigDecimal)格納変数 (ID:17, Key:内部数値格納変数)] コンポーネントの [データ設定イベント] 上で右クリックー [起動メソッド追加] とクリックします。 薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] ー [数値入力フィールド(ID:2)] をクリックします。

③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド (処理)を選びます。
[メソッド]の ◆をクリックします。
[表示したい文字列を設定する(String)] をクリックします。
引数を設定します。
説明:文字列
取得方法:イベント内包
メソッド/値:イベント対象データ

設定後、了解ボタンをクリックします。

№₫	₩ 起動メソッド情報						
	メソッド 表示したい文字列を設定する(String)						
NO	型	記印	取得方法	コンポーネント	メソッド/値		
0	String	文字列	イベント内包	-	イベント対象データ		
					了解 取消し		

6) [加算] コンポーネントの処理の変更

Lesson.5では加算コンポーネントの処理結果が直接 [数値入力フィールド] に設定されていましたが、 [文字列格納変数(Key:入力文字列格納変数)] と [任意精度実数(BigDecimal)格納変数(Key:内部数値格 納変数)] を追加したので以下のように修正を行います。

1. 演算を行った後に [文字列格納変数 (Key:入力文字列格納変数)]の文字列を「0」に初期化する

2. [任意精度実数(BigDecimal)格納変数(Key:内部数値格納変数)]の数値を処理結果に置き換える

接続確認

コンポーネント同士の接続を確認します。

[内部数値格納変数]の数値を処理結果に置き換える①

接続項目	接続関係	
接続元コンポーネント		
(イベント発生コンポーネント)		
発生イベント	処理完了イベント	
接続先コンポーネント❶		
※Lesson.5 で設定した [接続先:	■任意精度実数(BigDecimal)格納変数(ID:17,	
数値入力フィールド]を置き換え	Key:内部数值格納変数)	
వ		
起動メソッド	数値を文字列で設定する(String)	
<引数>	説明:数値の文字列表現	
	取得方法:イベント内包	
	メソッド/値:処理結果データ	

演算を行った後に [入力文字列格納変数]の文字列を「0」に初期化する2

接続先コンポーネント ❷ ※新しく追加する	■文字列格納変数(ID:16,Key:入力文字列格納変数)
起動メソッド	文字列を設定する(イベント発生なし)(String)
<引数>	説明:なし
	取得方法:固定值
	メソッド/値:0

操作

[加算(ID:15)] コンポーネントの接続を変更します。

── [任意精度実数(BigDecimal)格納変数(Key:内部数値格納変数)]の数値を処理結果に置き換える●
 ① イベントの接続先コンポーネントを変更します。

- 左側の [加算(ID:15)]] コンポーネントから接続されている右側 [数値入力フィールド(ID:2)] 上で 右クリックー [接続コンポーネント選択] – [任意精度実数(BigDecimal)格納変数 (Key:内部数値格 納変数) (ID:17)] を選択します。
- ② 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。

起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

[メソッド]の ▼をクリックします。

[数値を文字列で設定する(String)]をクリックします。

引数を設定します。

説明:数値の文字列表現

取得方法:イベント内包

メソッド/値:処理結果データ

設定後、了解ボタンをクリックします。

2. 起動メソッド情報						
	メソッド 数値を文字列で設定する(String)					
NO 型	i说B月	取得方法	コンポーネント	メソッド/値		
0 String	数値の文字列表現	イベント内包	-	処理結果データ		
	· · · · · · · · · · · · · · · · · · ·			· · · · · · · · · · · · · · · · · · ·		
				了解 取消し		

──演算を行った後に[入力文字列格納変数]の文字列を「0」に初期化する��──

③ イベントの接続先コンポーネントを選びます。
 左側の[加算(ID:15)] コンポーネントの[処理完了イベント]上で
 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] –
 [文字列格納変数(ID:16 Key:入力文字列格納変数)]をクリックします。

④ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

[メソッド]の ▼ をクリックします。

[文字列を設定する(イベント発生なし)(String)]をクリックします。

引数を設定します。

説明:なし 取得方法:固定値

メソッド/値:0

設定後、了解ボタンをクリックします。

▶ 起動メソッド情報				
メソッド 文字列を設定する(イベント発生なし)(String)				
NO 型	記印月	取得方法	コンポーネント	メソッド/値
0 String		固定値	-	0
				了解 取消し
				「解」「取消し」

<u>Step.2</u>[0]ボタンと[.(小数点)]ボタンを追加

任意桁の表示が可能になったので[0]ボタンと[.(小数点)]ボタンを追加しましょう。

1) [0] ボタンと [. (小数点)] ボタンの追加

完成図

[0] ボタンと [. (小数点)] ボタンを追加します。



準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ボタン	2	[画面構成部品]-[ボタン]-[ボタン]

操作 ボタンを追加しましょう。

必要なコンポーネントを追加します。
 ここでは[ボタン] コンポーネントを2つ追加します。
 作業領域で右クリック-[コンポーネント追加] - [画面構成部品] - [ボタン] - [ボタン]
 とクリックします。(2回繰り返します)

画面編集

 面面を作成します。
 画面編集をクリックします。
 [ボタン] コンポーネントをフレームに追加します。
 [両面編集] 画面上で右クリックー [ボタン] コンポーネントとクリックします。(2回繰り返します)
 追加できたら、ボタンの名前を変更し、体裁を整え閉じるをクリックします。

接続確認

コンポーネント同士の接続を確認します。

[0] ボタン、[.] ボタンの設定

接続項目	接続関係	
接続元コンポーネント	■ ギタン、 (0 ID:10)	
(イベント発生コンポーネント)		
発生イベント	アクションイベント	
接続先コンポーネント	■文字列格納変数(ID:16,Key:入力文字列格納変数)	
起動メソッド	指定した文字列と連結して置き換える(String)	
<引数>	説明:連結する文字列	
	取得方法:固定值	
	メソッド/値:0	

操 作

- ① 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[ボタン(ID:18)] コンポーネント上で右クリック- [イベント処理追加]
 [アクションイベント] とクリックします。
- 2 イベントの接続先コンポーネントを選びます。
 左側の[ボタン(ID:18)] コンポーネントの[アクションイベント]上で
 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] –
 [文字列格納変数(ID:16 Key:入力文字列格納変数)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリック- [起動メソッド設定]をクリックします。
 起動メソッド (処理)を選びます。
 [メソッド]の をクリックします。
 [指定した文字列と連結して置き換える(String)]をクリックします。
 引数を設定します。
 説明:連結する文字列
 取得方法:固定値
 メソッド/値:0
 設定後、了解ボタンをクリックします。

🛂 起動メソッド情報				X	
	メソッド 指定した文字列と連結して置き換える(String) ◆ □ 全メソッド対象				
NO 型	i说 ¹ 月	取得方法	コンポーネント	メソッド/値	
0 String	連結する文字列	固定値	-	0	
				了解 取消し	

④ ①~③の操作を繰り返して(または起動メソッドのコピーを利用して)[.]ボタンを設定します。
 固定値の変更をします。(ボタンの名前に合った固定値を入力します)

2) エラー処理

[. (小数点)] ボタンが追加されたことで、この時点でアプリケーションはボタンの組み合わせでエラー が発生するようになります。

[. (小数点)] ボタンを2回押すとエラーになり、エラーダイアログが表示されます。文字列の数値への 変換が失敗するためです。この状態になった場合、そのまま続けて数字を入力しても[文字列格納変数(Key: 入力文字列格納変数)]に保持されているデータは数値に変換できない状態になっているので、エラーが繰り 返し表示されます。このような場合には、[Cボタン]を押して文字列をクリアすることでその状態を抜ける ことができます。

このようにエラーが発生した場合の処理をビルダー上で記述しておくことができます。エラー処理の方法 としては、[文字列格納変数(Key:入力文字列格納変数)]から最後に入力された文字(2回目の小数点)を 1つ取り除く必要があります。(例: 0.2345. \rightarrow 0.2345)

エラーが発生するのは文字列を数値に変換するときなのでそこにエラー処理を記述しておきます。

[任意精度整数(BigInteger)格納変数]を追加して、エラーが発生したら[文字列格納変数(Key:入力文字 列格納変数)]の最後尾の文字を1文字削除するようにします。

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■整数(BigInteger)格納 変数	1	[処理部品] ー [変数] ー [任意精度 整数(BigInteger)格納変数]

操作

① 必要なコンポーネントを追加します。
 作業領域で右クリックー[コンポーネント追加] - [処理部品] - [変数]
 - [任意精度整数(BigInteger)格納変数] とクリックします。

接続確認

コンポーネント同士の接続を確認します。

エラーが発生したときの文字列の長さを取得する

接続項目	接続関係	
接続元コンポーネント		
(イベント発生コンポーネント)	■又于列格衲复致(ID:I0, Ney:八刀又于列格衲复数)	
発生イベント	データ設定イベント	
接続先コンポーネント❶	■整数(BigInteger)格納変数(ID:20)	
起動メソッド	数値(BigInteger)を設定する(BigInteger)	
<引数>	説明:数値	
	取得方法:メソッド戻り値	
	コンポーネント:	
	文字列格納変数(Key∶入力文字列格納変数)	
	メソッド/値:文字列の長さを取得する	
起動モード	エラー発生時起動	

エラーが発生したときに文字列の最後尾の1文字を削除する2

接続先コンポーネント❷	■整数(BigInteger)格納変数(ID:20)
起動メソッド	値を1減らす()
起動モード	エラー発生時起動

-文字削除した文字列に置き換える3

接続先コンポーネント❸	■文字列格納変数(ID:16,Key:入力文字列格納変数)
起動メソッド	指定インデックス間の部分文字列に置き換える(int, int)
<引数0>	説明:部分文字列の開始インデックス
	取得方法:固定值
	メソッド/値:0
<引数1>	説明:部分文字列の終了インデックス
	取得方法:メソッド戻り値
	コンポーネント : 整数(BigInteger)格納変数
	メソッド/値 : int 数値(BigInteger)を取得する
起動モード	エラー発生時起動

操 作

____エラー処理を追加しましょう。

──エラーが発生したときの文字列の長さを取得する❶──

イベントの接続先コンポーネントを選びます。
 左側の[文字列格納変数(ID:16 Key:入力文字列格納変数)]コンポーネントの[データ設定イベント]

上で右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] – [整数(BigInteger)格納変数(ID:20)] コンポーネントをクリックします。

② 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。

起動メソッド設定画面が表示されます。

起動メソッド(処理)を選びます。

- [メソッド]の ▼ をクリックします。
- [数値(BigInteger)を設定する(BigInteger)]をクリックします。
- 引数を設定します。

説明:数値

取得方法:メソッド戻り値

コンポーネント:文字列格納変数(ID:16 Key:入力文字列格納変数)

メソッド/値:文字列の長さを取得する

設定後、了解ボタンをクリックします。

M_ #	記動メソッド情報				×
		メソッド 数値(BigInteg	er)を設定する(BigInte	ger) 🔹 🔍 🚽 🗸 🗸 🗸 🗸 🗸	
NO	型	説明	取得方法	コンポーネント	メソッド/値
(BigInteger	数値	メソッド戻り値	文字列格納変数 [ID:16] (KEY:"入力文字列格納変数")	文字列の長さを取得する
	·	·			
					了解 取消し

③ エラーの時だけ起動したいので [起動モード]の [エラー発生時起動]を選択します。

WZ Platform アプリケーションビルダー		
ファイル 編集 アプリケーション オプション ヘルプ	コメント行	検索 ▼
アブリケーション名称 電卓アブリケーション		
	数値に変換後、 左オペランドに設定する	KEY:"加寬(+)15"
■ <i>■ ポタン</i> アクションイベント 10:14	数値に変換後、右オペランドに設定する	という。 (EY:"加算(+)15" 加算(+) D::15
(<u>key</u> ;"="	演算 を行う	KEY:"加算(+)15" 加算(+) D:15(+)
■ 加算(+) D(1) 2000	数値を文字列で設定する	NET:加度(*)13 (任意精度実数(BigDecimal)格納変数 10:17 いたいた思想を存在のたまた。
	数値に変換後、左右オペランドに設定する	□ 加算(+) □D:15 レビン・************************************
	文字列を設定する(イベント発生なし)	文字列格納変数 D:16 KFY:"入力文字列終納変数
	数値を文字列で設定する	 任意精度実数(BigDecimal)格納変数 10:17
	数値(BigInteger)を設定する	I整数(BigInteger)格纳变数 ID · 20
□ 任意精度実数(BigDecimal)格納支数 データ数まイベント □ 2:17	表示したい文字列を設定する	接続コンポーネント選択 接続コンポーネント選択 リ
KEY: TRBby@f&hryを数7 ボタン アクションイベント TD:18	指定した文字列と連結して置き換える	起動メソット設定… イベント番号設定 I
KEY:"0" アクションイベント II.119	指定した文字がと連結し Finally起動 エラー発生時起却	起動モード 起動メソット追加
KEY:""" / 整致(BigInteger)格纳实验 ID: 20	L' REVIEW	コピー 18-01510
(KEY: "整額(Biginteger)裕納実験"」 「アイコンR例 ■アブリケーション ■画面構成部品 ■ウィンド	*ウ	4刀り取り 別余
実行 実行 (新定可) 画面編集 (標票編集)	ロード 挿入 保存 上書き保存 ク	・ コメント編集…
<u>L</u>		ブレークポイント設定/解除

―エラーが発生したときに文字列の最後尾の1文字を削除する❷――

- ④ イベントの接続先コンポーネントを選びます。
 左側の[文字列格納変数(ID:16 Key:入力文字列格納変数)]コンポーネントの[データ設定イベント]
 上で右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 「整数(BigInteger)格納変数(ID:20)] コンポーネントをクリックします。
- ⑤ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定] をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド(処理)を選びます。
[メソッド]の → をクリックします。

[値を1減らす()]をクリックします。

設定後、了解ボタンをクリックします。

∿_ 赾	動メソッド情報						x
		メソッド 値を1減らす()		💌 🔲 全メソッド対象		
NO	型	i兑8月	取得方法	コンポーネント		メソッド/値	
						了解即	が し

⑥ エラーの時だけ起動したいので [起動モード]の [エラー発生時起動]を選択しておきます。

――文字削除した文字列に置き換える3―――

- ⑦ イベントの接続先コンポーネントを選びます。
 左側の[文字列格納変数(ID:16 Key:入力文字列格納変数)]コンポーネントの[データ設定イベント]
 上で右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 [文字列格納変数(ID:16 Key:入力文字列格納変数)]コンポーネントをクリックします。
- ⑧ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。

- 起動メソッド(処理)を選びます。
- [メソッド]の ___をクリックします。

[指定インデックス間の部分文字列に置き換える(int, int)]をクリックします。

引数を設定します。

引数0を設定します

説明:部分文字列の開始インデックス

取得方法:固定值

メソッド/値:0

引数1を設定します

説明 : 部分文字列の終了インデックス

取得方法:メソッド戻り値

コンポーネント:整数(BigInteger)格納変数(ID:20)

メソッド/値:数値(BigInteger)を取得する

設定後、了解ボタンをクリックします。

₩_ 赴	己動メソッド情報				×
		メソッド 指定インデックス間の	部分文字列に置き換える	3(int,int)	R.
NO	型	i兑 ¹ 月	取得方法	コンポーネント	メソッド/値
0	int	部分文字列の開始インデックス	固定値	-	0
1	int	部分文字列の終了インデックス	メソッド戻り値	整数(BigInteger)格納変数 [ID:20] (KEY:"整数(BigInteg	数値(BigInteger)を取得する
					了解 取消し

