設計・製造支援アプリケーション構築プラットフォーム

MZ Platform

テーブルデータ操作サンプル

ーテーブルデータ基本操作から帳票印刷までー

テーブルデータ操作サンプル

このサンプルでは、テーブルデータの基本操作、基本操作を組み合わせた複合操作、帳票の印刷までの基本事項について説明しています。

◆目次

第 1 章 はじめに **1**

第 2 章 テーブルデータ基本操作 **2**

2.1 列の追加 3

2.2 列の削除 7

2.3 行の追加 10

2.4 行の削除 14

2.5 セルデータの取得 17

2.6 セルデータの設定 20

2.7 列データの取得 23

2.8 列データの設定 26

2.9 行データの取得 29

2.10 行データの設定 32

2.11 列の移動 35

2.12 行の移動 38

2.13 部分テーブルの取得 41

2.14 並べ替え（ソート、一列） 46

2.15 並べ替え（ソート、複数列） 49

2.16 分類テーブルの作成 57

2.17 セル値を用いた計算 60

2.18 列単位の集計 66

2.19 行単位の集計 71

第 3 章 テーブルデータ複合操作 **76**

3.1 集計表 1 77

3.2 集計表２ 82

3.3 クロス集計（ピボットテーブル生成） 87

第 4 章 帳票印刷 **95**

4.1 テーブルデータの印刷 96

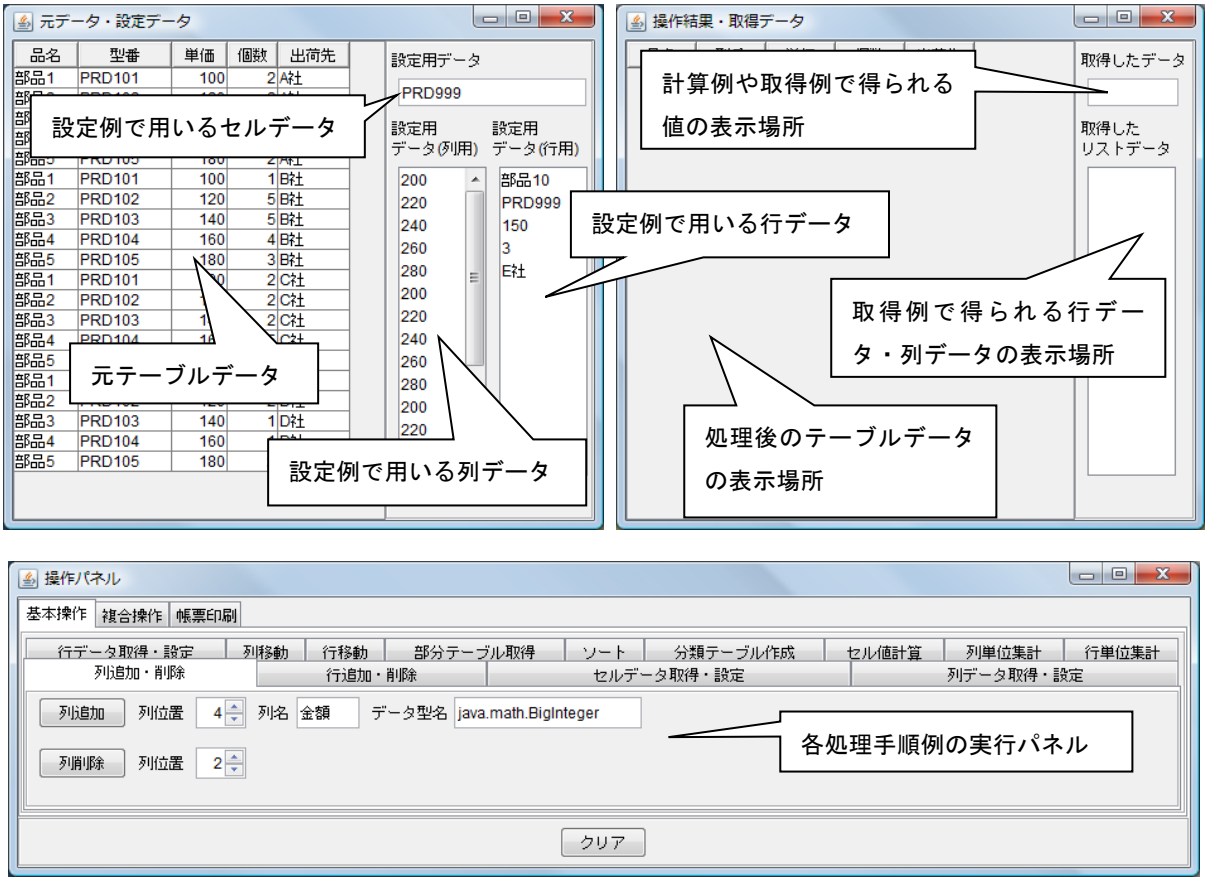
4.2 テーブル内のバーコード・ＱＲコード・イメージ印刷 101

4.3 繰り返し印刷 1（流し込み印刷）: テーブル編 104

4.4 繰り返し印刷 2（流し込み印刷）: ラベル・バーコード・QR コード・イメージ編 107

第1章 はじめに

本書では MZ Platform におけるテーブルデータの基本操作、基本操作を組み合わせた複合操作、帳票の印刷までの基本事項について説明しています。本書で説明するすべての処理手順例はサンプルアプリケーション“AP_DATA¥Sample¥テーブルデータ操作.mzax”に収められています。アプリケーションビルダーでロードし実行することにより、各処理手順例の動作の確認、処理手順の確認を行うことができます。このサンプルアプリケーションを実行すると次のような３つのフレームが表示されます。



左上のフレームには、元になるテーブルデータと各設定例で用いるデータを表示しています。右上のフレームには処理結果のテーブルデータと各例で得られるデータを表示します。下のフレームには各処理手順例を実行するための操作パネルが配置されています。

本書で説明する処理手順では、「ファンクション」コンポーネントの処理要求イベントのメソッド実行処理においてテーブルデータの処理を定義するとともに、初期データが設定された左上画面の「テーブル」コンポーネントのテーブルデータと必要情報とを引数として「ファンクション」コンポーネントのメソッドを実行し、戻り値として「ファンクション」から取得した処理結果を右上画面の「テーブル」コンポーネント等に設定することを基本とします。

