

関数電卓コンポーネント

1. 用途

非 GUI コンポーネントです。

このコンポーネントは、文字列として与えられた数式に対する実数値の代入計算を行います。四則演算の他、剰余、べき乗、開平、三角関数、対数、ラジアン-度変換の各種演算を扱うことができます。数式の長さや含まれる変数の数に制限はありません。

具体的な用途としては、テンプレートとなる計算式を設定しておいて、動的に変わる値に従った計算結果を必要とするケースなどが挙げられます。

アプリケーションビルダーのメニューからは、以下のようにして選びます。

[コンポーネント追加]-[処理部品]-[演算制御]-[関数電卓]

次に、関数電卓が処理可能な数式一覧を以下に記載します。

演算子および関数名	機能概要
$x+y$	加算 (x と y の和)
$x-y$	減算 (x と y の差)
$x*y$	乗算 (x と y の積)
x/y	除算 (x の y による商)
$x\%y$	剰余算 (x の y による剰余)
x^y	べき乗算 (x の y 乗)

<code>abs(x)</code>	絶対値 (x の絶対値)
<code>acos(x)</code>	逆余弦 (x の逆余弦、返値の単位はラジアン)
<code>asin(x)</code>	逆正弦 (x の逆正弦、返値の単位はラジアン)
<code>atan(x)</code>	逆正接 (x の逆正接、返値の単位はラジアン)
<code>atan2(x, y)</code>	逆正接 (y/x の逆正接、返値の単位はラジアン)
<code>ceil(x)</code>	x 以上の最小の整数値
<code>cos(x)</code>	余弦 (x の余弦、x の単位はラジアン)
<code>exp(x)</code>	自然対数の底 e のべき乗 (e の x 乗)
<code>floor(x)</code>	x 以下の最大の整数値
<code>log(x)</code>	自然対数 (x の自然対数)
<code>max(x, y)</code>	最大値 (x と y のうち、大きい値)
<code>min(x, y)</code>	最小値 (x と y のうち、小さい値)
<code>random()</code>	0 以上 1 未満の乱数
<code>rint(x)</code>	x に最も近い整数値
<code>sin(x)</code>	正弦 (x の正弦、x の単位はラジアン)
<code>sqrt(x)</code>	開平 (x の平方根)
<code>tan(x)</code>	正接 (x の正接、x の単位はラジアン)
<code>toDegrees(x)</code>	ラジアンから度への変換
<code>toRadians(x)</code>	度からラジアンへの変換

2. 用途

関数電卓が、実際のアプリケーション内で、どのような状況において使用されるかの例を挙げます。

- アプリケーション中で、利用者に計算させたい場合。

3. ここで使用されるイベントとメソッド

表1 イベント一覧

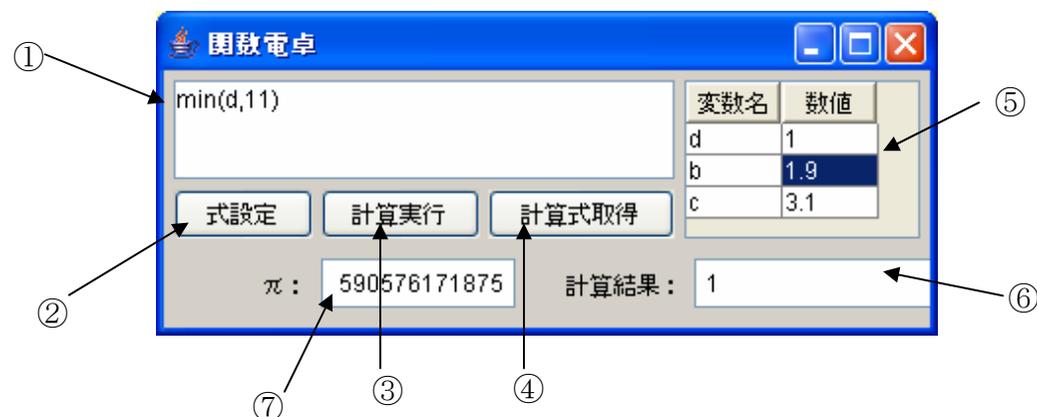
トリガ	対象データ	イベント番号
データ設定時	イベント対象データ	-
演算完了時	処理結果データ	-