⑨ エラーの時だけ起動したいので [起動モード] を [エラー発生時起動] を選択しておきます。

まとめ

ここまで進めるとビルダー上では以下のようになります。

		フレームを表示する	
【KEY:"電卓アブリケーション" 】			(KEY:"電卓"
	l	文字列を設定する	UD:16
			KEY:"入力文字列格納変数"
	アクションイベント	アブリケーションを終了する	アプリケーション
KEY:" teta	-		KEY:"電卓アプリケーション"
動値入力フィールド			
ID:2 KEY:"数値入力フィールド2"			
() ボタン	アクションイベント	指定した文字列と連結して置き換える	1 文字列格納変数
ID:3 KEY:"1"			10:16 KEY:"入力文字列格納変数"
 「ボタン	アクションイベント	指定した文字列と通結して実ま換える	1 文字列格納変数
ID:4 KEY:"2"	•	THE OR AFTICE HOLE OF A	ID:16 KEY:"入力文字列格納変数"
 –	アクションイベント	化合计学 大学研究 法结正 不要主任 二乙	
		指定した文子列と理結して直き換える	D:16 D:20 EV:"入力文字列終納変對"
(KET. 3	アクションイベント		
ID:6		指定した文字列と連結して置き換える	
	マクション・イッント		
ID:7		指定した文字列と連結して置き換える	
(KEY:"5"			
<u> </u> ホタン ID:8	799324421	指定した文字列と連結して置き換える	UD:16
KEY:"6"			KEY:"入力文字列格納変数"
	アクションイベント	指定した文字列と連結して置き換える	 ・・ ・・ ・・
KEY: "7"			KEY:"入力文字列格納変数"
ボタン	アクションイベント	指定した文字列と連結して置き換える	□ 文字列格納変数
KEY:"8"	•		ID.10 KEY:"入力文字列格納変数"
ボタン	アクションイベント	指定した文字列と連結して置き換える	文字列格納変数
ID:11 KEY:"9"	•		ID:16 KEY:"入力文字列格納変数"
ボタン	アクションイベント	数値変換/左右オペランド設定後、演算を行う	🚺 加算(+)
ID:12 KEY:"C"			ID:15 KEY:"加算(+)15"
ボ タン	アクションイベント	- 数値に変換後 右オペランドを設定する	
ID : 13 KEY : "+"			ID:15 KEY:"加算(+)15"
		湾首奏行ら	□加算(+)
		(風景 こ)) ノ	ID:15 KEY:"加算(+)15"
		あたしつがゆべい ナナマッニン・ドナシャナス	[加宜(+)
	l	────────────────────────────────────	ID:15 KEY:"加管(+)15"
	アクションイベント		
ID:14			
(KET			
	l	<u>演算を行う</u>	
UU具(+) ID:15		数値を文字列で設定する	D:17
KEY:"加算(+)15"			KEY: "内部数值格納変数"
		数値に変換後、左右オペランドを設定する	<u></u>
			KEY:"加算(+)15"
		文字列を設定する(イベント発生なし)	□ 文字列格納変数 ID:16
			KEY:"入力文字列格納変数"
□ 文字列格納変数	データ設定イベント	数値を文字列で設定する	☐ 任意精度実数(BigDecimal)格納変数 ID: 12
IU:16 KEY:"入力文字列格納変数"	• T	-	ID:17 KEY:"内唐陵波值格納変数"
		数値(BigInteger)を設定する	■ 整数(BigInteger)格納変数
		ErrorOnly	ID : 20 KEY : "整数(BigInteger)格納変数"
		値を1減らす	📑 整数(BigInteger)格納変数
		ErrorOnly	ID : 20 KEY : "整数(BigInteger)格納変数"
		指定インデックフ閉の部分文字列に異き換える	1 文字列格納変数
		ErrorOnly	ID:16 KEY:"入力文字列格納変数"
☐ 任意精度実数(BigDecimal)格納索書	次 データ設定イベント	表示」たい立 <u>字別を</u> 設つす z	武力フィールド 武値 大力フィールド
ID:17 KFY:"内部對值格納來對"	•	30110/こいスナクリで設定する	
	アクションイベント	北海に たみつむしい オイト マロキ ほこう	
ID:18 KEY:"0"		指定した义子列と連結して置き換える	ID:16 /EV:"入力士字列線結本##
	アカションチャント		
ID:19		指定した文字列と連結して置き換える	ID:16
			KEY:"八月又子列裕礽发致"
D: 20			
KEY: "整數(BigInteger)格納変数"			

Step.3 四則演算

これまでは[+(加算)]のみでしたので四則演算ができるように設定しましょう。

Lesson.5では演算として[+ボタン]のみが用意されていましたが、ここでは四則演算を実現することを 目的とします。通常の電卓では、演算子ボタン([+ボタン]など)や[=ボタン]を押した際に、直前に押 された演算子ボタンがあればそれに対応する演算処理を実行します。これを実現するためには、直前に押さ れた演算子ボタンの情報をどこかに記憶しておき、最後に演算子ボタンや[=ボタン]が押された際に前も って記憶しておいた演算処理を実行することができなくてはいけません。このような処理の実現方法はいく つか考えられますが、ここでは[算術演算子コンポーネント格納変数]を用いたアプリケーションを作成し ます。

1)変数の利用

- ここでは2つの変数を使用します。
- ・[算術演算子コンポーネント格納変数(入力演算子格納変数)]
- ・[算術演算子コンポーネント格納変数(内部演算子格納変数)]

この変数には直前に押された [演算子ボタン] に対応する [演算子] コンポーネントを格納しておきます。 その後、次の [演算子ボタン] または [=ボタン] が押された際に、変数に格納された [演算子] コンポー ネントが演算を実行し、また次の [演算子] コンポーネントを格納します。 これを繰り返すことで任意回数の四則演算を実現します。

考え方

- [算術演算子コンポーネント格納変数(入力演算子格納変数)]は、演算子が入力される度に[内部演算 子格納変数]に記憶されている演算を実行してから、[内部演算子格納変数]にその[演算子]コンポ ーネントを設定します。
- 2. [算術演算子コンポーネント格納変数(内部演算子格納変数)]は、[演算子]コンポーネントが格納さ れるたびにその左側の数値を[任意精度実数(BigDecimal)格納変数(Key:内部数値格納変数)]から 取得して設定します。

準備 ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■算術演算子コンポーネ ント格納変数	2	[処理部品]ー[変数]ー [算術演算子コンポーネント格納変数]
■サブルーチン	2	[処理部品]-[サブルーチン]-[サブルーチン]
■Null 判定	1	[処理部品] — [条件制御] — [Null 判定]

操作 演算子を格納する変数を用意しましょう。

- ① 必要なコンポーネントを追加します。
 ここでは四則演算で使用する5つのコンポーネントを追加します。
 作業領域で右クリックー[コンポーネント追加] [処理部品] [変数]
 [算術演算子コンポーネント格納変数] とクリックします(2回繰り返します)。
 作業領域で右クリック-[コンポーネント追加] [処理部品] [サブルーチン]
 [サブルーチン] とクリックします(2回繰り返します)。
 作業領域で右クリック-[コンポーネント追加] [処理部品] [条件制御] [Null 判定]
 とクリックします。
- ② コンポーネントの名前を変更しておきます。

 1つめの[算術演算子コンポーネント格納変数]を[入力演算子格納変数]
 2つめの[算術演算子コンポーネント格納変数]を[内部演算子格納変数]
 1つめの[サブルーチン]を[演算実行サブルーチン]
 2つめの[サブルーチン]を[初期化サブルーチン]
 [Null 判定]を[内部演算子 Null 判定]
 と変更します。

接続確認

■___」コンポーネント同士の接続を確認します。

演算実行サブルーチンの処理を呼び出す●

接続項目	接続関係
接続元コンポーネント	■算術演算子コンポーネント格納変数(ID:21,
(イベント発生コンポーネント)	Key:入力演算子格納変数)
発生イベント	データ設定イベント
接続先コンポーネント❶	■サブルーチン(ID:23, Key: 演算実行サブルーチン)
起動メソッド	処理を呼び出す()

入力(格納)された[演算子]を[内部演算子格納変数]に設定する2

接続先コンポーネント🛛	■算術演算子コンポーネント格納変数(ID:22,
	Key∶内部演算子格納変数)
起動メソッド	演算子を設定する(PFArithmeticOperator)
<引数>	説明:算術演算子コンポーネント
	取得方法:イベント内包
	メソッド/値:イベント対象データ

演算子の左側の数値を数値変換する

接続項目	接続関係
接続元コンポーネント	■算術演算子コンポーネント格納変数(ID:22,
(イベント発生コンポーネント)	Key∶内部演算子格納変数)
発生イベント	データ設定イベント
接続先コンポーネント	■算術演算子コンポーネント格納変数(ID:22,
	Key∶内部演算子格納変数)
起動メソッド	演算子の左側の数値を文字列で設定して数値変換する
	(String)
<引数>	説明:演算子の左側の数値の文字列表現
	取得方法:メソッド戻り値
	コンポーネント:任意精度実数(BigDecimal)格納変数
	(ID:17)
	メソッド/値:数値(BigDecimal)を取得する

操 作

─演算実行サブルーチンの処理を呼び出す●──

- ① 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[入力演算子格納変数(ID:21)] コンポーネント上で右クリックー[イベント処理追加]
 [データ設定イベント] とクリックします。
- 2 イベントの接続先コンポーネントを選びます。
 左側の[入力演算子格納変数(ID:21)] コンポーネントの[データ設定イベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] -

[演算実行サブルーチン(ID:23)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。
 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
 [メソッド]の ・ をクリックします。
 [処理を呼び出す()] をクリックします。
 設定後、「了解ボタンをクリックします。

<mark>₩</mark> 2 走	己動メソッド情報					
		メソッド 処理を	呼び出す()	-	🔲 全メソッド対象	
NO	型 説明		取得方法	コンポーネント		メソッド/値

─入力(格納)された[演算子]を[内部演算子格納変数]に設定する��───

- イベントの接続先コンポーネントを選びます。
 - 左側の [入力演算子格納変数(ID:21)] コンポーネントの [データ設定イベント] 上で 右クリック- [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリック- [接続コンポーネント選択] -「内部演算子格納変数(ID:22)] コンポーネントをクリックします。
- ⑤ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。

[メソッド]の ▼をクリックします。

[演算子を設定する(PFArithmeticOperator)]をクリックします。

引数を設定します。

説明:算術演算子コンポーネント

取得方法:イベント内包

メソッド/値:イベント対象データ

設定後、了解ボタンをクリックします。

M _z	起動メソッド情報					×
	メソッド [演算子を設定する(PFArithmeticOperator)				🗌 全メソッド対象	
NO	型	I.兑8月	取得方法	コンポーネント	×.	ノッド/値
	0 PFArithmeticOperator	算術演算子コンボーネント	イベント内包	-	イベ	ント対象データ
						了解 取消し

―演算子の左側の数値を数値変換する――

- ⑥ 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[内部演算子格納変数(ID:22)] コンポーネント上で右クリックー [イベント処理追加]
 [データ設定イベント] とクリックします。
- ⑦ イベントの接続先コンポーネントを選びます。

左側の [内部演算子格納変数(ID:22)] コンポーネントの [データ設定イベント] 上で 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] – [内部演算子格納変数(ID:22)] コンポーネントをクリックします。

⑧ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定] をクリックします。
起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
[メソッド]の ◆ をクリックします。
[演算子の左側の数値を文字列で設定して数値変換する(String)] をクリックします。
引数を設定します。
説明:演算子の左側の数値の文字列表現

取得方法:メソッド戻り値

コンポーネント:任意精度実数(BigDecimal)格納変数(ID:17 Key:内部数値格納変数) メソ<u>ッド</u>/値:数値(BigDecimal)を取得する

設定後、了解ボタンをクリックします。

N	2 起	動メソッド情報				×
メソッド 演算子の左側の数値を文字列で設定して数値変換する(String) 🔹 🗌 全メソッド対象						/ド対象
1	0/	型	i兑8月	取得方法	コンポーネント	メソッド/値
	0	String	演算子の左側の数値の文字列表現	メソッド戻り値	任意精度実数(BigDecimal)格納変数 [ID:17] (KE	Y:"内 数値(BigDecimal)を取得する
Γ						
ŀ						
						了解 取消し

2)サブルーチンとは

1つのイベント処理で複数のメソッドを起動している場合、それらのメソッドを「サブルーチン」として まとめておくと、他のイベント処理で再利用するときなどに便利です。



サブルーチン化していない場合:GUI部品AとGUI部品BでGUI部品①~③の処理をしている これをそれぞれに記述しているためわかりづらい



サブルーチン化している場合:GUI部品①~③の処理をまとめてサブルーチンとすると、 同じ処理の繰り返しをスッキリ記述できる



3)サブルーチンの利用

- ここでは2つのサブルーチンを使用します。
- ・[演算実行サブルーチン]
- ・[初期化サブルーチン]



- 1. [サブルーチン(演算実行サブルーチン)]は、[算術演算子コンポーネント格納変数(Key:内部演算子 格納変数)]と[文字列格納変数(Key:入力文字列格納変数)]がセットされていたらその演算を実行 し、その後[内部演算子格納変数]と[文字列格納変数(Key:入力文字列格納変数)]を初期化すると いう処理を行います。
- 2. [サブルーチン(初期化サブルーチン)]は、アプリケーション起動時と[C(クリア)ボタン]を 押したときに、初期化処理を呼び出します。初期化処理とは[内部演算子格納変数]を初期化(空に) して[文字列格納変数(Key:入力文字列格納変数)]に「0」を設定するということです。

接続確認

」コンポーネント同士の接続を確認します。

変数が NULL(空)でなかったら演算する●

接続項目	接続関係
接続元コンポーネント	■サゴリーチン(ID:22 Kov: 定質字行サブリーチン)
(イベント発生コンポーネント)	■ リラルーナン(ID-23, Ney: 演算実1] リラルーナン)
発生イベント	アクションイベント
接続先コンポーネント❶	■Null判定(ID:25)
起動メソッド	オペランド設定後、演算を行う(Object)
<引数>	説明:オペランド
	取得方法:メソッド戻り値
	コンポーネント:算術演算子コンポーネント格納変数
	(ID:22, Key:内部演算子格納変数)
	メソッド/値:演算子を取得する

初期化する2

接続先コンポーネント 🕑	■算術演算子コンポーネント格納変数(ID:22,
	Key:内部演算子格納変数)
起動メソッド	初期化する()

入力文字列格納変数に0を入れて初期化する

接続先コンポーネント❸	■文字列格納変数(ID:16,Key:入力文字列格納変数)
起動メソッド	文字列を設定する(イベント発生なし)(String)
<引数>	説明:なし
	取得方法:固定值
	メソッド/値:0

内部演算子格納変数を初期化する●

接続項目	接続関係
接続元コンポーネント	
(イベント発生コンポーネント)	■ リ ノルーテン(ID-24, Key: 初期化 リ ノルーテン)
発生イベント	アクションイベント
接続先コンポーネント❶	■算術演算子コンポーネント格納変数(ID:22,
	Key:内部演算子格納変数)
起動メソッド	初期化する()

入力文字列格納変数に0を入れて初期化する2

接続先コンポーネント ❷	■文字列格納変数(ID:16,Key:入力文字列格納変数)
起動メソッド	文字列を設定する(String)
<引数>	説明:設定する文字列
	取得方法:固定值
	メソッド/値:0

操作

─変数が NULL(空)でなかったら演算する**①**───

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[演算実行サブルーチン(ID:23)] コンポーネント上で右クリックー[イベント処理追加]
 [アクションイベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。