第2章 テーブルデータ基本操作

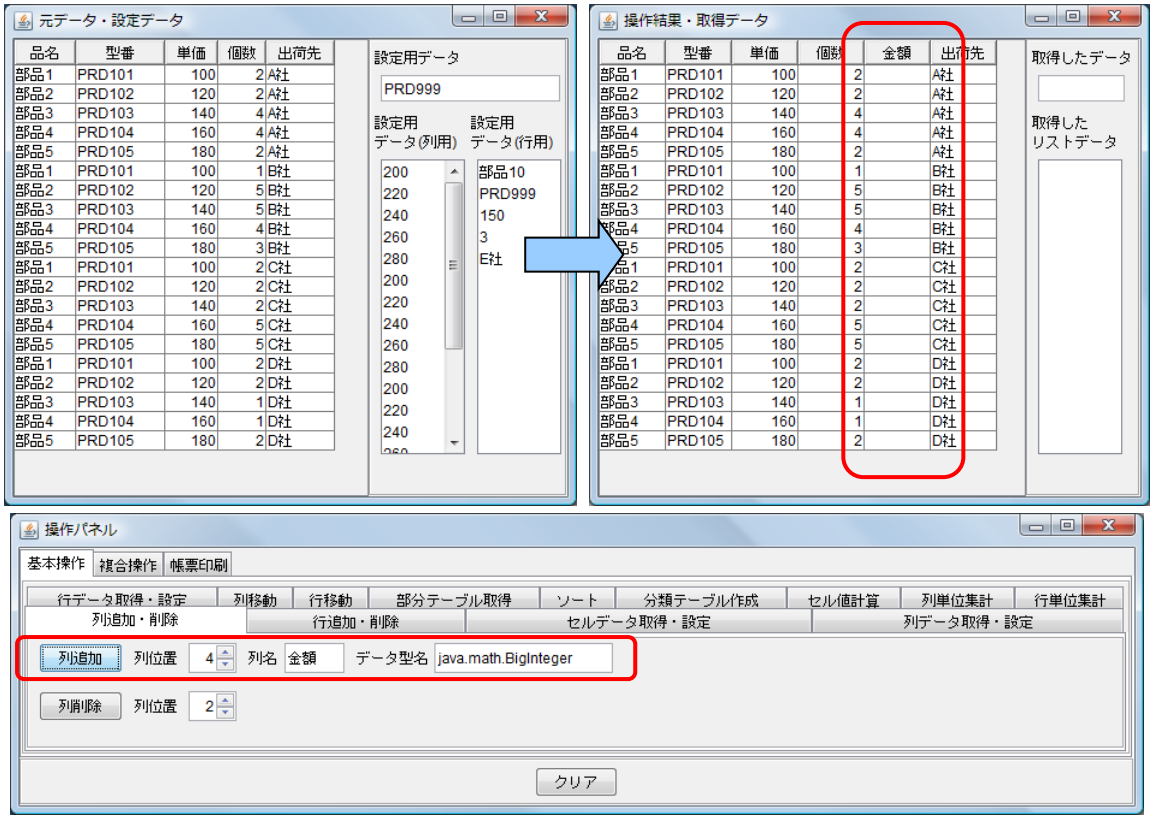
本章では MZ Platform におけるテーブルデータの基本的な処理手順について説明します。
本章でテーブルデータの処理のために使用する主なコンポーネントは次のとおりです。

コンポーネント名	場所	説明
■テーブル格納変数	*[コンポーネント追加]*－［処理部品］－［変数］－［テーブル格納変数］	テーブルデータを格納し、様々な操作を行うコンポーネント。
■テーブルサブセットフィルタ	*[コンポーネント追加]*－［処理部品］－［制御フィルタ］－［テーブルサブセットフィルタ］	テーブルデータから部分テーブルデータを抽出するコンポーネント。
■グラフデータ作成(複合コンポーネント)	*[複合コンポーネント追加]*－[グラフ]－[グラフ化モジュール.mzcx]	テーブルデータから様々なグラフ用のテーブルデータを生成するためのコンポーネント。本章では分類テーブルを生成するために使用します。
■ファンクション	*[コンポーネント追加]*－［処理部品］－［サブルーチン］－［ファンクション］	与えられた引数に対し、処理結果を返すコンポーネント。処理をまとめるために使用します。本章では、元になるテーブルデータと処理に必要な情報とをファンクションのメソッドに引数として与え、処理結果を戻り値として取得することを基本とします。
■ユーティリティ起動	*[コンポーネント追加]*－［処理部品］－［ユーティリティ］－［ユーティリティ起動］	幾つかの便利な機能を持つコンポーネント。本章ではクラス名からクラスを取得するために使用しています
■文字列格納変数	*[コンポーネント追加]*－［処理部品］－［変数］－［文字列格納変数］	文字列を格納し、様々な操作を行うコンポーネント。本章では、計算式を設定しセル値を代入するために使用します。
■関数電卓	*[コンポーネント追加]*－［処理部品］－［演算制御］－［関数電卓］	演算を行うコンポーネント。本章では、セル値に対する計算に使用します。
■基本統計処理（数値）	*[コンポーネント追加]*－［処理部品］－［統計］－［基本統計処理（数値）］	数値のリストデータに対して、合計値、平均値等を計算するコンポーネント。本章では、テーブルデータの列または行に対して、集計値を計算するために使用します。
■繰り返し制御（FOR）	*[コンポーネント追加]*－［処理部品］－[条件制御]－[繰り返し制御(FOR)]	繰り返し処理を行うコンポーネント。本章では、行または列について繰り返し処理を行う際に使用します。
■オブジェクト分岐	*[コンポーネント追加]*－［処理部品］－［条件制御］－［オブジェクト分岐］	オブジェクトにより条件分岐を行うコンポーネント。本章では、テーブル列のデータ型による処理の分岐等に使用します。