表2 メソッド一覧

主な機能	機能説明
setFormula(String)	計算式を設定します。
getFormula()	実行する計算式を取得します。
evaluate(PFLabeledObjectList)	setFormula()で設定した計算式に変数が存在する時、PFLabeledObjectList の値を代入して計算を実行します。
getResult()	計算結果を取得します。
getPI()	π の値を取得します。

4. コンポーネント使用例

以下のサンプルアプリケーションでは、前述の「主な機能」を全て使用しています。



サンプルアプリケーション画面項目説明

番号	画面項目名称	説明
①	計算式設定テキストフィールド	計算式を設定します。
②	計算式設定ボタン	①で設定した計算式を関数電卓コンポーネントに設定します。
③	計算実行ボタン	計算を実行します。
④	計算式取得ボタン	②を押下して設定した計算式をダイアログで表示します。
⑤	変数設定テーブル	計算式の変数部分の値を設定します。
⑥	計算結果表示数値フィールド	③の押下時に計算結果を表示します。

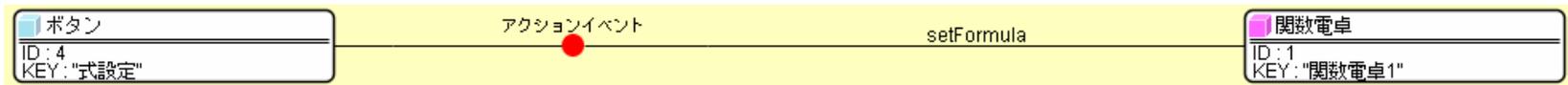
⑦

π 値表示数値フィールド

π の値を表示します。

以降では、1 つ 1 つの処理について、ビルダ上での設定を中心に解説していきます。アプリケーションビルダーから、“AP_DATA/Sample/関数電卓.mzax” をロードしてください。

4.1. setFormula(String)



M2 起動メソッド情報

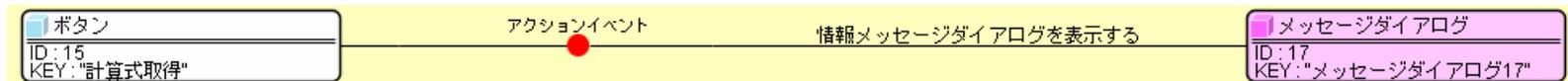
メソッド `setFormula(String)` 全メソッド対象

NO	型	説明	取得方法	コンポーネント	メソッド/値
0	String		メソッド戻り値	テキストエリア [ID:8] (KEY:"テキストエリア8")	テキストを取得する

閉じる

「setFormula(String)」というメソッドを起動して、計算式を入力したテキストエリアの表示文字列を関数電卓の計算式に設定しています。

4.2. getFormula()