左側の[演算実行サブルーチン(ID:23)] コンポーネントの[アクションイベント]上で、 右クリック- [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリック- [接続コンポーネント選択] -[内部演算子 Null 判定(ID:25)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。

起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。

[メソッド]の **・** をクリックします。

[オペランド設定後、演算を行う(Object)]をクリックします。

引数を設定します。

説明 : オペランド

取得方法:メソッド戻り値

コンポーネント:算術演算子コンポーネント格納変数(ID:22, Key:内部演算子格納変数) メソッド/値:演算子を取得する

設定後、了解ボタンをクリックします。

🛃 起動メソッド情報			★ 25 (()) (×
NO 型		- ット	を行う(UDject) ▼ □ 至メワット対象	マリッドノ債
0 Object	オペランド	メワッド戻り値	□ コンボー・コンゴ 算術演算子コンボーネント格納変数 [ID:22] (KEY:"内	演算子を取得する
	,			
				了解 取消し

---初期化する2-----

- ④ イベントの接続先コンポーネントを選びます。
 左側の[演算実行サブルーチン(ID:23)] コンポーネントの[アクションイベント]上で、
 右クリック-[起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリック-[接続コンポーネント選択] [内部演算子格納変数(ID:22)] コンポーネントをクリックします。
- ⑤ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。

[メソッド]の <u></u>をクリックします。

[初期化する()]をクリックします。

設定後、了解ボタンをクリックします。

M	起動メソッド情報					x
		メソッド 初期化する()			🔹 🔳 全メソッド対象	
NC) 型	説明	取得方法	コンポーネント		メソッド/値
L						了解

―変数に0を入れて初期化する❸――

⑥ イベントの接続先コンポーネントを選びます。
 左側の[演算実行サブルーチン(ID:23)] コンポーネントの[アクションイベント]上で、
 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] -

[文字列格納変数(ID:16 Key:入力文字列格納変数)] コンポーネントをクリックします。

⑦ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定] をクリックします。
起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
[メソッド]の ◆ をクリックします。
[文字列を設定する(イベント発生なし)(String)]をクリックします。
引数を設定します。
説明:なし

取得方法:固定值

メソッド/値:0

設定後、了解ボタンをクリックします。

🛂 起動メソッド情報				×
	メソッド 文字列を設定する(イ·	ベント発生なし)(String) –	📃 全メソッド対象
NO 型	記明	取得方法	コンボーネント	メソッド/値
0 String		固定値	-	0
				了解 取消し

──内部演算子格納変数を初期化する**●**───

- ⑧ 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[初期化サブルーチン(ID:24)] コンポーネント上で右クリックー[イベント処理追加]
 [アクションイベント] とクリックします。
- ⑨ イベントの接続先コンポーネントを選びます。

左側の [初期化サブルーチン(ID:24)] コンポーネントの [アクションイベント] 上で、 右クリック- [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリック- [接続コンポーネント選択] -[内部演算子格納変数(ID:22)] コンポーネントをクリックします。

⑩ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。
起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
[メソッド]の ◆ をクリックします。
[初期化する()] をクリックします。
設定後、「了解ボタンをクリックします。

₩2 赴	己動メソッド情報					X
		メソッド 初期化する()			🔹 🔳 全メソッド対象	
NO	쀂	記印	取得方法	コンポーネント		メソッド/値

----変数に0を入れて初期化する**2**-----

- ① イベントの接続先コンポーネントを選びます。
 左側の[初期化サブルーチン(ID:24)] コンポーネントの[アクションイベント]上で、
 右クリック-[起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリック-[接続コンポーネント選択] [文字列格納変数(ID:16 Key:入力文字列格納変数)()] コンポーネントをクリックします。
- 12 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。

[メソッド]の ▼をクリックします。

[文字列を設定する(String)]をクリックします。

引数を設定します。

説明:文字列

取得方法:固定值

メソッド/値:0

設定後、了解ボタンをクリックします。

🛂 起動メソッド情報				×
	メソッド 文字列を読	役定する(String)		▼
NO 型	ii 兌0月	取得方法	コンポーネント	メソッド/値
0 String	文字列	固定値	-	0
				了解 取消し

4) Nu | | 判定

与えられた値が空(Null)かどうかを判定するメソッドを起動されたときに、処理完了イベントから判定結 果をイベント番号で取得します。

接続確認 コンポーネント同士の接続を確認します。

接続項目	接続関係
接続元コンポーネント	■Nu11 判定 (ID:25)
(イベント発生コンポーネント)	
発生イベント	処理完了イベント
接続先コンポーネント❶	■算術演算子コンポーネント格納変数(ID:22,
	Key:内部演算子格納変数)
起動メソッド	演算子の右側の数値を文字列で設定して数値変換する
	(String)
<引数>	説明:演算子の右側の数値の文字列表現
	取得方法:メソッド戻り値
	コンポーネント:任意精度実数(BigDecimal)格納変数
	(ID:17)
	メソッド/値:数値(BigDecimal)を取得する
イベント番号	0(Nullでない場合=内部演算子が設定されている)

演算子の右側の数値を設定する●

演算する❷

接続先コンポーネント🕗	■算術演算子コンポーネント格納変数(ID:22,
	Key∶内部演算子格納変数)
起動メソッド	演算を実行する()
イベント番号	0(Nullでない場合=内部演算子が設定されている)

操作

──内部演算子が設定されていたら演算子の右側の数値を設定する●────

- ① 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[内部演算子 Null 判定(ID: 25)] コンポーネント上で右クリック- [イベント処理追加]
 [処理完了イベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。 左側の[内部演算子 Null 判定(ID: 25)] コンポーネントの[処理完了イベント]上で、 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。

右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] - [内部演算子格納変数(ID: 22)] コンポーネントをクリックします。

③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。

[メソッド] の **・**をクリックします。

[演算子の右側の数値を文字列で設定して数値変換する(String)]をクリックします。 引数を設定します。

説明:演算子の右側の数値の文字列表現

取得方法:メソッド戻り値

コンポーネント:任意精度実数(BigDecimal)格納変数(ID:17)

メソッド/値:数値(BigDecimal)を取得する

設定後、了解ボタンをクリックします。

▶ 起動メソッド情報				×	
メソッド 👔 算子の右側の数値を文字列で設定して数値変換する(String) 🔹 🔍 全メソッド対象					
NO 型	i兑 ¹ 月	取得方法	コンポーネント	メソッド/値	
0 String	演算子の右側の数値の文字列表現	メソッド戻り値	任意精度実数(BigDecimal)格納変数 [ID:17] (KEY:"内	数値(BigDecimal)を取得する	
				了解 取消し	

④ イベント番号を設定します。

[内部演算子格納変数(ID: 22)] コンポーネントの上で右クリックー [イベント番号設定] - [イベント番号設定...]をクリックします。 定常起動のチェックをオフにして[evaluate メソッドで演算結果が false のとき]をチェックします。

―演算する2――

- 「イベントの接続先コンポーネントを選びます。
 左側の[内部演算子 Null 判定(ID: 25)] コンポーネントの [処理完了イベント] 上で、
 右クリック- [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリック- [接続コンポーネント選択] [内部演算子格納変数(ID: 22)] コンポーネントをクリックします。
- ⑥ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定] をクリックします。
起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
[メソッド]の ◆ をクリックします。
[演算を実行する()] をクリックします。
設定後、了解ボタンをクリックします。

挫 赴	己動メソッド情報					×
		メソッド 演算を実行する()			▼ 🔲 全メソッド対象	
NO	型	説明	取得方法	コンポーネント		メソッド/値
						「解」取消し

⑦ イベント番号を設定します。

[内部演算子格納変数(ID: 22)]コンポーネントの上で右クリックー[イベント番号設定] -[イベント番号設定...]をクリックします。

定常起動のチェックをオフにして[evaluate メソッドで演算結果が false のとき]をチェックします。

5) [+ボタン] コンポーネントの起動メソッド変更

[(+) ボタン] コンポーネントにはこれまでは3つの起動メソッドが設定されていましたが、その3つの 作業を[入力演算子格納変数]で行うように設定を変更したので[入力演算子格納変数]へ接続を変更しま す。

起動メソッド1つを変更し、残りの起動メソッド(2つ)は削除します。

接続確認

____コンポーネント同士の接続を確認します。

接続項目	接続関係	
接続元コンポーネント		
(イベント発生コンポーネント)		
発生イベント	アクションイベント	
接続先コンポーネント	■算術演算子コンポーネント格納変数(ID:21,	
	Key:入力演算子格納変数)	
起動メソッド	演算子を設定する(PFArithmeticOperator)	
<引数>	説明:算術演算子コンポーネント	
	取得方法:コンポーネント	
	コンポーネント : 加算(+)(ID:15)	

入力演算子を設定する

操 作

[(+) ボタン] コンポーネントの起動メソッドを変更します。

① イベントの接続先コンポーネントを変更します。
 左側の[(+) ボタン(ID:11)] コンポーネントから接続されている一番上の[加算(+)(ID:15)]
 メソッド上で右クリック-[接続コンポーネント選択] - [入力演算子格納変数(ID:21)] と
 クリックします。

② 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリック- [起動メソッド設定]をクリックします。
 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
 [メソッド]の ◆ をクリックします。
 [演算子を設定する(PFArirhmeticOperator)]をクリックします。
 引数を設定します。
 説明:算術演算子コンポーネント
 取得方法:コンポーネント
 コンポーネント:加算(+)(ID:15)

設定後、了解ボタンをクリックします。

₩ 起動メソッド情報						
メソッド [演算子を設定する(PFArithmeticOperator)						
NO 캪	Ð.	I兑8月	取得方法	コンポーネント		メソッド/値
0 PF	FArithmeticOperator	算術演算子コンポーネント	コンポーネント	加算(+)[ID:15](KEY:"加算(+)15")	-
						了解 取消し
L						

③ 左側の [(+) ボタン] コンポーネントから接続されている [加算(+)(ID:15)] メソッド の上で右クリックー [起動メソッド削除] をクリックします。(もう1つも同様の操作で削除します)

6) [=ボタン]、[加算(+)] コンポーネントの起動メソッド変更・削除

[(=) ボタン] コンポーネントにはこれまでは2つの起動メソッドが設定されていましたが、その2つの 作業を [演算実行サブルーチン] で行うように設定を変更したので [演算実行サブルーチン] へ接続を変更 します。

起動メソッド1つを変更し、残りの起動メソッド(1つ)は削除します。

[加算(+)] コンポーネントにはこれまでは3つの起動メソッドが設定されていましたが、その3つの作業を[任意精度実数(BigDecimal)格納変数(Key:内部数値格納変数)] 1つで行うように設定を変更したので 起動メソッド1つを変更し、残りの起動メソッド(2つ)は削除します。

接続確認

ニュンポーネント同士の接続を確認します。

接続項目	接続関係
接続元コンポーネント (イベント発生コンポーネント)	■ボタン(=)
発生イベント	アクションイベント
接続先コンポーネント	■サブルーチン(ID:23, Key:演算実行サブルーチン)
起動メソッド	処理を呼び出す()

[(=) ボタン] コンポーネントの接続先を変更する
操作 - [(=) ボタン] コンポーネントの起動メソッドを変更― イベントの接続先コンポーネントを変更します。 左側の [(=) ボタン] コンポーネントから接続されている一番上の [加算(+)(ID:15)] メソッド上で右クリックー「接続コンポーネント選択] - 「演算実行サブルーチン(ID:23)] と クリックします。 2 接続したコンポーネントの処理を選びます。 接続したコンポーネントの上で右クリックー「起動メソッド設定」をクリックします。 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。 [メソッド]の ▼をクリックします。 [処理を呼び出す()]をクリックします。 設定後、了解ボタンをクリックします。 🛂 起動メソッド情報 × メソッド 処理を呼び出す() 💽 全メソッド対象 NO 型 説明 取得方法 コンポーネント メソッド/値 了解 取消し

③ 左側の [(=) ボタン] コンポーネントから接続されている [加算(+)(ID:15)] メソッド の上で右クリック- [起動メソッド削除] をクリックします。

―― [加算(+)] コンポーネントの起動メソッドを変更――

 ④ 左側の [加算(+)(ID:15)] コンポーネントから接続されている [任意精度実数(BigDecimal)格納 変数(Key:内部数値格納変数)(ID:17)] のメソッドだけを残して他の2つ([加算(+)(ID:15)] メソッドと [文字列格納変数(ID:16 Key:入力文字列格納変数)] メソッド)を削除します。

7) [アプリケーション]、[Cボタン] コンポーネントの起動メソッド変更・削除

アプリケーション起動時と[C(クリア)ボタン]を押したときに前に定義した初期化処理を実行するように変更します。

[アプリケーション] コンポーネントは起動時に [文字列格納変数(Key:入力文字列格納変数)] を初期化 していましたが、その作業を [初期化サブルーチン] で行うように設定を変更したので [初期化サブルーチ ン] へ接続を変更します。

[C(クリア)ボタン]コンポーネントはイベント発生時に[加算(+)]を初期化していましたが、その 作業を[初期化サブルーチン]で行うように設定を変更したので[初期化サブルーチン]へ接続を変更しま す。

接続確認 コンポーネント同士の接続を確認します。

[アプリケーション] コンポーネントの変数を初期化する

接続項目	接続関係
接続元コンポーネント	■アプリケーション
(イベント発生コンポーネント)	
発生イベント	アプリケーション開始イベント
接続先コンポーネント	■サブルーチン(ID:24, Key:初期化サブルーチン)
起動メソッド	処理を呼び出す()

「C(クリア)ボタン]コンポーネントの変数を初期化する

接続項目	接続関係
接続元コンポーネント (イベント発生コンポーネント)	■ボタン(C)
発生イベント	アクションイベント
接続先コンポーネント	■サブルーチン(ID:24, Key:初期化サブルーチン)
起動メソッド	処理を呼び出す()

操 作

----- [アプリケーション] コンポーネントの起動メソッドを変更------

 ① イベントの接続先コンポーネントを変更します。
 左側の[アプリケーション] コンポーネントから接続されている[文字列格納変数(ID:16 Key:入力 文字列格納変数)]メソッド上で右クリック- [接続コンポーネント選択] - [初期化サブルーチン (ID:24)]とクリックします。

② 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。

起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。

[メソッド]の **▼**をクリックします。

[処理を呼び出す()]をクリックします。

設定後、了解ボタンをクリックします。

M2 i	起動メソッド情報				×
		メソッド 処理す	(呼び出す()	▼ 2メソッド対象	
NO	型	i兑8月	取得方法	コンポーネント	メソッド/値
					了解 取消し

------ [C (クリア) ボタン] コンポーネントの起動メソッドを変更------

① イベントの接続先コンポーネントを変更します。
 左側の[C(クリア)ボタン]コンポーネントから接続されている[加算(+)(ID:15)]
 メソッド上で右クリック- [接続コンポーネント選択] - [初期化サブルーチン(ID:24)] と
 クリックします。

② 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定]をクリックします。
 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
 [メソッド]の ◆をクリックします。
 [処理を呼び出す()]をクリックします。
 設定後、了解ボタンをクリックします。