2.1 列の追加

動作イメージ

以下の図のように、テーブルデータに列を追加するための処理手順について説明します。



使用メソッド

テーブルデータに列を追加するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

コンポーネント	■テーブル格納変数
メソッド	指定位置に列を追加する(列名・型指定)(int, String, Class)
引数	第 1 引数：追加位置 第 2 引数：列名 第 3 引数：型
戻り値	なし
列の位置、列の名前、データ型を指定して、列を追加します。データ型は、文字列の場合「java.lang.String」、整数の場合「java.math.BigInteger」、実数の場合「java.math.BigDecimal」、日付の場合「java.util.Date」、論理値の場合「java.lang.Boolean」を設定します。	

類似する処理行うメソッドとして次のメソッドがあります。

メソッド	指定位置に列を追加する(int)
引数	第１引数：追加位置
戻り値	なし
基本メソッドの第１引数以外を省略したメソッドです。列名は空文字、データ型は文字列型に設定されます。	

メソッド	指定位置に列を追加する(列名・型・データリスト指定)(int, String, Class, PFObjectList)
引数	第１引数：追加位置 第２引数：列名 第３引数：型 第４引数：データリスト
戻り値	なし
基本メソッドに列データを追加したメソッドです。	

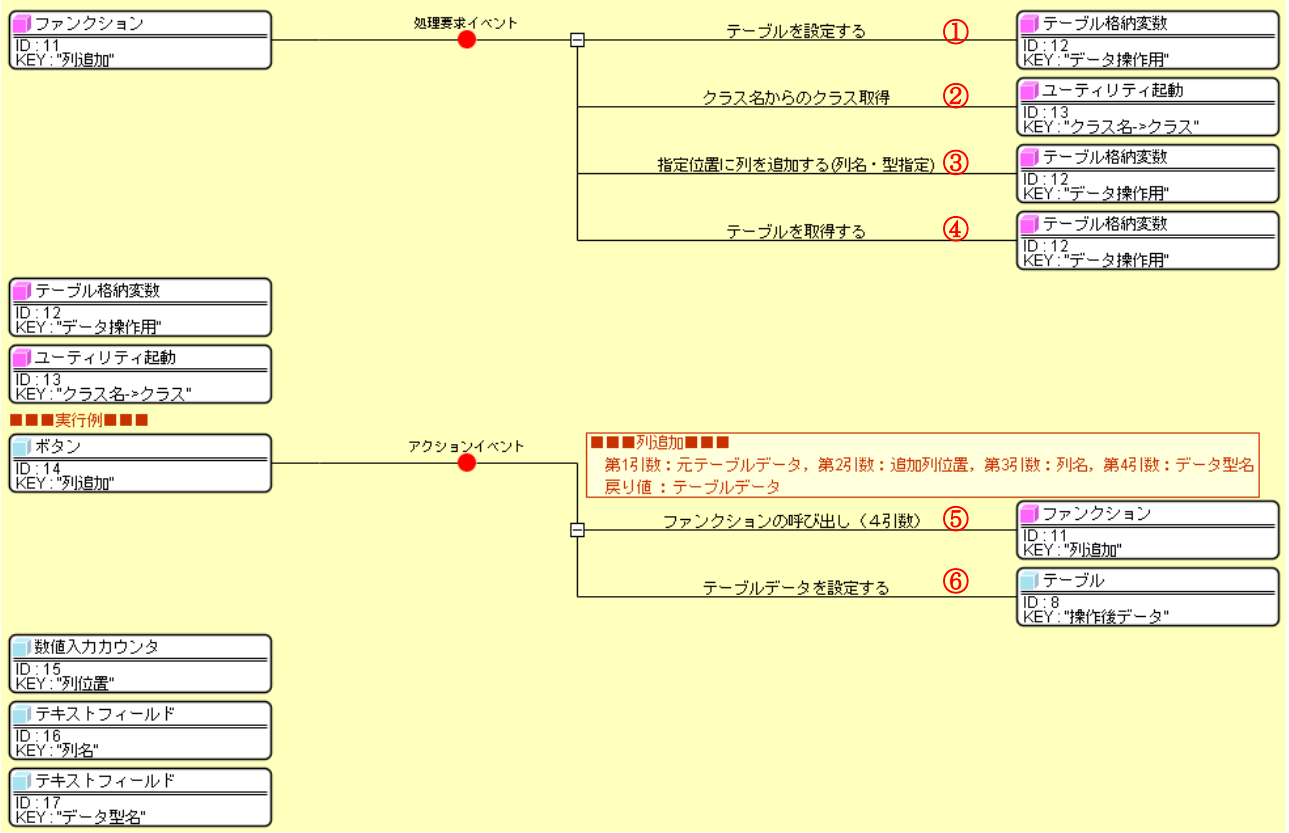
メソッド	最終尾に列を追加する(列名・型指定)(String, Class)
引数	第１引数：列名 第２引数：型
戻り値	なし
列の名前、データ型を指定して、最終列に列を追加します。	

メソッド	最終尾に列を追加する()
引数	なし
戻り値	なし
最終列に列名が空欄で文字列型の列を追加します。	

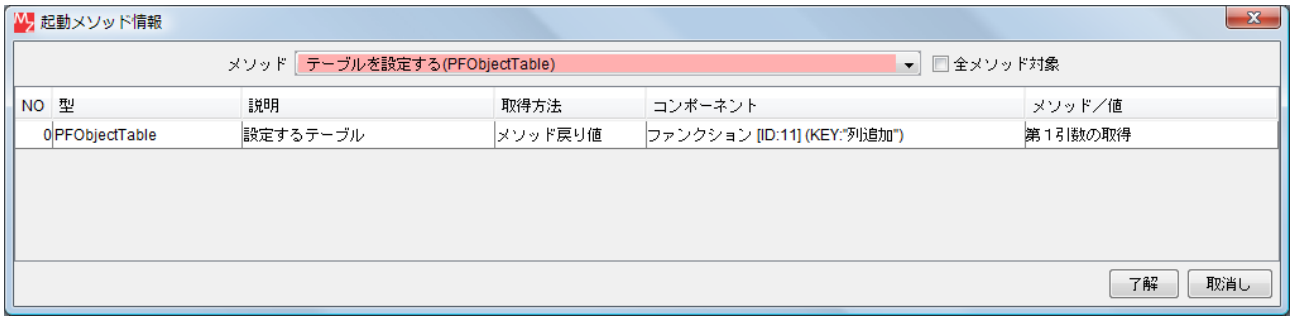
メソッド	最終尾に列を追加する(列名・型・データリスト指定)(String, Class, PFObjectList)
引数	第１引数：列名 第２引数：型 第３引数：データリスト
戻り値	なし
列の名前、データ型、列データを指定して、最終列に列を追加します。	