M2 起動メソッド情報

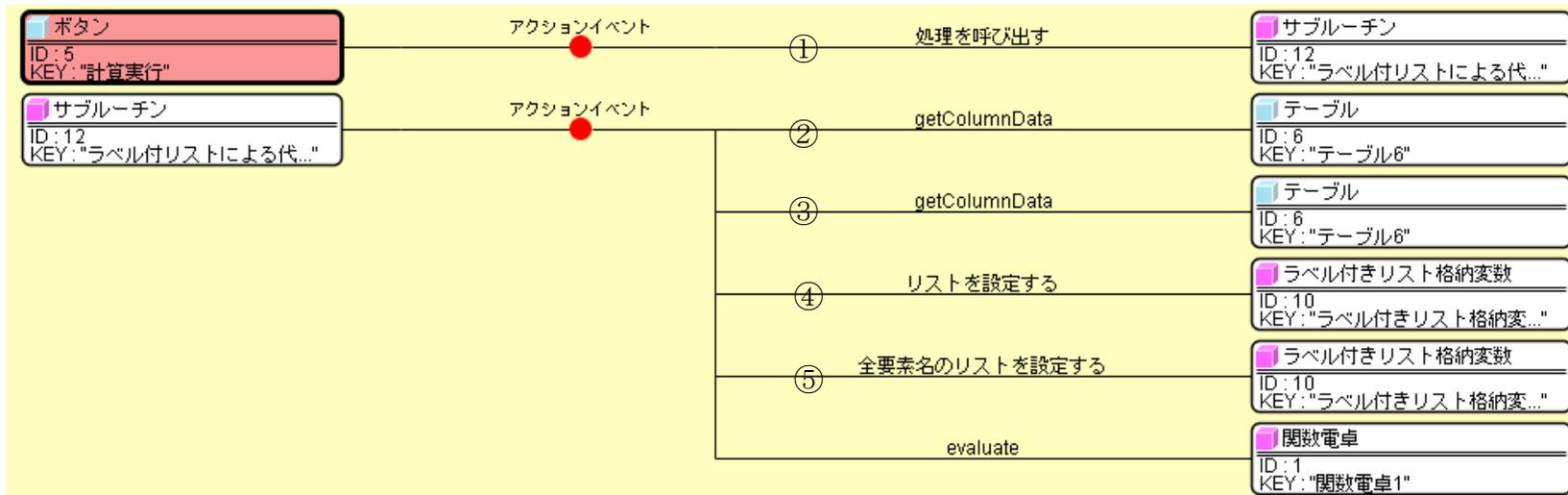
メソッド `情報メッセージダイアログを表示する(Component,String,String)` 全メソッド対象

NO	型	説明	取得方法	コンポーネント	メソッド/値
0	Component	親コンポーネント	コンポーネント	フレーム [ID:2] (KEY:"関数電卓")	-
1	String	メッセージ	メソッド戻り値	関数電卓 [ID:1] (KEY:"関数電卓1")	getFormula
2	String	タイトル	固定値	-	計算式取得

閉じる

起動したメソッドの、第2引数に「getFormula0」というメソッドによって、設定されている計算式を取得した結果をダイアログで表示しています。

4.3. evaluate (PFLabeledObjectList)



“式実行”ボタンのアクションイベント発生時に、“ラベル付リストによる代入値設定”サブルーチンを呼び出します。呼び出されたサブルーチン内では、「evaluate(PFLabeledObjectList)」を実行するための、値を引数に設定して、式を実行します。サブルーチン内の処理の詳細を以下に記載します。

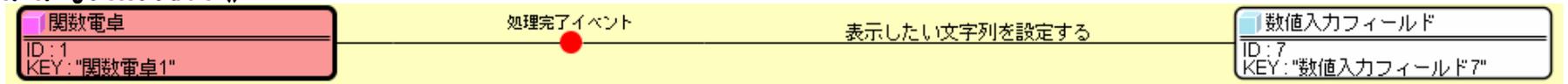
(以下の順番で処理を行います。)

- ①：テーブルから変数名の列データを取得します。
- ②：テーブルから数値の列データを取得します。
- ③：②で取得したデータをラベル付きリスト格納変数に設定します。

④：①で取得したデータをラベル付きリスト格納変数の要素名(ラベル)に設定します。

⑤：「evaluate(PFLabeledObjectList)」というメソッドを起動して計算を実行します。

4.4. getResult()



M2 起動メソッド情報

メソッド 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	メソッド戻り値	関数電卓 [ID:1] (KEY:"関数電卓1")	getResult

閉じる

「getResult()」というメソッドの戻り値(計算実行結果)を取得して、その結果を数値入力フィールドの表示文字列として設定しています。

4.5. getPI()



M2 起動メソッド情報

メソッド 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	メソッド戻り値	関数電卓 [ID:1] (KEY:"関数電卓1")	getPI

閉じる

「getPI0」というメソッドの戻り値(π の値)を取得して、その結果を数値入力フィールドの表示文字列として設定しています。

4.6. その他

計算式で使用している変数がテーブルで定義されていない場合は以下のエラー画面が表示されます。