		メソッド 処理を	呼び出す()		
NO	型	記明	取得方法	コンポーネント	メソッド/値
					了解 取消し

まとめ

ここまで進めるとビルダー上では以下のようになります。

■ アプリケーション	アプリケーション開始イベント	コレールをまデオる	
KFY・"雷貞アゴリケーション"	e	JU-4280733	ID:1 KEY:"雷貞"
		2011年20月0日9	
	マクションズ べつち		
ID:1		アブリケーションを終了する	
(KEY:"電卓"			(KEY:"電卓アブリケーション"
一 数値人力フィールド ID:2			
KEY: "数値入力フィールド2"			
	アクションイベント	指定した文字列と連結して置き換える	■ 文字列格納変数
ID:3 KEY:"1"	•		ID:16 KEY:"入力文字列格納変数"
ボタン	アクションイベント	指定した文字列と連結して置き換える	1 文字列格納変数
ID:4 KEY:"2"		micorext / / Auto care /////	ID:16 KEY:"入力文字列格納変對"
	アクションイベント	****** +	1 文字列格納変数
10:5		指定した文子列と連結して直き換える	
D:6		指定した文字列と連結して置き換える	D:16
KEY:"4"			KEY:"入力文字列格納変数"
	アクションイベント	指定した文字列と連結して置き換える	□ 文字列格納変数
KEY:"5"	-		ID:10 KEY:"入力文字列格納変数"
■ ボタン	アクションイベント	指定した文字列と連結して置き換える	📄 文字列格納変数
ID:8 KEY:"6"			ID:16 KEY:"入力文字列格納変数"
	アクションイベント	化宁 두 중국대왕 계산 중 무士语 유지	① 文字列格納変数
ID:9	•	指定した文子列と理能して直き換える	ID:16 /////:#3.由本宗和#\$?###################################
ID:10		指定した文字列と連結して置き換える	ID:16
KEY:"8"			KEY:"入力文字列格納変数"
	アクションイベント	指定した文字列と連結して置き換える	□ 文字列格納変数
KEY:"9"	-		KEY:"入力文字列格納変数"
■ボタン	アクションイベント	処理を呼び出す	■サブルーチン
ID:12 KEY:"C"			ID:24 KEY:"初期化サブルーチン"
	アクションイベント	演算之本融合士。	算術演算子コンボーネント格納変数
	•	/風見丁で設定する	ID:21 ////:
	アクションイベント		
ID:14		処理を呼び出す	ID: 23
(KEY:"="			(KEY:")賃算実行サブルーチン"
1加算(+)	処理完了イベント	数値を文字列で設定する	任意稽度実数(BigDecimal)格納変数
KEY:"加算(+)15"			KEY:"内部数值格纳変数"
文字列格納変数	データ設定イベント	数値を文字列で設定する	一任意精度実数(BigDecimal)格納変
ID:16 KEY:"入力文字列格納変数"	-		ID:17 KEY:"内部数值格納変数"
		教値(BigInteger)を設定する	🛑 整数(BigInteger)格納変数
		ErrorOnly	ID:20 KEY:"整数(BigInteger)格纳密数"
			wath (BigInteger)格納変對
		1進を15敗わす ErrorOnly	ID:20
			KEY: 登訳(Biginteger)恰相发现
		指定インデックス間の部分文字列に置き換える	U.2.子列格的复数 ID:16
		Endionity	KEY:"入力文字列格納変数"
任意精度実数(BigDecimal)格納変 ID: 12	数 データ設定イベント	表示したい文字列を設定する	■ 数値入力フィールド
KEY:"内部数值格納変数"			KEY: "数値入力フィールド2"
■ ボタン	アクションイベント	指定した文字列と連結して置き換える	1 文字列格納変数
ID : 18 KEY : "0"			ID:16 KEY:"入力文字列格納変数"
	アクショ <u>ン</u> イベント	指定! た立字列と请結! て罢き換うる	(二) 文字列格納変数
ID:19 KEY:" "	•	福足のため子がた座船ので置き決たる	ID:16 KEY:"入力文字列悠纳李数"
ID:20			
NCT: "建筑(Biginteger)格納変数…"	Ele an anna a cara		
 具他演員子コンボーネント格納変 ID:21 	取 テータ設定イベント	処理を呼び出す	ロ ケフルーナン ID:23
KEY:"入力演算子格納変数"			KEY:"演算実行サブルーチン"
		演算子を設定する	算術演算子コンボーネント格納変
			ID:22 KEY:"内部演算子格納変数"
算術演算子コンポーネント格納変	数 データ設定イベント	演算子の左側の静値を文字列で設定して数値変換する	算術演算子コンポーネント格納変数
ID:22 KFY:"内部演算子格纳变数"			ID:22 KEY:"内部演算子格纳变数"
	アクションイベント		
		- オペランド設定後、演算を行う	
(KEY: 演員夫行リブルーナブ			
		初期化する	 員前演員ナコノホーインド恰相変形 ID:22
			KEY:"内部演算子格納変数"
		文字列を設定する(イベント発生なし)	□ 文字列格納変数
			KEY:"入力文字列格納変数"
サブルーチン	アクションイベント	- 初期化する	🧐 算術演算子コンポーネント格納変数
ID:24 KEY:"初期化サブルーチン"	•	4	ID:22 KEY:"内部演算子格納変数"
		十字副を記 キュ	□ 文字列格納変数
		大丁/丁乙訳定りつ	ID:16 KFY:"入力文字別終始來對"
	海理会マイベット		● 管街演管子→、ポー→、「な使力報
ID: 25		演算子の右側の数値を文字列で設定して数値変換する (No.01)	
(KEY:"內部;實算子Null則定"		[N0.0]	KEY:"内部演算子格納変数"
		演算を実行する	 算術演算チョンボーネント格納変数 ID:22
		[NO:0]	KEY:"内部演算子格納変数"

8)四則演算の完成

(-) ボタン、(×) ボタン、(÷) ボタンを追加して四則演算できるようにします。

完成図 四則演算ができる状態にします。



考え方

[(+) ボタン] や [加算(+)] を接続した場合と同じように、[(-) ボタン]、[(×) ボタン]、[(÷) ボタン] が押された情報を [入力演算子格納変数] に渡します。また、[任意精度実数(BigDecimal)格納変数 (Key:内部数値格納変数)] 内で [減算(-)] [乗算(×)] [除算(÷)] を処理します。

準備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ボタン	3	[画面構成部品]-[ボタン]-[ボタン]
■減算(-)	1	[処理部品]-[演算制御]-[減算(-)]
■乗算(×)	1	[処理部品]-[演算制御]-[乗算(×)]
■除算(÷)	1	[処理部品] - [演算制御] - [除算(÷)]

操作

 必要なコンポーネントを追加します。 コンポーネントの数が多いので一括追加します。 作業領域で右クリックー[コンポーネントー括追加]とクリックします。 右側の領域にコンポーネントの分類が表示されるのでここから [画面構成部品] - [ボタン] - [ボタン] を3つ追加、
[処理部品] - [演算制御] - [減算(-)]を1つ追加、
[処理部品] - [演算制御] - [乗算(×)]を1つ追加、
[処理部品] - [演算制御] - [除算(÷)]を1つ追加します。

 コンポーネントの名前を変更しておきます。 1つめの[ボタン(ID:26)]を[-]
 2つめの[ボタン(ID:27)]を[×]
 3つめの[ボタン(ID:28)]を[÷]
 と変更します。

画面編集

画面を作成します。
 画面編集をクリックします。
 [ボタン] コンポーネントを3つフレームに追加し、体裁を整えます。

接続確認

コンポーネント同士の接続を確認します。

[(-) ボタン] [(×) ボタン] [(÷) ボタン] を接続する

接続項目	接続関係
接続元コンポーネント	■ ポタン (ID:26 Kov:-)
(イベント発生コンポーネント)	
発生イベント	アクションイベント
接続先コンポーネント	■算術演算子コンポーネント格納変数(ID:21,
	Key:入力演算子格納変数)
起動メソッド	演算子を設定する(PFArithmeticOperator)
<引数>	説明:算術演算子コンポーネント
	取得方法:コンポーネント
	コンポーネント:減算(-)

[減算(-)] [乗算(×)] [除算(÷)] を接続する

接続項目	接続関係
接続元コンポーネント	■ 油笛 () (ID:20)
(イベント発生コンポーネント)	
発生イベント	処理完了イベント
接続先コンポーネント	■任意精度実数(BigDecimal)格納変数 (ID:17,
	Key:内部数值格納変数)
起動メソッド	数値を文字列で設定する(String)
<引数>	説明:数値の文字列表現
	取得方法:イベント内包
	メソッド/値:処理結果データ

操作

- [(-) ボタン] [(×) ボタン] [(÷) ボタン] を接続する――

- ① 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[(-) ボタン(ID:26)] コンポーネント上で右クリック---[イベント処理追加]
 ---[アクションイベント] とクリックします。
- 2 イベントの接続先コンポーネントを選びます。
 左側の[(-) ボタン(ID:26)] コンポーネントの[アクションイベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] –
 [入力演算子格納変数(ID:21)] コンポーネントをクリックします。
- ③ 接続したコンポーネントの処理を選びます。
 - 接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。
 - 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
 - [メソッド]の ▼ をクリックします。
 - [演算子を設定する(PFArithmeticOperator)]をクリックします。
 - 引数を設定します。
 - 説明:算術演算子コンポーネント
 - 取得方法:コンポーネント
 - コンポーネント:減算(-)(ID:29)

設定後、了解ボタンをクリックします。

🛂 起動メソッド情報				×
メソッド [演算子を設定する(PFArithmeticOperator)				
NO 型	記印	取得方法	コンポーネント	メソッド/値
0 PFArithmeticOperator	算術演算子コンポーネント	コンボーネント	減算(-) [ID:29] (KEY:"減算(-)29")	-
				了解 取消し

④ ①~③の操作を繰り返して、[(×) ボタン] [(÷) ボタン] を設定します。
 メソッド引数の [コンポーネント] はそれぞれの演算によって異なります。

―― [減算 (-)] [乗算 (×)] [除算 (÷)] を接続する――

- ⑤ 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の [減算(-)(ID:29)] コンポーネント上で右クリック- [イベント処理追加]
 [処理完了イベント] とクリックします。
- ⑥ イベントの接続先コンポーネントを選びます。
 左側の[減算(-)(ID:29)] コンポーネントの[処理完了イベント]上で
 右クリックー[起動メソッド追加]とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。

右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] – [任意精度実数(BigDecimal)格納変数 (ID:17 Key:内部数値格納変数)] コンポーネントをクリック します。

⑦ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。

[メソッド]の ▼をクリックします。

[数値を文字列で設定する(String)]をクリックします。

引数を設定します。

説明:数値の文字列表現

- 取得方法:イベント内包
- メソッド/値:処理結果データ

設定後、了解ボタンをクリックします。

Mz	起動メソッド情報				×
		メソッド 数値を文字	列で設定する(String)	▼ 全メソッド対象	
N) 型	i兑8月	取得方法	コンポーネント	メソッド/値
	0 String	数値の文字列表現	イベント内包	-	処理結果データ
					了解 取消し

- ⑧ ⑤~⑦を繰り返して [乗算(×)(ID:30)] コンポーネント、[除算(÷)(ID:31)] コンポーネントに 接続します。
- ⑨ 確認します。

実行(設定可)で実行します。 電卓アプリケーションで四則演算できることを確認します。

保存します。

(次の複合コンポーネントで利用するので必ず保存します)

ここまで進めるとビルダー上では以下のようになります。

アプリケーション	アプリケーション開始イベント	- フレームを表示する	1 71-4
 KEY:"電卓アブリケーション"	——————————————————————————————————————		ID:1 KEY:"電卓"
		処理を呼び出す	
	l		ID:24 KEY:"初期化サブルーチン"
[] JU-4	アクションイベント	アポリケーションを終フする	■ アプリケーション
 ID:1 KFY·"雷卣"	•	アンリケーションで11213	
「数値入力フィールド			
ID:2 KEY:"数値入力フィールド?"			
	アクションイベント	北安」た安安和と遠純」が異きゆうる	1 文字列格納変数
ID:3 KEY:"1"	•	1年しに文十列と連結して直さ換える	
■ ボタン	アクションイベント	北宗」た 古字列と 遠結」 字 里きゆう る	□ 文字列格納変数
ID:4 KEY:"2"	•	THE OR XTYPE HED CHERKING	D:16 KEY:"入力文字列格納変数"
■ ボタン	アクションイベント	北宗」た 古字列と 遠結」 字異き換える	1 文字列格納変数
ID:5 KEY:"3"	•	<u>指進した文十列と連結して直き換える</u>	■ ID:16 KFY:"入力文字列格納変数"
	アクションイベント	北合にたみ宮町には狭いて黒きゆうる	
ID:6 KEY:"4"	•	1年しにスナ州と理転して直さ挟える	————————————————————————————————————
I ボタン I	アクションイベント	七字」た 古字別とはは、 ケ里キ焼 うえ	□ 文字列格納変数
ID:7 KEY:"5"	•	<u> </u>	ID:16 KEY:"入力文字列格纳密勒"
	アクションイベント		
ID:8 KEY:"6"		指正しに丈子列と理結して直き換える	ID:16 KFY:"入力文字列終納來逝"
	アクションイベント	北空 두 구슬지요 'ㅋ?ㅋ 가 편 キ 슈 드 기	
ID:9 KEV:"7"	•••••	指定した文子列と連結して直き換える	
	アクションイベント		
ID:10 KEY:"8"	•••••	指定した文子列と連結して直き換える	ID:16 I/FY:"入力文字列終納來對"
	アクションイベント	水合 나 구슬지 나 날카 가 찍 부산 등 기	
ID:11 KEY:"9"	•••••	指定した文子列と連結して置き換える	ID:16 KFY・"入力文字列格納変数"
	アクションイベント	あり 三田 大口ボスドリレーナ	
ID:12 KEY:"C"	•	処理を呼び出す	
	アクションイベント		
ID:13 KEY:"+"	• • • • • • • • • • • • • • • • • • • •		□D:21 KEY:"入力演算子格纳变数"
■ボタン	アクションイベント	· · · · · · · · · · · · · · · · · · ·	
ID:14 KEY:"="	•••••	処理を呼び出す	ID:23 KEV:**家質実行サブルーチン*
[加賀(+)	処理完了イベント		「任意精度実数(BigDecimal)格納家族
ID:15 KEY:"加賀(+)15"		戦闘を又子列で設定する	ID:17 KFY:"内部對值格納密對"
□ 文字列格納変数	データ設定イベント		□任意精度実数(BigDecimal)格纳密書
ID:16 KFY:"入力文字列終納変数"		3 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	ID:17 KFY:"内部對值格納密對"
		また/方 /口: a line オーラルニュート マ	■整数(BigInteger)格納変数
	-	<u> </u>	ID:20 KEV:"繁雄(BigInteger)核纳密数"
		its to all the set	<u> 整数(BigInteger)格納変数</u>
		IEで10時のす	ID:20 KEY:"整款/Rightegen)格纳密数 "
		他告了、二、占つ明示が八天告知一要主体こう	
	l	<u>10</u> 定1フナックス間の部分文子別に直さ換える ErrorOnly	ID:16 KFY·"入力文字列格纳密数"
☐ 任意精度実数(BiαDecimal)格納密書	ダンデータ設定イベント		
ID:17 KEY:"内部對值格納麥對"	•	教示したい文子列を説定する	ID:2 KEY:"数値入力フィールド?"
	アクションイベント	化合正 七 大合理(1), '#4+, 四 書 4	
ID:18 KEY:"0"		指正したメチ列と連結して直さ換える	ID:16 KFY·"入力文字別終納來逝"
	アクションイベント	他告日 たみ合理し さかし マ 聖主後 ニッ	
ID:19 KEV:" "		指定した文子列と連結して置き換える	■ <u> </u>
「整数(BigInteger)格納変数			
ID:20 KEV:"繁新(BigInteger)块纳恋选标。"			