接続例

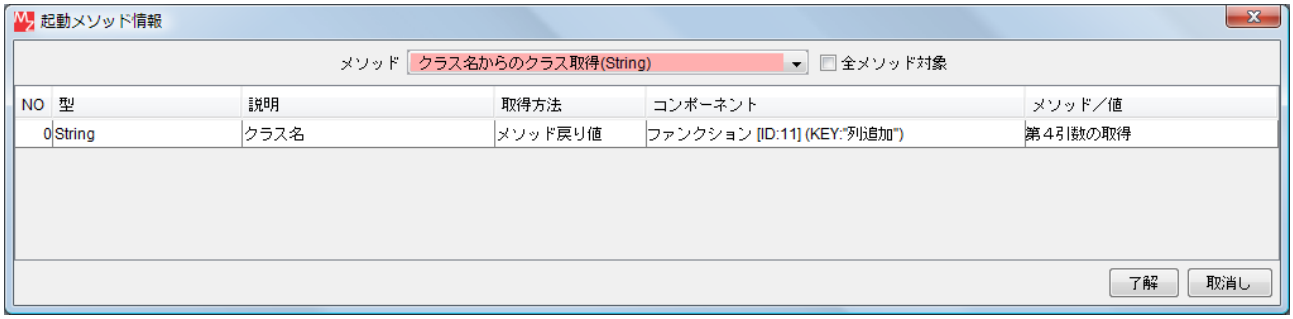
動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの列位置に、「テキストフィールド」に指定された列名及びデータ型で列を追加し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで列追加の処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第１引数に元テーブルデータ、第２引数に追加列位置、第３引数に列名、第４引数にデータ型名を指定してメソッド「ファンクションの呼び出し（4 引数)」を実行すると、結果のテーブルデータが戻り値として得られるように定義しています。



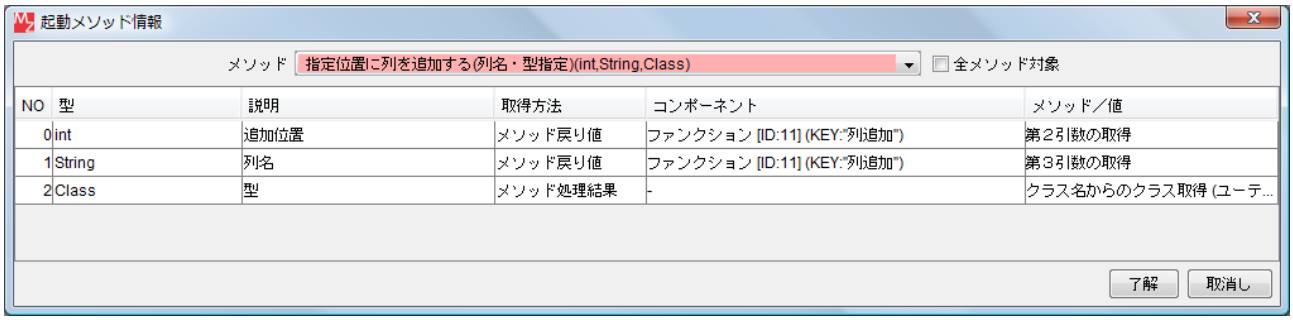
①の接続情報：処理対象のテーブルデータを設定する。



②の接続情報：クラス名の文字列からクラス（データ型）に変換する。

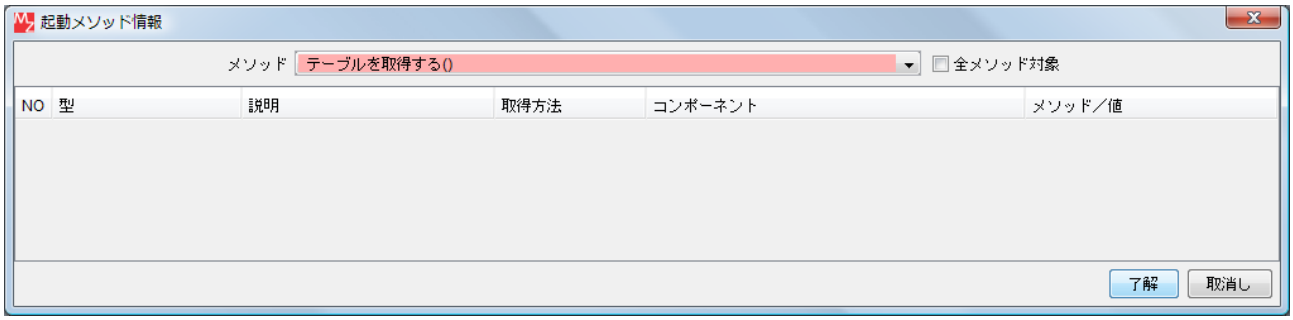


③の接続情報：追加する列位置（列インデックス）、列名、データ型を指定して列を追加する。



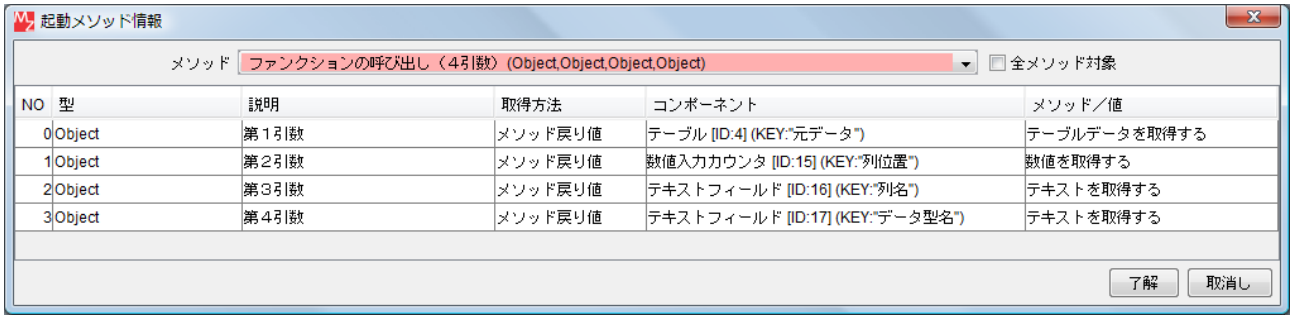
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	追加位置	メソッド戻り値	ファンクション [ID:11] (KEY:"列追加")	第2引数の取得
1	String	列名	メソッド戻り値	ファンクション [ID:11] (KEY:"列追加")	第3引数の取得
2	Class	型	メソッド処理結果	-	クラス名からのクラス取得 (ユーテ...