算術演算子コンポーネント格納変数	データ設定イベント	_ 処理を呼び出す	<mark> </mark>
ID:21 KEY:"入力演算子格納変数"	t		ID:23 KEY:"演算実行サブルーチン"
		演算子を設定する	算術演算子コンポーネント格納変数
			ID:22 KEY:"内部演算子格納変数"
■算術演算子コンポーネント格納変数	データ設定イベント	演算子の左側の数値を文字列で設定して数値変換する	算術演算子コンボーネント格納変数
ID:22 KEY:"内部演算子格納変数"	•		ID:22 KEY:"内部演算子格納変数"
■サブルーチン	アクションイベント	オペランド設定後、演算を行う	Null判定
ID:23 KEY:"演算実行サブルーチン"	• · · · · · · · · · · · · · · · · · · ·	ſ	ID:25 KEY:"内部演算子Null判定"
		初期化する	■算術演算子コンポーネント格納変数 10:22
			ID:22 KEY:"内部演算子格納変数"
		文字列を設定する(イベント発生なし)	■ 文字列格納変数
			ID.10 KEY:"入力文字列格納変数"
■サブルーチン ID:24	アクションイベント	初期化する	 算術演算子コンポーネント格納変数 ID:22
KEY:"初期化サブルーチン"			KEY:"内部演算子格納変数"
		文字列を設定する	■ 文字列格納変数 ID:16
			KEY:"入力文字列格納変数"
■ Null判定 ID:25	処理完了イベント	□ 演算子の右側の数値を文字列で設定して数値変換する	□ 算術演算子コンボーネント格納変数 ID:22
KEY:"内部演算子Null判定"		[IO:U]	KEY:"内部演算子格納変数"
		演算を実行する	■算術演算子コンポーネント格納変数 ID:22
		[10:0]	KEY:"内部演算子格納変数"
<u> ボタン</u> D:26	アクションイベント	演算子を設定する	_] 算術演算子コンポーネント格納変数 ID : 21
KEY:"-"			KEY:"入力演算子格納変数"
<u> ボタン</u> D: 27	アクションイベント	演算子を設定する	 算術演算子コンボーネント格納変数 ID:21
KEY:"×"			KEY:"入力演算子格納変数"
<u> ホタン</u> D:28	アクションイベント	演算子を設定する	■」算術演算子コンボーネント格納変数 ID:21
KEY:"+"			
■) 版算(-) ID : 29	処理完了イベント	数値を文字列で設定する	1任意精度実破(BigDecimal)格納変数 ID:17
KEY:"減算(-)29"			
■ 乗具(×) D_: 30	処理完了イベント	数値を文字列で設定する	UD:17.
KEY:"兼算(×)30"	hn 700		
■「际算(+) ID:31	処理完了イベント	数値を文字列で設定する	UID:17
			LKEY:"内部数值格納変数"

<u>Step.4 複合コンポーネントによる階層化</u>

複合コンポーネントとは、いくつかのコンポーネントとひとまとめにして新たなコンポーネントを定義す ることです。以下のような場合に便利です。

- 1. 繰り返し使われる操作をまとめておく
- 2. ビルダー上の記述が長くなってしまった場合に整理する
- 3.機能単位にまとめておき開発作業の効率を上げたい
- 4. 後のメンテナンス時に見やすくしたい

複合コンポーネントは、違う階層にコンポーネントをまとめておく方法です。

W MZ Platform アブリケーションビルダー				
ファイル 編集 アブリケーション オブション ヘルブ コメント行検索・				
アプリケーション名称 電卓アブリケーション				
	アクションイベント	数字を文字列として追加する	■電卓モデル □ 10 : 32 □ 10 : 32	
KEY:"8"	•		KEY:"	
	アクションイベント	数字を文字列として追加する	電卓モデル ID: 32 ID	
KEY:"9"	•		KEY:"	
	アクションイベント	数字を文字列として追加する	電卓モデル D:32	
			KEY:"	
D:19	アクションイベント	数字を文字列として追加する	■1電卓モデル ID:32	
KEY				
<u> ボタン</u> D:12	アクションイベント	初期化処理を呼び出す		
KEY:"C"				
ID:13		+押下処理を呼び出す	<u>■ 単単モナル</u> ID_: 32	
(KEY:"+"	700-006			
		=押下処理を呼び出す	D: 32 (の): 32	
	アクションイベント			
		- 押ト処理を呼び出す		
	アクションイベント			
10:27 IVEV:"x"		×押ト処理を呼び出す		
	アクションイベント	- 1811年7月1日日 - 1811年1日	■ 電卓モデル	
ID : 28 KEY : "+"	• • • • • • • • • • • • • • • • • • • •	州下処理を呼び出す	ID : 32 KEY : ""	
(一)電卓モデル		まこしたいかつ利を設定する	製値入力フィールド	
ID: 32 KEY: ***		30/10/2010/17/230/29/3	ID:2 KEY: "数値入力フィールド2"	
[アイコン凡例] 🛑 アプリケーション	ン 📄 画面構成部品 🗐 ウィンドウ 🧧	メニュー 🗐バネル 🗐処理部品 🗐剤	复合(画面構成) 🛑 複合(処理) <mark>-</mark>リモート	
実行 実行 実行 (設定	(1) 画面編集 帳票編集 (コード 挿入 保存 上書き保存	そ クリア 終了	
	WZ Platform アプリケーションビルタ	/_		
	ファイル 編集 アブリケーション オブ	ション ヘルプ	עאב	•行検索 ▼
	コンボーネント名称 電卓モデル			Ē 🖬 🖄 🖄
\backslash	(一)電卓モデル			
	ID: 32 KEY: ""			
		処理完了イベント	教値を文字列で設定する	■ 任意精度実数(BigDecimal)格納変数
	ID:32-15 KEY:"加算(+)15"			ID:32-17 KEY:"内部数值格納変數"
\mathbf{h}	1 文字列格納変数	データ設定イベント	数値を文字列で設定する	III 任意精度実数(BigDecimal)格納変数
\mathbf{A}	ID:32-16 KEY:"入力文字列格納変数"			ID:32-17 KEY:"内部数值格納変数"
			数値(BigInteger)を設定する	■整数(BigInteger)格纳变数
		Er	rorOnly	ID:32-20 KEY:"整数(BigInteger)格納変数"
			値を1減らす	■整数(BigInteger)格納変数
\backslash		Er	rorOnly	ID: 32-20 KEY:"整数(BigInteger)格纳变数"
\mathbf{h}		Ļ	指定インデックス間の部分文字列に置き換える	○ 文字列格納変数 □ 2:32-16 □
\backslash		Er	roroniy	KEY:"入为文字列格納変数"
\mathbf{h}	I 任意精度実数(BigDecimal)格納変数 ID: 32-17	データ設定イベント	イベントを伝播させる	■ 電卓モデル ID:32
\backslash	KEY:"内部数值格納変数"			KEY:"
\mathbf{A}	ID: 32-20			
	KEY:"整要(Biginteger)格納変要…"	ゴニカ砂字 イット		
	算冊/真算1コンホーキン118/03.800 D:32-21 /(D:0-113.15)を行ったが一本が11		処理を呼び出す	ID: 32-23
\setminus	(KEY: 八川道县于恰纳发现			(KEY.) 演算実行リブルーナブ
) <u>演算</u>	ID: 32-22 KFY:"内部當管子格納密點"
λ	算術演算子コンポーネント格納密数	データ設定イベント		■ 算術演算子コンポーネント格納変数
	ID:32-22 KEY:"内部演算子格纳変数"		→風見〒の2年1時の2敗1世でス子91で設定して数1世変換する 	ID:32-22 KEY:"内部演算子格納麥發"
		マクションレイベント		
	[アイコン凡例] 🛑 アプリケー	ション 🗾 画面構成部品 🗾 ウィンドウ	────────────────────────────────────	面構成) 🗐 複合(処理) 🗐 リモート
1	実行実行	テ(設定可) 画面編集 帳票編集	ロード 挿入 保存 上書き保存	クリア 終了

考え方

- 1. [複合コンポーネント] を追加する
- 2. 内部処理の部分を複合コンポーネントに追加する
- 3. 元の階層から [複合コンポーネント] で追加したコンポーネントを削除する

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■複合コンポーネント	1	

┢// 複合コンポーネントを追加しましょう。

- 必要なコンポーネントを追加します。
 作業領域で右クリック- [複合コンポーネント作成] [コンポーネント]を追加します。
- ② 追加した [複合コンポーネント] をダブルクリックします。



[複合コンポーネント]の中に入ります。画面が緑色に変わります。

<u>Step.5 複合コンポーネントの利用</u>

複合コンポーネントを利用しましょう。

1)複合コンポーネントの追加

複合コンポーネントの中を作ります。上の階層の内部処理の部分を複合コンポーネントに入れるので、 [挿入] メニューを使い保存してあるファイルを挿入します。



W MZ Platform アプリケーションビルダー	
ファイル 編集 アプリケーション オブション ヘルブ	コメント行検索 💌
コンポーネント名称 <no name=""></no>	È 🖻 📤 😤
No Name>	
ID: 32 KEY: ""	
【アイコン凡柳】 🗐 アブリケーション 📄 画面構成部品 🗐 ウィンドウ 📑 メニュー 🎫 👘 処理部品 🥤	■複合(画面構成) 🛑複合(処理) 🛑リモート
実行(実行設定可)(画面編集)(帳票編集)(ロード)(挿入)(保有))上書きり	保存 クリア 終了



複合コンポーネントに追加されます。

₩ MZ Platform アプリケーションビルダー			
ファイル 編集 アブリケーション オブション ヘルブ			コメント行検索 💌
コンポーネント名称 電卓モデル			t 🗈 🛋 🗲
■電車モデル D:32 VEY.***・	アクションイベント		
	アクションイベント	指定した文字列と連結して置き換える	文字列格納変数
D::32-3 KEY:11" ポタン D::32-4	アクションイベント	指定した文字列と連結して置き換える	ID:32:16 KEY:"入力文字列格納変数" 文字列格納変数 ① 文字列格納変数 D:32:16 日本本本和経体体帯制作
KET. 2 「ボタン D: 32-5 KEY **3*	アクションイベント	指定した文字列と連結して置き換える	
	アクションイベント	指定した文字列と連結して置き換える	
D:32-6 米容2:4- 米多ン D:32-7 大容2:15-3-7	アクションイベント	指定した文字列と連結して置き換える	D:32-16 达子·利龙子·利格纳麦数************************************
	アクションイベント	指定した文字列と連結して置き換える	5 一 文字列格納変数
Iter/***** Iter/***** Iter/***** Iter/***** Iter/****** Iter/******	アクションイベント	指定した文字列と連結して置き換える	したビイボクカ <u>文字列増給納支数</u>
ポタン ID:32:10 KEY:18* 一世の10	70937401	指定した文字列と連結して置き換える	文字列格納変数 D:32-16 KEY:"(入力文字列格納変数"
【アイコン凡例】 ■ アブリケーション 画画 実行 実行(設定可)	面構成部品 ■ウィンドウ 画面編集 帳票編集	 ●メニュー ●バネル ●処理部品 ● ●「「「」」」 ● ●<!--</td--><td>■ 雑合(通面構成) ■ 雑合(処理) ■ リモート 保存 クリア 終了</td>	■ 雑合(通面構成) ■ 雑合(処理) ■ リモート 保存 クリア 終了

MZ Platform THE SHELL		
ファール 編集 アブリケーション オブション・ルブ		□メント行検索 ▼
コレポーネント名称 電卓モデル		n 🔁 🛋 🛋
■電気==4L 10:32 KEY:=	799524425	
	アクションイベント 指定した文字列と連結し1	て置き換える し: 32-16 KEY: "人力文字列格納変散
ボタン D:32-4 KEY:"2"	アクションイベント 指定した文字列と連結して	【置き換える 文字列格納変数 [D:32-16 KEY:"入力文字列格納変数"
ボタン D:32-5 KEY:"3"	アクションイベント 指定した文字列と連結して	て置き換える ID:32-16 KEY:"人力文字列格納変数"
<u> ボタン</u> D:32-6 KEY:"4"	アクションイベント 指定した文字列と連結して	て置き換える し:32:16 KEY:"入力文字列格納変数"
ボタン D:32-7 KEY:"5"	アクションイベント 指定した文字列と連結して	て置き換える (図) 文字列格納変数 ID:32-16 KEY:"入力文字列格納変数"
ボタン D:32-8 KEY:"6"	アクションイベント 指定した文字列と連結して	て置き換える 「文字列格納変数 ID:32-16 KEY:"入力文字列格納変数"
ボタン D:32:9 KEY:" 7"	アクションイベント 指定した文字列と連結して	て置き換える 「D:32-16 KEY:"入力文字列格納変数"
ボタン D: 32-10 KEY:18"	アクションイベント 指定した文字列と連結して	【置き換える (図) 32-16 (EV: "人力文字列始約支数" () ***3108400794
【アイコン凡例】 ● アブリケーション ■)画 実行 実行(設定可)	厨構成部品 ●ウィンドウ ●メニュー ●パネル ●火 画面編集 「帳票編集 □ード 挿入 【保存	理部品 司 複合画面構成) 司 複合(処理) <mark>ラ</mark> リモート 上書き保存 クリア 終了

2) 複合コンポーネントの編集

挿入したファイルを整理します。

ここで作成しようとしている [電卓モデル] 複合コンポーネントにはGUI部品を含めないことにしますので、複合コンポーネント内から GUI 部品のコンポーネントを削除します。

操作 複合コンポーネントの中を編集しましょう。

> ここでは [ボタン] コンポーネント全部と [数値入力フィールド] コンポーネント、 [フレーム]コンポーネントを削除します。

① [ボタン] コンポーネントを削除します。

[ボタン(1)] コンポーネントの上で右クリックー [削除] をクリックします。

MZ Platform アプリケーションビルダー			
ファイル 編集 アプリケーション オプション ヘ	ルブ		コメント行検索 🔻
コンボーネント名称 電卓モデル		A.	Ē 🖬 🖄 🖄
 ● 電卓モデル ● 32:	アクションイベント		
(ボタン)	アクションイベント	指定した文字列と連結して置き換える	■ 文字列格納変数
KEY #3 イベント処理追加 ボタン メタン	アクションイベント	指定した文字列と連結して置き換える	KEY:"入力文字列格納変数" KEY:"入力文字列格納変数 For PS列格納変数 For PS列格納変数 For PS PS
ボタ 切り取り ボタ 削除	79824421	指定した文字列と連結して置き換える	ID:32-16 ICEY: "入力文字列格納変数" □ 文字列格納変数
KEY:**3**	709324421	指定した文字列と連結して置き換える	KEY:"入力文字列格納変数"
KEY:"4" メソッド起動位置検索 1ボタン ID:32.7 公開メソッド設定	アクションイベント	指定した文字列と連結して置き換える	KEY:"入力文字列格納変数" 文字列格納変数 ID:32-16
KEY:"5" インデント ボタン インデント	アクションイベント	指定した文字列と連結して置き換える	KEY: "人力文字列指納変数" 文字列格納変数 D: 32:16
(KEY: B ボタン D: 32-9 D: 32-9	アクションイベント	指定した文字列と連結して置き換える	KEY: 人力又子が格納変数 マ字列格納変数 ローンマット レロンマット ローンマット レロンマット レロンマット
	709824421	指定した文字列と連結して置き換える	
	7/31511/2016		- 古字副線紬亦動
[アイコン凡例] ■アプリケーション	■画面構成部品 ■ウィンドウ	■メニュー ■バネル ■処理部品 ■	▶複合(画面構成) 📕 複合(処理) 📕 リモート
実行 実行(設定可)	画面編集	ロード 挿入 保存 上書き係	存 クリア 終了

- ② 確認ダイアログボックスが表示されるので [はい] をクリックします。
- ③ [ボタン(1)] コンポーネントが削除されます。
- ④ 以下のボタンを削除します。

[ボタン(2)] ~ [ボタン(9)]、[ボタン(0)]、[ボタン(C)]、[ボタン(+)]、[ボタン(=)] [ボタン(0)]、[ボタン(.)]、[ボタン(-)]、[ボタン(×)]、[ボタン(÷)] [Shift]キーを押しながら、コンポーネントをクリックしていくと、まとめて選択できます。

▦	以下のようになり)ます。		
	MZ Platform アプリケーションビルタ ファイル 編集 アプリケーション オプ	/_ ション ヘルプ		
	コンポーネント名称 電卓モデル			
	■ 電車モデル D:32 KEY:"" フレーム D:32-1 KEY:"電車" 動値入力フィールド D:32-2	アクションイベント		
	KEY:*鉄値入力フィールド2* 加算(+) ID: 32-15 KEY:*加算(+)15*	処理完了イベント	数値を文字列で設定する	☐ 任意補度実数(BigDecimal)格納変数 ID: 32-17 KEV: *内部数值格約支数;
	○ 文字列格納変数 ID:32-16 KEV:3.2.5 KEV:3.2.5 KEV:3.2.5 KEV:3.2.5 KEV:3.2.5 KEV:3.2.5 KEV:3.2.5 KEV:3.2.5 KEV:3	データ設定イベント		
	(Ket: / OIX F/WBellX (X		<u>数値(BigInteger)を</u> 設定する ErrorOnly	
			値を1減らす ErrorOnly	ID: 32-20 KEY: "整数(BigInteger)格纳変数"
			指定インデックス間の部分文字列に置き換 ErrorOnly	える
	I 任意精度実数(BigDecimal)格纳支数 ID:32-17 KEY:"内部数值格纳支数"	データ設定イベント	表示したい文字列を設定する	数値入力フィールド D:32:2 KEY:"数値入力フィールド2"
	U-3-2420(BigIntegen格納支数…") (とて): 2420(BigIntegen格納支数…") 目 資源濃算子コンポーネント格納支数 ID: 32-21 KEY: "入力演算子格納支数"	データ設定イベント	日 <u>処理</u> を呼び出す	サブルーチン D:32-23 KEY:?資算実行サブルーチン* G 空気空気 フッシル・シント40600000 *
	[アイコン凡例] 🛑 アブリケー	ション 🗐 画面構成部品 🛑 ウィン	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	複合(画面構成) ● 複合(処理) ● リモート
	実行実行	〒(設定可) 画面編集 幌票編集	にした。 「「」」「」」「」「」「」」「」」「」「」」「」「」」「」」「」」「」」「」」	存 クリア 終了

⑤ [数値入力フィールド]を削除します。

確認

[数値入力フィールド (ID:32-2)] がメソッドとして接続されているコンポーネントがあるので、 その分を削除します。

[任意精度実数(BigDecimal)格納変数(Key:内部数値格納変数)]に接続されている [数値入力フィールド(ID:32-2)]の上で右クリック- [削除]をクリックします。

👑 MZ Platform アプリケーションビルダー		
ファイル 編集 アブリケーション オブション ヘルブ	コメント行	ī検索 ▼
コンボーネント名称 電卓モデル		🗈 🖻 🖄 📥
10:32-16 KEY:"人力文字列格納変数"	T	10:32-17 KEY:"内部数值格納変数"
	料値(BigInteger)を設定する	📑 整数(BigInteger)格納変数
	ErrorOnly	ID:32-20 KEY:"整数(BigInteger)格纳変数"
	値を1減らす	 整数(BigInteger)格納変数
	ErrorOnly	ID:32-20 KEY:"整数(BigInteger)格納変数"
	指定インデックス間の部分文字列に置き換える	文字列格納変数
	ErrorOnly	ID:32-16 KEY:"入力文字列格納変数"
●任意精度実数(BigDecimal)格納変数 データ設定イベント	表示したい文字列を設定する	数値入力フィールド
ID:32-17 KEY:"内部数值格納変数"		接続コンポーネント選択
整数(BigInteger)格纳实数		接続コンポーネント選択 ▶
ID:32-20 KEY:"整款(BigInteger)格纳変数"		起動メソッド設定
算術演算子コンボーネント格納変数 データ設定イベント	処理を呼び出す	▲ イベント番号設定 ▶
ID:32-21 KEY:"入力演算子格納変数"	T	起動モード・
	演算子を設定する	_ 「起動メソッド追加 ▶
■ 算術演算子コンポーネント格納変数 データ設定イベント ID: 22.22	演算子の左側の数値を文字列で設定して数値変換する	
KEY:"内部演算子格纳变数"		切り取り
■ サブルーチン ID:32-23 ■ アクションイベント	オペランド設定後、演算を行う	₩BR
KEY:**満賀実行サブルーチン"		コメント編集
	初期化する	接続コンポーネント宣言位置検索
		ゴレークポイント設定ノ4776余
	文字列を設定する(イベント発生なし)	ID: 32-16
	とう	【KEY:"入力文字列格納変数" 】 ▼ 標式) ■加合(加理) ■ リエート
実行 実行(設定可) 画面編集 帳票編集	ロード 挿入 保存 上書き保存 ク	7リア 終了

- ⑥ 確認ダイアログボックスが表示されるのではいをクリックします。
- ⑦ [数値入力フィールド (ID:32-2)] コンポーネントの上で右クリック- [コンポーネント削除] をクリックします。
- ⑧ 確認ダイアログボックスが表示されるのではいをクリックします。
- ③ [フレームボタン] コンポーネントを削除します。
 同様に、[フレーム (ID:32-1)] コンポーネントの上で右クリック- [コンポーネント削除] をクリックし、[フレーム (ID:32-1)] コンポーネントを削除します。



以下のようになります。

WZ Platform アプリケーションビルダー				- • • ×
ファイル 編集 アブリケーション オブション ヘルブ			コメント行検索	-
コンポーネント名称 電卓モデル			Ē.) 🗈 🔄 🖄
■電卓モデル 10:32- (KEY:	建完了イベント	数値を文字列で設定する	任意補度実数(BigDet D : 32-17 (Eq. 17)部技術格的支援 (任今法帝室数/BinDet	cimal)格納変数 <u>f</u>
□ - 12-16 KEY: "入力文字列格納変数"			D: 32-17 KEY: "内部数值格納変對	<u>timanan12</u> <u>ti</u> 変数
			ID : 32-20 KEY : "整数(BigInteger) ● 整数(BigInteger)格纳	<u>格納変数…</u> "
		ErrorOnly	ID:32-20 KEY:"整数(BigInteger)科	格納変數"
		指定インデックス間の部分文字列に置き換 ErrorOnly	またる 「 」 文字列格納変数 ID:32-16 KEY: 「人力文字列格納変	
任登積度実数(BigDecimal)格納支数 デー・ で、32-17 にない、する国政道格納支数・ 登数(BigInteger)格納支数 [0] 32-20 注数(BigInteger)格納支数・ で、当該取(BigInteger)格納支数・	タ設定イベント			
算術演算子コンボーネント格納変数 デー: 10:30.01	タ設定イベント	고	サブルーチン	
长空""大力演算子格纳实锁"		- 演算子を設定する	KEY:**道宜実行サブル- 「算術演算子コンポー ID:32-22 KEY:*内部演算子格納索	- チン" ネント格納変 数
「算術演算子コンボーネント格納支数 デー・ D:32-22 KEY・T7480(賞量子格纳支数)		演算子の左側の数値を文字列で設定して数値	変換する 算術演算子コンポージ ID:32-22 KEY:"内部演算子格納強	ネント格納変 該 ^{**}
[アイコン凡例] 一 アブリケーション 一画面構成	成部品 🛑 ウィンド	ウ 🧐メニュー 🗐パネル 🗐処理部品 🧯] 複合(画面構成) 🛛 複合(処理) 🧧	リモート
実行 実行(設定可) 画面紙	編集 帳票編集	ロード 挿入 保存 上書き付	¥存 クリア 終了	

3)イベント伝播

[数値入力フィールド] コンポーネントが削除されたので、[任意精度実数(BigDecimal)格納変数 (ID:32-17 Key:内部数値格納変数)] コンポーネントの接続先がなくなっています。[任意精度実数(BigDecimal)格納変 数] コンポーネントの接続先は上の階層の[数値入力フィールド]です。それには「外部へのイベント通知」 が必要になります。

これを実現するのが[イベント伝播]というメソッドになります。これによって複合コンポーネント内部に ある[任意精度実数(BigDecimal)格納変数]でデータ設定イベントが発生したときに[電卓モデル]に通知 され、外部のコンポーネントにとっては複合コンポーネントの中からデータ設定イベントが発生したように 見えることになります。

この反対で外部から複合コンポーネント内部に対してメソッド起動要求が必要な場合には「公開メソッド 設定」を行います。

接続確認

____コンポーネント同士の接続を確認します。

接続元コンポーネント	■任意精度実数(BigDecimal)格納変数 (ID:32-17,	
(イベント発生コンポーネント)	Key:内部数值格納変数)	
発生イベント	データ設定イベント	
接続先コンポーネント	■電卓モデル(ID:32)	
起動メソッド	イベントを伝播させる(PFEvent)	
<引数>	説明:対象イベント	
	取得方法:イベント	

複合コンポーネントのイベントを上の階層に送る

操作

」[イベント伝播] メソッドを設定します。

- ① イベントの接続先コンポーネントを選びます。
 左側の[任意精度実数(BigDecimal)格納変数(ID:32-17 Key:内部数値格納変数)] コンポーネントの
 [データ設定イベント]上で
 右クリックー[起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー[接続コンポーネント選択] ー
 [電卓モデル(ID:32)] コンポーネントをクリックします。
- ② 接続したコンポーネントの処理を選びます。
 接続したコンポーネントの上で右クリックー [起動メソッド設定] をクリックします。
 起動メソッド設定画面が表示されます。起動メソッド(処理)を選びます。
 [メソッド]の ◆ をクリックします。
 [イベントを伝播させる(PFEvent)] をクリックします。
 引数を設定します。

説明:対象イベント

取得方法 : イベント

設定後、了解ボタンをクリックします。

<mark>₩</mark> 2	記載メソッド情報				
		メソッド イイ	ベントを伝播させる(PFEven	1) • 全:	イソッド対象
NO	型	記 月	取得方法	コンポーネント	メソッド/値
(PFEvent	対象イベント	イベント	-	-
					了解 取消し

4)サブルーチンの利用

複合コンポーネント内の処理の一部をサブルーチンにまとめます。 [入力演算子格納変数]の処理をサブルーチンにまとめます。

準 備

ここでは以下のコンポーネントを追加します。

コンポーネント名	必要数	
■ サブルーチン	4	[処理部品]-[サブルーチン]-[サブルーチン]

操 作

- 必要なコンポーネントを追加します。
 作業領域で右クリックー[コンポーネントー括追加] [処理部品] [サブルーチン]
 [サブルーチン] を4つ追加します。
- ② [サブルーチン] コンポーネントの名前を変更しておきます。

 1 つめの [サブルーチン(ID:32-32)] を [+押下げサブルーチン]
 2 つめの [サブルーチン(ID:32-33)] を [-押下げサブルーチン]
 3 つめの [サブルーチン(ID:32-34)] を [×押下げサブルーチン]
 4 つめの [サブルーチン(ID:32-35)] を [÷押下げサブルーチン]
 と変更します。

接続確認 コンポーネント同士の接続を確認します。

演算ボタンの押下げ処理をサブルーチンにまとめる

接続項目	接続関係
接続元コンポーネント	■サブルーチン(ID:22-22 Kov:上畑下げサブルーチン)
(イベント発生コンポーネント)	= 5500 - 550 -
発生イベント	アクションイベント
接続先コンポーネント	■算術演算子コンポーネント格納変数(ID:32-21,
	Key:入力演算子格納変数)
起動メソッド	演算子を設定する(PFArithmeticOperator)
<引数>	説明:算術演算子コンポーネント
	取得方法:コンポーネント
	コンポーネント : 加算(+) (ID:32-15)

操作

演算子をサブルーチンにまとめます。

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[+押下げサブルーチン(ID:32-32)] コンポーネント上で
 右クリックー[イベント処理追加] [アクションイベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。

左側の[+押下げサブルーチン(ID:32-32)] コンポーネントの [アクションイベント]上で右クリック- [起動メソッド追加]とクリックします。 薄灰色の四角い枠が追加されます。 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。 右側に追加された薄灰色の四角い枠の上で右クリック- [接続コンポーネント選択] -[入力演算子格納変数(ID:32-21)]をクリックします。

③ 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。

- 起動メソッド(処理)を選びます。
- [メソッド]の **▼**をクリックします。
- [演算子を設定する(PFArithmeticOperator)]をクリックします。
- 引数を設定します。

説明:算術演算子コンポーネント

- 取得方法:コンポーネント
- コンポーネント : 加算(+) (ID:32-15)
- 設定後、了解ボタンをクリックします。

起動メソッド情報						×
	メソッド 演算子を設定する(PF	ArithmeticOperator)	-	🔲 全メソッド対象		
) 型	說明	取得方法	コンポーネント		メソッド/値	
0 PFArithmeticOperator	算術演算子コンポーネント	コンポーネント	加算(+)[ID:32-15](KEY:"加算(+)15	") -		
					了解取	消し
	起動メソッド情報 이 型 0 PFArithmeticOperator	起動メソッド情報 メソッド 演算子を設定する(PF 2 型 説明 0 PFArithmeticOperator 算術演算子コンポーネント	起動メソッド情報 メソッド 演算子を設定する(PFArithmeticOperator) 2 型 説明 取得方法 0 PFArithmeticOperator 算術演算子コンポーネント コンポーネント	起動メソッド情報 メリッド 演算子を設定する(PFArithmeticOperator) ▼ つ 型 説明 取得方法 コンポーネント o PFArithmeticOperator 算術演算子コンポーネント コンポーネント 加算(+)[ID:32-15](KEY:"加算(+)15	起動メソッド情報 メリッド 演算子を設定する(PFArithmeticOperator)	起動メゾッド情報 メリッド [渡算子を設定する(PFArithmeticOperator) - ① 全メリッド対象 ○ 型 説明 取得方法 コンポーネント メリッド/値 ○ PFArithmeticOperator 算術演算子コンポーネント 知覚(+)[D:32-15](KEY:*加算(+)15*) - 7解 取

④ ①~③の操作を繰り返して、[-押下げサブルーチン] [×押下げサブルーチン]
 [÷押下げサブルーチン]を接続します。
 (サブルーチンによってメソッドのコンポーネントを変更します)

5) 複合コンポーネントの中のメソッドを公開する

複合コンポーネントの中に設定したメソッドを上の階層から呼び出します。 上の階層からメソッドを呼び出すには複合コンポーネントの中のメソッドを上の階層に「公開」して使える ようにする必要があります。

「公開」するメソッドは上の階層で必要なものだけを公開します。

上の階層では [ボタン] コンポーネントと [数値入力フィールド] コンポーネントだけを残して、他のコ ンポーネントを複合コンポーネントに入れます。したがって、[ボタン] コンポーネントから接続してあるメ ソッドをすべて公開する必要があります。[数値入力フィールド] は結果を表示するところで使われるので、 複合コンポーネントで処理された計算結果を表示します。