④の接続情報：ファンクションの戻り値としてテーブルデータを返す。



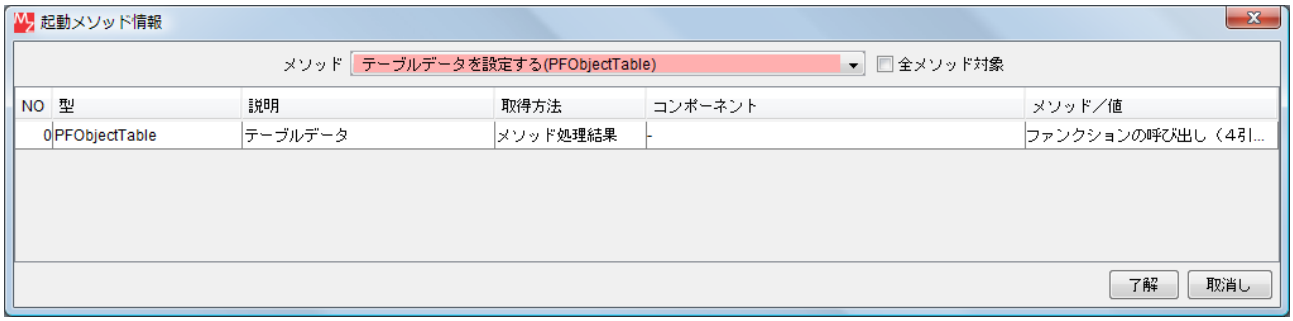
NO	型	説明	取得方法	コンポーネント	メソッド／値

⑤の接続情報：定義した列追加処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	メソッド戻り値	数値入力カウンタ [ID:15] (KEY:"列位置")	数値を取得する
2	Object	第3引数	メソッド戻り値	テキストフィールド [ID:16] (KEY:"列名")	テキストを取得する
3	Object	第4引数	メソッド戻り値	テキストフィールド [ID:17] (KEY:"データ型名")	テキストを取得する

⑥の接続情報：結果をテーブルに設定する。

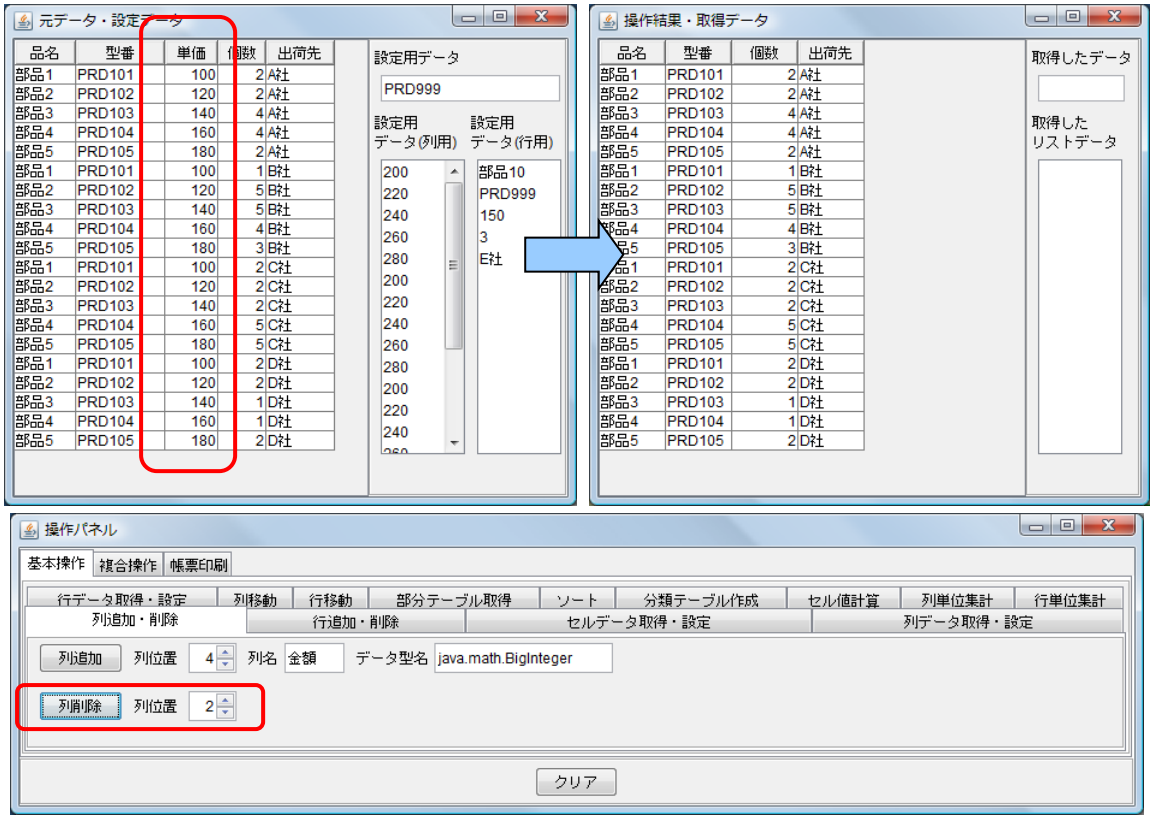


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（4引...