ここで必要なのは以下の7つのメソッドです。これらを公開します。

W MZ Platform アプリケーションビルダー					
ファイル 編集 アプリケーション	ファイル 編集 アブリケーション オブション ヘルブ コメント行検索 💌				
アブリケーション名称 電卓アブ!	リケーション		e 🖻 🔁		
■ アプリケーション	アプリケーション開始イベント	つしてした表示する			
=	₽₽₽	20 AC3013 2	TD:1 KEY:"電卓"		
		에理 제태가 비가			
		242040119	ID:24 KEY: "初期化サブルーチン"		
ボタン	アクションイベント	北京」た 立字列と連結」で 果寺通うる	1 文字列格納変数		
ID:3 KEY:"1"	•	温定した文子列と産品して置き決入す	D:16 KEY:"入力文字列格納変数"		
「ボタン	アクションイベント	北京」た 立字列を連結」 て 果寺協 うる			
ID:4 KEY:"2"	•	指定した大子州と理解して置き決入す	D:16 KEY:"入力文字列格納変散"		
	アクションイベント	北京」た 古京和とはは」 て 果きゆうる			
ID:5 KEY:"3"	•	<u>指定した文子列と連結して直さ換える</u>	ID:16 KFY:"入力文字列格納変数"		
「ボタン	アクションイベント	北京」た天京和りはは」で異きゆうる	文字列格納変数		
ID:6 KEY:"4"	•	温足したステクに運転して直き換える	1D:16 KEY:"入力文字列格納麥對"		
	アクションイベント	北岸」も安定知らはは」が異き換える			
ID:7 KEY:"5"	•	<u>指定した文子列と連結して直さ換える</u>	ID:16 KFY:"入力文字列格納変数"		
	 アクションイベント	地学にたみ宮町には彼して東書橋長を			
ID:8 KEY:"6"		<u> </u>			
- ボタン	アクションイベント	化中国本 天空和長時4月 不要主任 二乙	文字列格納変数		
ID:9 KEY:"7"		<u>指定した文子列と連結して置き換える</u>			
(Ref) /	アクションイベント				
ID:10 ////////////////////////////////////		指定した乂子列と連結して直さ換える			
		化中国本 天空和長時4月 不要主任 二乙			
ID:11 IZ:V:"0"		指定した文子列と連結して直さ換える			
(RET. 3)					
ID:18 KEY:"0"		指定した乂子列と連結して直さ換える			
ID:19		指定した文子列と連結して直さ換える			
■ ボタン					
ID:12		処理を呼び出す			
ID:13 IZEV:"+"		通具十を設定する			
KET: 1					
ID:26	-	演算士を設定する			
ID:27		演算子を設定する			
KELL A 「ボタン」	 アクションイベント				
ID:28		演算子を設定する			
	アクションイベント				
ID:14		処理を呼び出す			
(KET					

操作 複合コンポーネントのメソッドを公開しましょう。

- ① [電卓モデル] 複合コンポーネントをダブルクリックして複合コンポーネントに入ります。
- ② [電卓モデル]の複合コンポーネントで [右クリック] [公開メソッド設定] をクリックします。



③ 公開するメソッドを選びます。

[文字列格納変数]から選びます。

[文字列格納変数(ID:32-16)]で右クリックー[公開メソッド設定...]をクリックします。

 	×
 電卓モデル [D:32] (KEY:") 加算(+) [D:32-15] (KEY:"九算(+)15") 文字列格納変数 [D:32-16] (KEY:"九力文字が 公開メンッド設定,") 登数(BigDecimal)格納変数 [D:3(-17) 登数(BigIneger)格納変数 [D:32-20] (KEY:"全数(Bounteger)格納変数2001 算術演算子コンポーネント格納変数 [D:32-21] (KEY:"九方演算子格納変数") 算術演算子コンポーネント格納変数 [D:32-22] (KEY:"小部演算子格納変数") サブルーチン [D:32-23] (KEY:"演算実行サブルーチン") リブルーチン [D:32-24] (KEY:"前期(セサブルーチン") Null判定 [D:32-25] (KEY:"小部演算子Null判定") 滅算(-) [D:32-29] (KEY:"小部算(-)29") 乗算(*) [D:32-30] (KEY:"東算(*)30") 除算(+) [D:32-31] (KEY:"+押下Ifサブルーチン") サブルーチン [D:32-33] (KEY:"+押下Ifサブルーチン") サブルーチン [D:32-34] (KEY:"+押下Ifサブルーチン") サブルーチン [D:32-35] (KEY:"+押下Ifサブルーチン") 	
	閉じる

④ [指定した文字列と連結して置き換える(String)]を公開します。
 [指定した文字列と連結して置き換える(String)]をクリックします。
 追加(公開化)をクリックし、設定をクリックします。

🛂 メソッド選択	×
	🔲 全メソッド対象
メソッド一覧	
全角カナ文字を半角に置き換える()	*
全角英数記号を半角に置き換える()	
半角カナ文字を全角に置き換える()	
半角英数記号を全角に置き換える()	
指定した文字列と連結して置き換える(String)	
指定インデックス間の部分文字列に置き換える(int,int)	=
指定インデックス間の部分文字列に置き換える(半角単位)(int,int)
指定文字列と一致するすべての文字列を置換する(String,String)	
指定文字列と一致する最初の文字列を置換する(String,String)	
改行文字を取得する()	
文字列の長さを取得する()	
文子列を取得する()	
文子列を設定する(String) 一方可加まれつする(King)	
乂子列を設定する(イベント発生なし)(String) 	
正規表現に適合するすべての文子列を直換する(String, String) 工想ま現に淹合する長知の立定別を実施する(String, String)	
止税衣税に過言する取約の文子列を直換する(String,String) 明女伝業から前方にお完良さの如公立実知を取得する (int)	
脱住位値がつ前方に指定支きの部方文子列を取得する(int) 相左位署から前方の指字文字列未完全取分文字列を取得する(eff	ing boolean)
p元生12世かっつ前7月の7日たスター100 C 2001 ステラルC4X14.9 の(50)	ing,boolean)
追加(公開化) 训除(非公開化)	
公開メリッド一覧	
	\frown
(

⑤ 公開メソッド名を変更します。公開したメソッドの上で右クリックー [公開メソッド名変更] をクリックします。

₩ 公開メソッド設定	
 ■ 電卓モデル [D:32] (KEY:") ■ 加算(+) [D:32-15] (KEY:"加算(+)15") ■ 文字列格納変数 [D:32-16] (KEY:"入力文字列格納変数") ● concatThenReplace(String) -> 指定した、子**** ▲ ConcatThenReplace(String) -> 指定した、子**** ▲ ConcatThenReplace(String) -> 指定した、子**** ▲ ConcatThenReplace(String) -> 指定した、子**** ▲ ConcatThenReplace(String) -> 指定した、子***** ▲ ConcatThenReplace(String) -> 指定した、子***** ▲ ConcatThenReplace(String) -> 指定した、子************************************	▲指定した文字列と連結して置き換える(Strin
	閉じる

⑥ 日本語表記として「数字を文字列として追加する」と入力し了解をクリックます。



、 、 公開メソッドは以下のようになります。

確認

₩ 公開メソッド設定	x
 ■ 電卓モデル [ID:32] (KEY:"') ■ 加算(+) [ID:32-15] (KEY:"九力文字列格納変数") ■ なっのcatThenReplace(String)> 数字を文字列として追加する(String) ■ となっのにはているのでは、「「「「」」」」」 ■ となっていたいでは、「」」 ■ となっていたいでは、「」 ■ となっていたいでは、「」」 ■ となっていたいでは、「」」 ■ となっていたいでは、 ■ となっていたいでは、	◆数字を文字列として迫加する(String)
	閉じる

⑦ 公開するメソッドを選びます。

[サブルーチン] から選びます。

[演算実行サブルーチン(ID:32-23)]で右クリックー[公開メソッド設定...]をクリックします。

⑧ [処理を呼び出す()]をクリックします。
 追加(公開化)をクリックし、設定をクリックします。

👑 メソッド選択	×
	🔲 全メソッド対象
メソッド一覧	
イベントを渡して処理を呼び出す(PFEvent)	
追加(公開化)	
公開メソッド一覧	
	ビー サキャンセル

- ⑨ メソッド名をわかりやすくします。公開したメソッドの上で右クリックー [公開メソッド名変更] をクリックします。
- ⑩ 日本語表記として「=押下処理を呼び出す」と入力し了解をクリックします。



- ① ⑦~⑩を繰り返して以下のサブルーチンの [処理を呼び出す()] を公開し、次のように名前を変更します。
 - サブルーチン(ID:32-24) [初期化処理を呼び出す] サブルーチン(ID:32-32) [+押下処理を呼び出す] サブルーチン(ID:32-33) [-押下処理を呼び出す] サブルーチン(ID:32-34) [×押下処理を呼び出す] サブルーチン(ID:32-35) [÷押下処理を呼び出す] 公開メソッド設定完了後は次のようになります。閉じるをクリックします。

 ● 電卓モデル [ID:32] (KEY."*) ● 加算(+) [ID:32-15] (KEY."カウ芋列格納変数") ● 文字列格納変数 [ID:32-16] (KEY."人力文字列格納変数") ● concatThenReplace(String) -> 数字を文字列として追加する(String) ● 任意精度実数(BigDecimal)格納変数 [ID:32-17] (KEY."内部救值格納変数") ● 任意精度実数(BigDecimal)格納変数 [ID:32-20] (KEY."雪数(BigInteger)格納変数20") ● 算術演算子コンボーネント格納変数 [ID:32-21] (KEY."大力演算子格納変数") ● 算術演算子コンボーネント格納変数 [ID:32-22] (KEY."雪数(BigInteger)格納変数") ● 算術演算子コンボーネント格納変数 [ID:32-22] (KEY."計算子格納変数") ● サブルーチン [ID:32-23] (KEY."前期]仕サブルーチン") ● call() -> #即下処理を呼び出す() ● Null判定 [ID:32-31] (KEY."新算子Null判定") ● 成算(+) [ID:32-32] (KEY."市押下Ifサブルーチン") ● call() -> #即下処理を呼び出す() ● サブルーチン [ID:32-33] (KEY."+#₱下Ifサブルーチン") ■ call() -> #IPT処理を呼び出す() ● サブルーチン [ID:32-33] (KEY."+#FTIfサブルーチン") ■ call() -> #IPT処理を呼び出す()
=■ サフルーチン [ID:32-35] (KEY:"÷押トけサフルーチン")

6) 公開してあるメソッドを上の階層から使用する

複合コンポーネントで公開したメソッドを上の階層から使用します。 現在設定されているメソッドを削除して複合コンポーネントのメソッドに置き換えます。

操 作	現在設定されているメソッドを削除しましょう。
-----	------------------------

① 複合コンポーネントから元の階層に戻ります。

右上の[編集サポートボタン]をクリックして1階層上に上がります。

🛂 MZ Platform アプリケーションビルダー			- 0 X
ファイル 編集 アブリケーション オブション ヘルブ		コメント行検索	\frown
コンポーネント名称 電卓モデル			1 🗈 💽 😤
● 電卓モデル □ :32 ···· └空 (*) ···· ··· ··· ··· ··· ··· ··· ··· ···	♡h ¥k/z-vstrat	C 一 任意 諸 唐 実 款 Cloid	Decimal)格納変数
D:32-15 KEY:"加谊(+)15"	一般地を大子列で記	R走9 つ ID: 32-17 KEY: "内部数値格納速	Citt"
	ペント 数値を文字列で計 	<u>食定する</u> 日 任意精度実数(BigE ID:32-17 KEY:"内部数値格納測	vecimal)格納変数 E数"
	数値(BigInteger)を	設定する 「整数(BigInteger)格	納変数
	ErrorOnly	ID: 32-20 KEY: 整数(BigIntege	<u>n)格納変数…"</u>
	値を1減ら ErrorOnly	す ID: 32-20 レビン: *教教(Piginteger)作品	11歳後期
	指定インデックス間の部分す ErrorOnly		
日日 日日 日日 日日 日日 日日 日日 日日 日日	^{イント} イペントを伝播	させる	
ID:32-17 KEY:"内部数值格纳変数"		KEY:"	
D:32-20 KEY:"整数(BigInteger)格纳变数*			
■ 算術演算子コンポーネント格納変数 データ設定す ID:32-21	^{ベント} 処理を呼び出	Lす ID: 32-23	
【KEY:"入力演算子格納变数")	KEY: *演算実行サブル	<u>/ーチン"</u> -ネント格納変数
	消算丁で設定	9 つ ID: 32-22 KEY: "内部演算子格組	波班"
■ 算術演算子コンボーネント格納変数 データ設定す ID:32-22	ペント 演算子の左側の数値を文字列で	設定して数値変換する	-ネント格納変数
	a*16	KEY: "内部演算子格組	<u>滚鼓"</u>
[アイコン凡例] 🛑 アブリケーション 🔤 画面構成部品	■ウィンドウ =>メニュー = パネル =	処理部品 🧧複合(画面構成) 🛑複合(処理)	■リモート
実行 実行(設定可) 画面編集	「「「「「「「「」」」を見ていていた。「「「」」「「「」」」を見ていていた。「「」」「「」」「「」」「「」」「「」」「」」「「」」」を見ていていた。	上書き保存 クリア 終了	

確認

元の階層に戻ります。

WZ Platform アプリケーションビルダー			
ファイル 編集 アブリケーション オブション	ヘルプ		コメント行検索 🔻
アブリケーション名称 電卓アブリケーション			💼 💼 🔄
〒プリケーション	アプリケーション開始イベント	フレームを表示する	
KEY:"電卓アブリケーション"			ID:1 KEY:"電卓"
		処理を呼び出す	
			ID:24 KEY:*初期化サブルーチン"
	アクションイベント	アブリケーションを終了する	■ アブリケーション
KEY:"電卓"	•		KEY: "電卓アブリケーション"
数値入力フィールド			
KEY: "数値入力フィールド2"			
	アクションイベント	指定した文字列と連結して置き換える	□ 文字列格納変数 □D:16
KEY:"1"			KEY:"入力文字列格纳变数"
	アクションイベント	指定した文字列と連結して置き換える	
KEY:"2"			KEY:"入力文字列格纳変数"
ID:5	アクションイベント	指定した文字列と連結して置き換える	□ 文字列格纳变数 ID:16
KEY:"3"			KEY:"入力文字列格納変数"
ID:6	7099924X2F	指定した文字列と連結して置き換える	UD:16
(KEY:"4"	7000-000-000-0		
		指定した文字列と連結して置き換える	ID:16
(KET:"5"	アクションイベント		
ID:8		指定した文子列と連結して置き換える	D:10 (D:13 中立字列線(地球)##
	アクションイベント		
D :9 KEY-" 7"		指定した文子列と連結して置き換える	ID:16 ID:20 IC:20 ID:20
	Thursday		一 古字制纹油漆料
[アイコン凡例] 🛑 アブリケーション	■画面構成部品 🛑 ウィンドウ	──メニュー ──バネル ──処理部品	🥻 複合(画面構成) 🛑 複合(処理) 🛑 リモート
実行 実行 康行 康行 康行 康 定日	J) 画面編集 帳票編集	ロード 挿入 保存 上書き付	森子 クリア 終了

② メソッドを削除します。

[アプリケーション]と接続されている[初期化サブルーチン(ID:24)]を削除します。 [初期化サブルーチン(ID:24)]の上で右クリック-[削除]をクリックします。