2.2 列の削除

動作イメージ

以下の図のように、テーブルデータの列を削除するための処理手順について説明します。



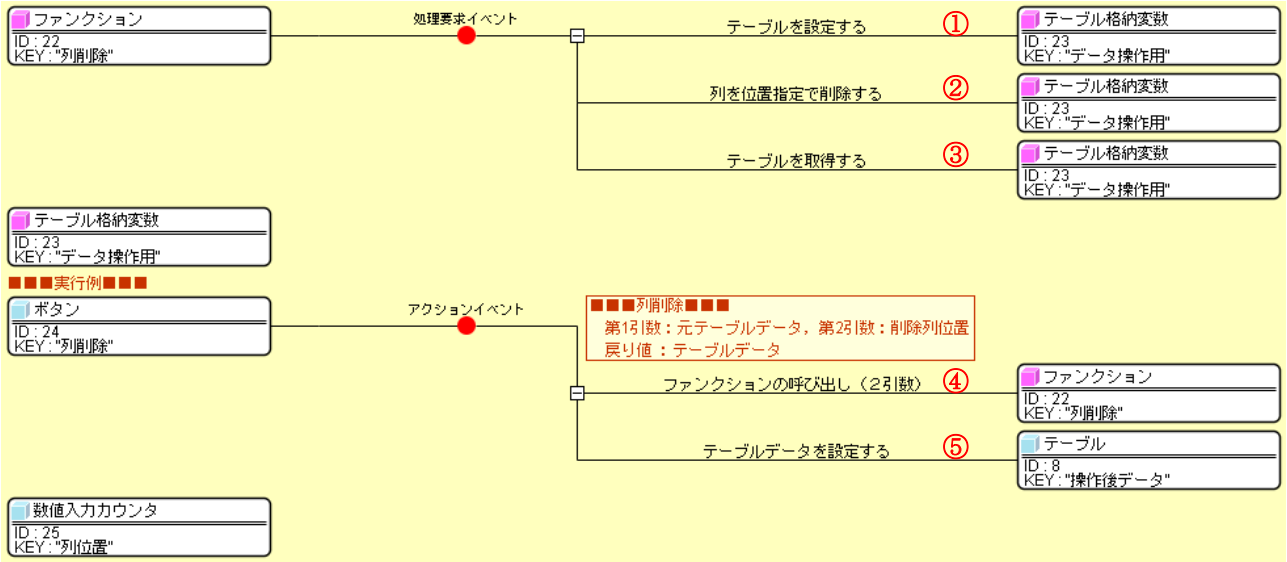
使用メソッド

テーブルデータの列を削除するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

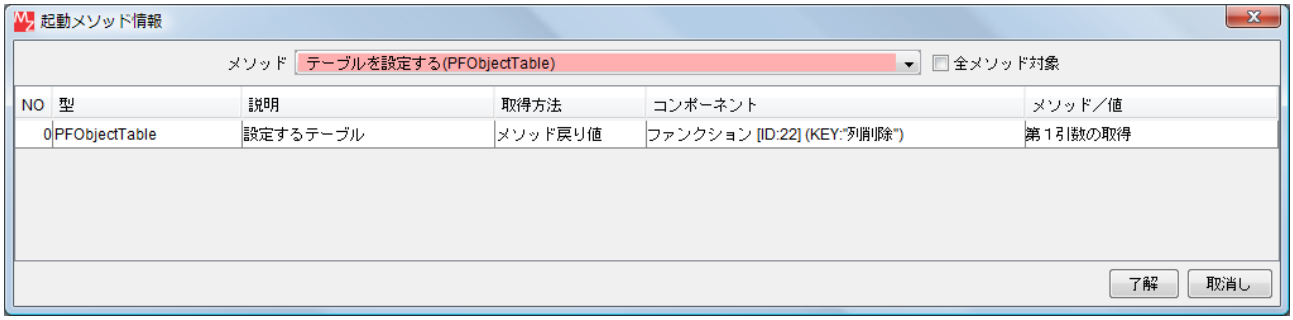
コンポーネント	■テーブル格納変数
メソッド	列を位置指定で削除する(int)
引数	第 1 引数：指定位置
戻り値	なし
列の位置を指定して、列を削除します。	

接続例

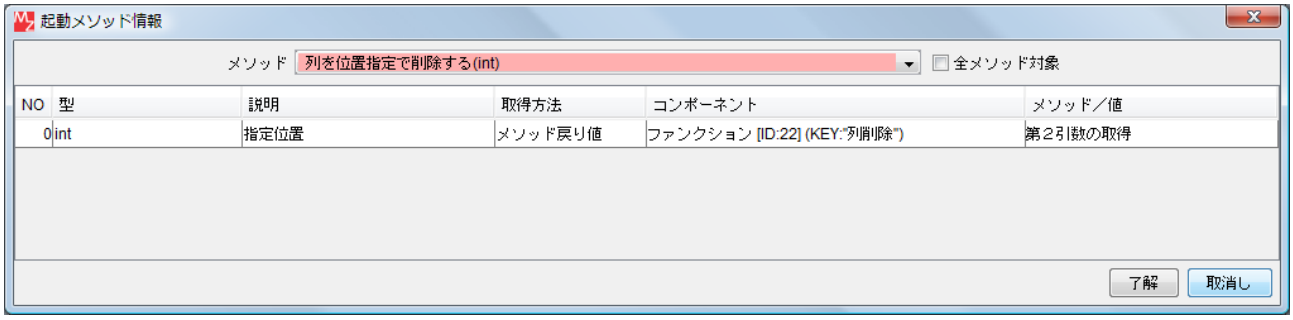
動作イメージのように、「数値入力カウンタ」で指定された位置の列を左側テーブルデータから削除し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで列削除の処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に削除列位置を指定してメソッド「ファンクションの呼び出し（2 数)」を実行すると、結果のテーブルデータが戻り値として得られるように定義しています。



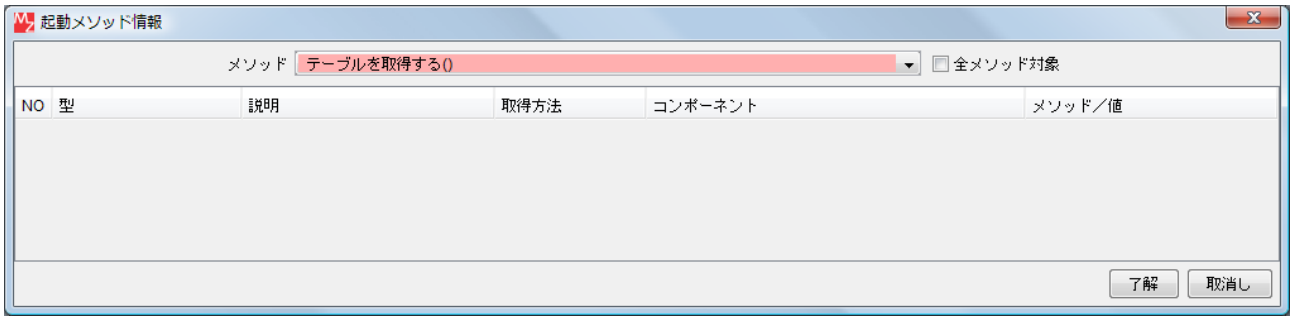
①の接続情報：処理対象のテーブルデータを設定する。



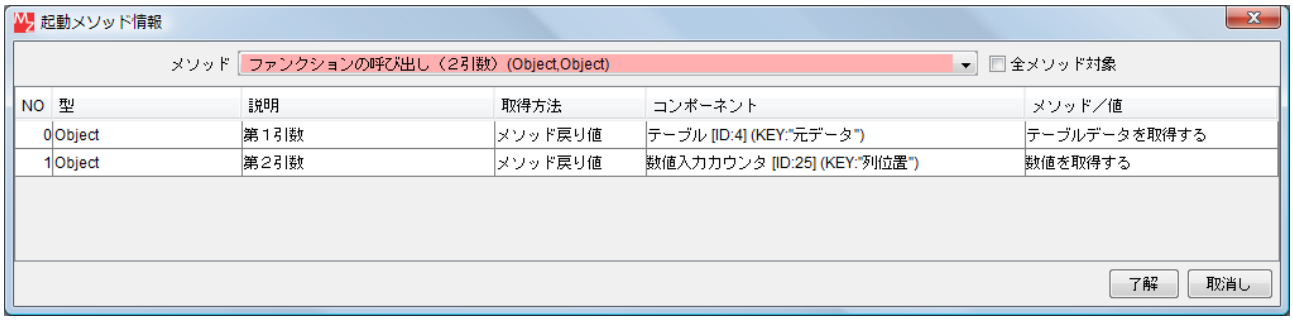
②の接続情報：削除する列位置（列インデックス）を指定して列を削除する。



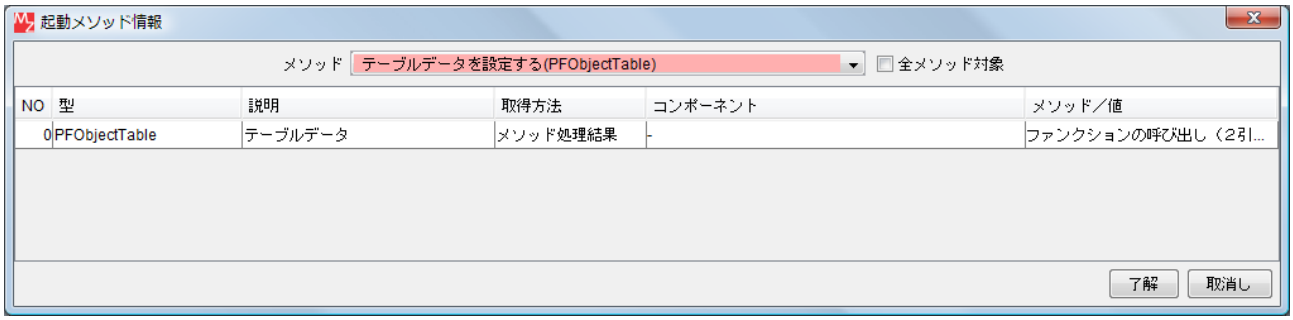
③の接続情報：ファンクションの戻り値としてテーブルデータを返す。



④の接続情報：定義した列削除処理を実行する。



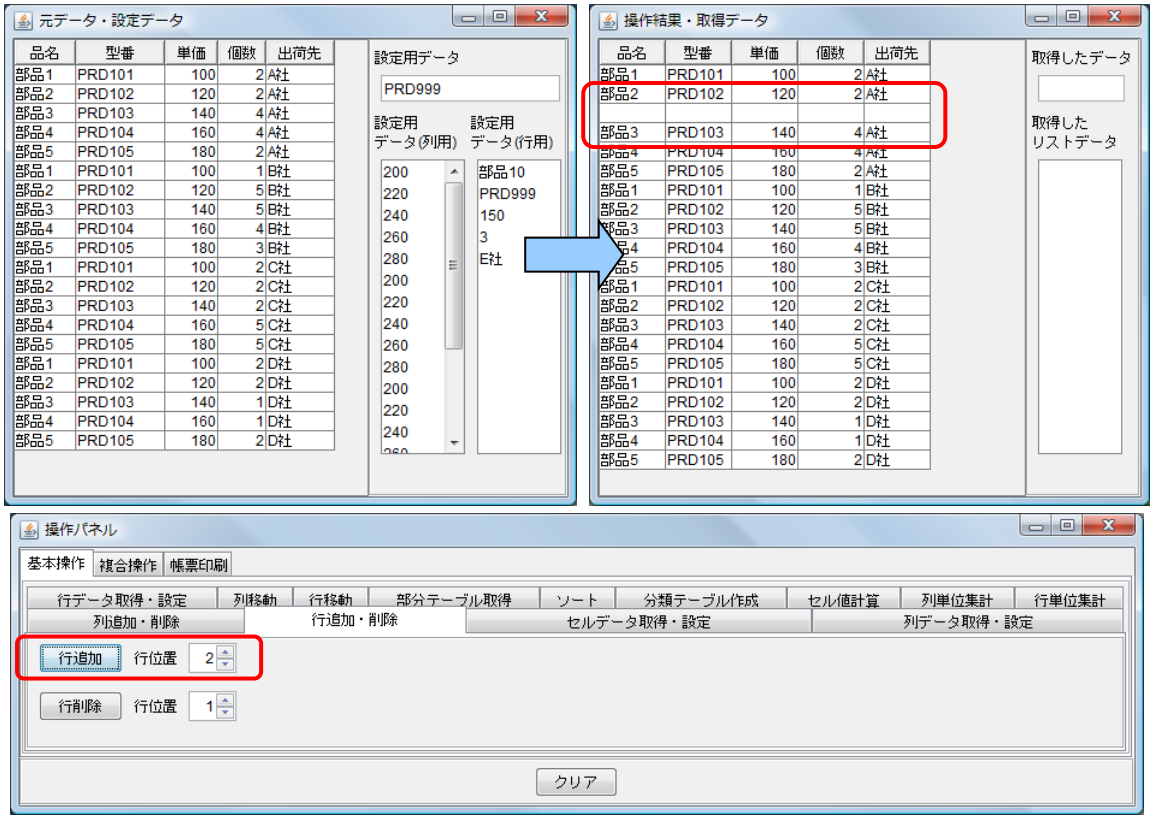
⑤の接続情報：結果をテーブルに設定する。



2.3 行の追加

動作イメージ

以下の図のように、テーブルデータに行を追加するための処理手順について説明します。



使用メソッド

テーブルデータに行を追加するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

コンポーネント	■テーブル格納変数
メソッド	指定位置に行を追加する(int)