- ③ [起動メソッドを削除します。よろしいですか?]のメッセージが表示されるので [はい]をクリックします。
- ④ ②~③を繰り返して

すべてのメソッドを削除します。



以下のようになります。

Z Platform アブリケーションビルター (山、編集 アブリケーション・オゴション・	5 lb 7		<u>しし</u>
1ル 編集 アブリケーション オブション ・ リケーション名称 雷吉アゴリケーション	\WJ		2 F1719.**
	manting to a man date		
	7707-99 JANK61 A.2.P	フレームを表示する	1)/U=A
Y:"電卓アラリケーション"	700-016-001		KEY:"電車"
1		アブリケーションを終了する	
			(KEY:"電車アラリケーション"
2			
Y:"数値入力フィールド2"			
3	P9982442F		
Y:*1*			
4	アクションイベント		
Y.:"2"			
x92	アクションイベント		
<u>č"3"</u>			
*/9)	アクションイベント		
Č:"4"	•		
ポタン	アクションイベント		
7:"5"			
#92	アクションイベント		
°."6"	•		
#92	アクションイベント		
9 (:"7"	•		
ポタン	アクションイベント		
10	•		
#92	アクションイベント		
11.0.	•		
(1.0)	アクションイベント		
12			
	man a const		
13			
Y:"+"			
14	7995V4KVF		
<u>/:*=*</u>			
10算(+) 15	処理完了イベント		
Y:"加算(+)15"	-		
文字列格納変数	データ設定イベント		
16	•		
壬意精度実数(BigDecimal)格納支数	データ設定イベント		
1/	•		
ポタン	アクションイベント		
18			
ポタン	アクションイベント		
19			
基款(BigInteger)格纳实数			
20 (~ "\$P\$\$\$ (BigInteger)格纳索\$\$			
三次により、「「「」」」 「「」」 「「」」 「」」 「」 「」 「」 「」 「」 「」 「	データ設定イベント		
21 (17) 力读算子格纳实验"	•		
二ノシガス員一世的民気 宣従演算子コンガーネント格納客時	データ設定イベント		
22 () () () () () () () () () () () () () (
	アクションイベント		
23			
	700 - 14 - 14		
24 24			
<u>(::::::::::::::::::::::::::::::::::::</u>	An Ellipsian Tal Annual		
NUII#102	処理完了イベント		
Y:"内部演算子Null判定"			
26	アクションイベント		
/:*=*			
77	アクションイベント		
čixi 📃 📃	-		
Kay	アクションイベント		
⁴ √	-		
成寬(-)	処理完了イベント		
29	•		
	処理完了イベント		
(×)			
ŧξ(×) 30 ☆"乗ξ(×)30"			
乗算(×) 30 (<u>***乗算(×)30*</u> 徐算(÷)	処理完了イベント		
●算(×) 30 (*)重算(×)30* (*)重算(×)30* (*)重算(×)30* (*)(*)重(×)31*	処理完了イベント		
●篇(*) 30 30 注意(*)30" 注意(*)31" 電卓モデル	処理完7イベント		
単直(*) 30 (*) 単位(*) 学道(*) 学道(*) (*) 単位(*) 21 第0 (*) 単位(*) 21 第0 (*) 21 (*) (*) (*) (*) (*) (*) (*) (*)	処理完了イベント		
単直(*) (*) 無直(*)30' 学運(*) 学運(*) 学達(*)31' 豊卓モアル 夏少/ (*) (*) (*) (*) (*) (*) (*) (*)	処理未7イベント		

操作 現在設定されている不要なコンポーネントを削除しましょう。 ここでは電卓の画面編集で使用しているコンポーネントは残し、 処理関係の複合コンポーネントに設定したコンポーネントをすべて削除します。

- コンポーネントを削除します。
 [加算(+)(ID:15)]の上で右クリック- [削除] をクリックします。
- ② [コンポーネントを削除します。よろしいですか?]のメッセージが表示されるので [はい]をクリックします。
- ③ ①~②を繰り返して
 - 以下の12個のコンポーネントを削除します。
 - 1) [加算(+)(ID:15)](①~②で削除済み)
 - 2) [文字列格納変数 (ID:16)]
 - 3) [任意精度実数(BigDecimal)格納変数(ID:17)]
 - 4) [整数(BigInteger)格納変数(ID:20)]
 - 5) [入力演算子格納変数(ID:21)]
 - 6) [内部演算子格納変数(ID:22)]
 - 7) [演算実行サブルーチン(ID:23)]
 - 8) [初期化サブルーチン(ID:24)]
 - 9) [内部演算子 Null 判定(ID:25)]
 - 1 O) [減算 (一) (ID:29)]
 - 11) [乗算 (×) (ID:30)]
 - 1 2) [除算 (÷) (ID:31)]



以下のようになります。

アプリケーション	アプリケーション開始イベント	フレームを表示する	
KEY : "電卓アブリケーション"			ID:1 KEY:"電卓"
■ フレーム	アクションイベント	アプリケーションを終了する	■ アプリケーション
ID:1 KEY:"電卓"			KEY:"電卓アブリケーション"
──数値入力フィールド			
ID : 2 KEY : "数値入力フィールド2"			
■ ボタン	アクションイベント		
ID:3 KEY:"1"			
■ ボタン	アクションイベント		
ID:4 KEY:"2"			
	アクションイベント		
ID:5 KEY:"3"	_		
	アクションイベント		
ID:6 KEY:"4"			
ボ タン	アクションイベント		
ID:7 KEY:"5"			
「 ボタン	アクションイベント		
ID:8 KEY:"6"			
ボタン	アクションイベント		
ID : 9 KEY : " 7 "			
ゴボタン	アクションイベント		
ID :: 10 KEY : "8"			
「ボタン	アクションイベント		
)		
ボタン	アクションイベント		
KEY:"C"			
「ボタン」	アクションイベント		
KEY:"+"			
「ボタン」	アクションイベント		
(KEY:"="			
■ボタン ID:18	アクションイベント		
(KEY':"O"			
■ボタン ID:19	アクションイベント		
(KEY':". "			
<u> ボタン</u> ID:26	アクションイベント		
(KEY:"-"			
<u> ボタン</u> D:27			
(KEY:"×"			
<u>」ボタン</u> ID:28	799374X7F		
(KEY:"+"			
 ID: 32 			

操作 起動メソッドに複合コンポーネントの公開メソッドを追加しましょう。

- ① イベントの接続先コンポーネントを選びます。
 左側の[アプリケーション] コンポーネントの[アプリケーション開始イベント]上で
 右クリックー [起動メソッド追加] とクリックします。薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] –
 「電卓モデル(ID:32)]をクリックします。
- ② 接続したコンポーネントの処理を選びます。

接続したコンポーネントの上で右クリック- [起動メソッド設定]をクリックします。
起動メソッド設定画面が表示されます。
起動メソッド(処理)を選びます。
[メソッド]の ◆ をクリックします。
[初期化処理を呼び出す()]をクリックします。
設定後、「了解ボタンをクリックします。

🛃 起動メソッド情報	服		X
NO型	I兑8月	メソッド *押下処理を呼び出す() 取得 *押下処理を呼び出す() 初期化処理を呼び出す() 初期化処理を呼び出す() 数字を文字列として追加する(String) + 押下処理を呼び出す() - 押下処理を呼び出す() = 押下処理を呼び出す()	メソッド/値
			て解 取消し

③ ①~②の操作を繰り返してすべてのコンポーネントを接続します。
 接続先は次ページの画面図を参考にしてください。
 数字ボタンは引数(固定値、数字)も設定します。



■ アプリケーション	アプリケーション開始イベント	フレームを表示する	
KEY:"電卓アブリケーション"			ID:1 KEY:"電卓"
		初期化処理を呼び出す	電卓モデル
			ID:32 KEY:"電卓モデル"
	アクションイベント	アプリケーションを終了する	■ アプリケーション
IDU:1 KEY:"電卓"	•		KEY:"電卓アブリケーション"
■数値入力フィールド			
LEY: "数値入力フィールド2"			
	アクションイベント	数字を文字列として追加する	■電卓モデル ID: 32
	-		INEY:"電卓モデル"
	アクションイベント	数字を文字列として追加する	■電卓モデル ID:32
KEY:"2"			ID:32 KEY:"電卓モデル"
「ボタン」	アクションイベント	数字を文字列として追加する	■電卓モデル ID:32
	-		KEY:"電卓モデル"
	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID:32
			KEY:"電卓モデル"
	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID:32
			KEY:"電卓モデル"
	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID:32
			KEY:"電卓モデル"
ID:9	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID:32
KEY.:"7"			KEY:"電卓モデル"
<u>■ボタン</u> ID:10	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID:32
KEY:"8"			KEY:"電卓モデル"
<u>■ホタン</u> ID:11	7993747F	数字を文字列として追加する	■ 電卓セテル ID:32
(KEY:"9"			KEY:"電卓モデル"
本タノ D:18	PD9324 X2F	数字を文字列として追加する	ID: 32 ID: 32
(KEY:"0"	700-000		
ID:19		数字を文字列として追加する	电早モデル D:32
	77 (12) - 2) (22) 1		
ID:12		初期化処理を呼び出す	■ ■ ■ ■ ■ ■ ■ ■ ■ ■
	アクションイベント		
ID:13		+ 押下処理を呼び出す	■ 电子 C / // ID:32 //EV:"電告エデル"
	アクションイベント		
ID:14		三押ト処理を呼び出す	
	アクションイベント		
ID:26		- 押ト処理を呼び出す	
	アクションイベント		
ID:27 KEY:"x"		×押ト処理を呼び出す	
	アクションイベント	. + 田丁丁 - Ju - 7 田 - +: u - 1	
ID : 28 KEY : " +"		☆押下処理を呼び出す	ID:32 KEY:"雷貞モデル"
「電電卓モデル			

※ボタンの順序を整理した後の様子です。

7)電卓モデルからの発生イベントを数値入力フィールドに関連付ける

[電卓モデル] からの発生イベントを [数値入力フィールド] に関連付けます。

[電卓モデル]内部での作業により、[電卓モデル]からはデータ設定イベントが発生することになっていま す。このイベント処理を追加して、そのイベント内包データを [数値入力フィールド]に設定します。

接続確認コンポーネント同士の接続を確認します。

「雷卣モデル]	からの発生イベントを	「数値入力フィールド] に関連付ける
し电早に ノ / ビ」	がりの先生インイトを	し 奴 恒 ハ ノ ノ イ ニノレ い	」に関連目りる

接続項目	接続関係
接続元コンポーネント	■雪卢エデル(ID:32)
(イベント発生コンポーネント)	■ 単 半 て
発生イベント	データ設定イベント
接続先コンポーネント	■数値入力フィールド(ID:2)
起動メソッド	表示したい文字列を設定する(String)
<引数>	説明:文字列
	取得方法:イベント内包
	メソッド/値:イベント対象データ

操作 [電卓モデル]からの発生イベントを [数値入力フィールド] に関連付けましょう。

- 使用するイベントを選択し、コンポーネントを接続する準備をします。
 左側の[電卓モデル(ID:32)] コンポーネント上で
 右クリック- [イベント処理追加] [データ設定イベント] とクリックします。
- ② イベントの接続先コンポーネントを選びます。
 左側の[電卓モデル(ID:32)] コンポーネントの
 [データ設定イベント]上で右クリックー [起動メソッド追加] とクリックします。
 薄灰色の四角い枠が追加されます。
 右側に追加された薄灰色の四角い枠にコンポーネントを割り当てます。
 右側に追加された薄灰色の四角い枠の上で右クリックー [接続コンポーネント選択] ー
 [数値入力フィールド(ID:2)]をクリックします。

- ③ 接続したコンポーネントの処理を選びます。
 - 接続したコンポーネントの上で右クリック- [起動メソッド設定] をクリックします。 起動メソッド設定画面が表示されます。 起動メソッド(処理)を選びます。 [メソッド]の ◆ をクリックします。 [表示したい文字列を設定する(String)] をクリックします。
 - 引数を設定します。
 - 説明:文字列
 - 取得方法:イベント内包
 - メソッド/値:イベント対象データ
 - 設定後、了解ボタンをクリックします。

🛂 起動メソッド情報				×	
メソッド 表示したい文字列を設定する(String) 🔹 全メソッド対象					
NO 型	説明	取得方法	コンポーネント	メソッド/値	
0 String	文字列	イベント内包	-	イベント対象データ	
				i	
				了解 取消し	
<u></u>					

ここまで進めるとビルダー上では以下のようになります。

アプリケーション	アプリケーション開始イベント	フレームを表示する	■ フレーム
KEY:"電卓アブリケーション"			ID:1 KEY:"電卓"
		初期化処理を呼び出す	■ 電卓モデル
			ID:32 KEY:"電卓モデル"
<u> </u>	アクションイベント	アブリケーションを終了する	アプリケーション
ID:1 KEY:"電卓"	•		KEY:"電卓アプリケーション"
一数値入力フィールド			
ID:2 KEY: "数値入力フィールド2"			
	アクションイベント	数字を文字列として追加する	■ 電卓モデル
ID:3 KEY:"1"	•		ID:32 KEY:"電卓モデル"
	アクションイベント	数字を文字列として追加する	■ 電卓モデル
ID:4 KEY:"2"	•		ID:32 KEY:"電卓モデル"
「ボタン」	アクションイベント	数字を文字列として追加する	電卓モデル 10
	•		ID:32 KEY:"電卓モデル"
	アクションイベント	数字を文字列として追加する	電卓モデル し ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
KEY:"4"	-		IDE 32 KEY:"電卓モデル"
ボタン	アクションイベント	数字を文字列として追加する	電卓モデル し ・ ・ 22 ・ ・ ・ 22 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
KEY:"5"	-		ID-32 KEY:"電卓モデル"
ボタン	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID: 32
	-		KEY:"電卓モデル"
ボタン ID:9	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID:32
KEY": "7"	-		KEY:"電卓モデル"
<u> 「 ボタン</u> 」 D:10	アクションイベント	数字を文字列として追加する	■電卓モデル □ : 32
KEY:"8"			KEY:"電卓モデル"
<u> ボタン</u> ID:11	アクションイベント	数字を文字列として追加する	■ 電卓モデル ID:32
KEY:"9"			KEY:"電卓モデル"
<u> ボタン</u> D:18	アクションイベント	数字を文字列として追加する	10:32
KEY:"O"			KEY:"電卓モデル"
<u> ボタン</u> D:19		数字を文字列として追加する	■ 電卓モデル ID:32
KEY:". "			KEY:"電卓モデル"
<u> ボタン</u> D:12		初期化処理を呼び出す	
KEY:"C"			KEY:"電卓モデル"
<u> ホタン</u> D:13	アクションイベント	+押下処理を呼び出す	■ 電卓セテル ID: 32
(KEY:"+"			KEY: "電卓モデル"
ホタノ D:14		=押下処理を呼び出す	■ 1 ¹ 電早モナル ID:32
(KEY:"="			KEY:"電卓モデル"
ホタノ D: 26		- 押下処理を呼び出す	ID: 32
(KEY:"-"	7/121-21/2214		(KEY:"電卓モデル"
10:27		×押下処理を呼び出す	● 単早モアル ID:32 ID:32
	マカションチャント		(KEY:"町早モナル" (П) (日) (日) (日)
ID: 28		+押下処理を呼び出す	● 単早モナル ID:32 ID:0:00mmまままでは、
	ビニカ現在ノット		(KEY:"電車モデル"
 ● 电早てテル ID:32 ID:32 	テージ設定1ヘノト	表示したい文字列を設定する	
(KEY:"電卓モデル"			KEY: "数値人刀フィールド2"