引数	第 1 引数：追加位置
戻り値	なし
行の位置を指定して、行を追加します。	

類似する処理行うメソッドとして次のメソッドがあります。

メソッド	指定位置に行を追加する(データリスト指定)(int, PFObjectList)
引数	第 1 引数：追加位置 第 2 引数：データリスト
戻り値	なし
行の位置、行データを指定して、行を追加します	

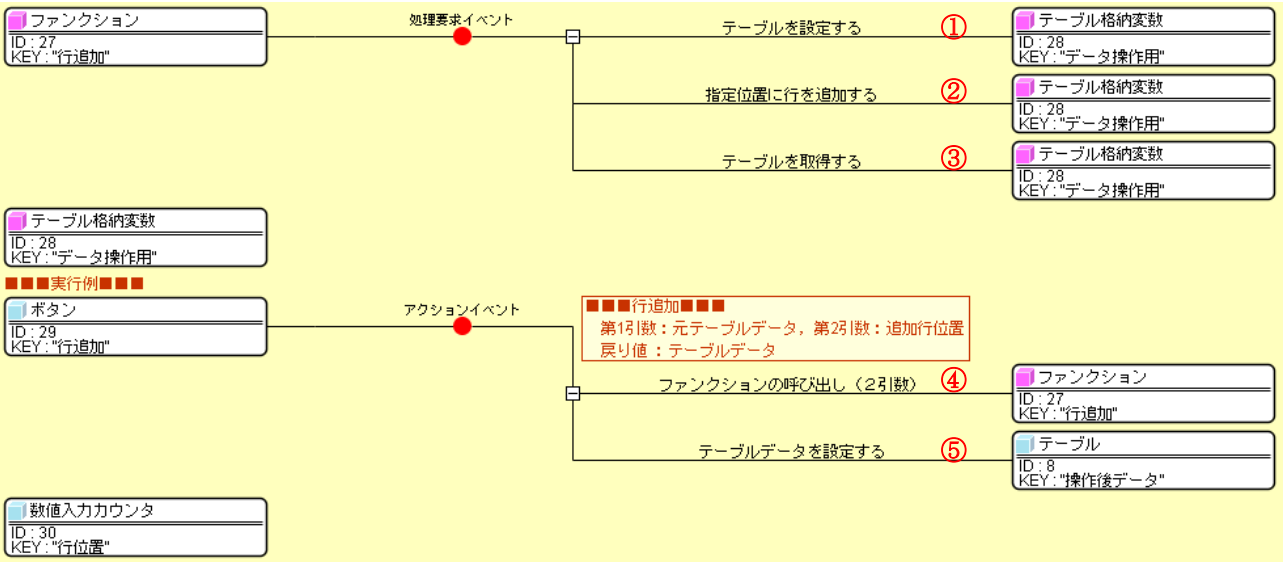
メソッド	最後尾に行を追加する
引数	なし
戻り値	なし
最終行に行を追加します。	

メソッド	最後尾に行を追加する(データリスト指定)(PFObjectList)
引数	第 1 引数：データリスト
戻り値	なし
行データを指定して最終行に行を追加します。	

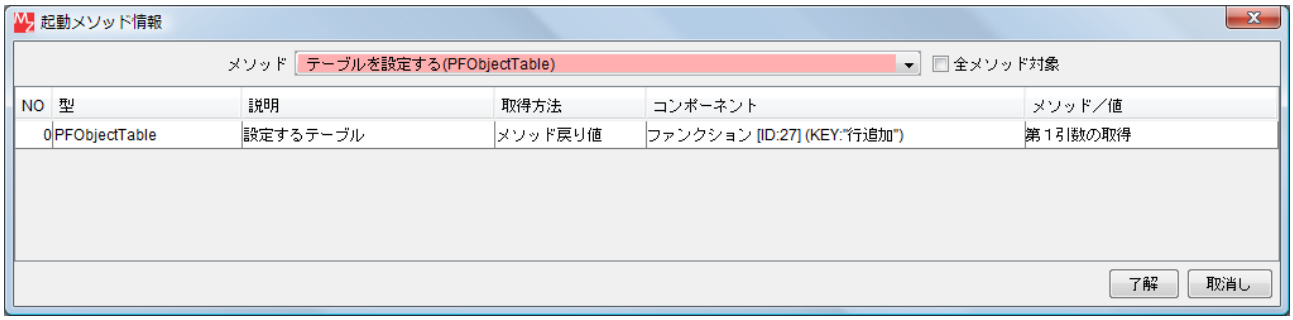
メソッド	最後尾に行を追加する(テーブル指定)(PFObjectTable)
引数	第 1 引数：テーブルデータ
戻り値	なし
テーブルデータを指定して最終行以降に行を追加します。	

接続例

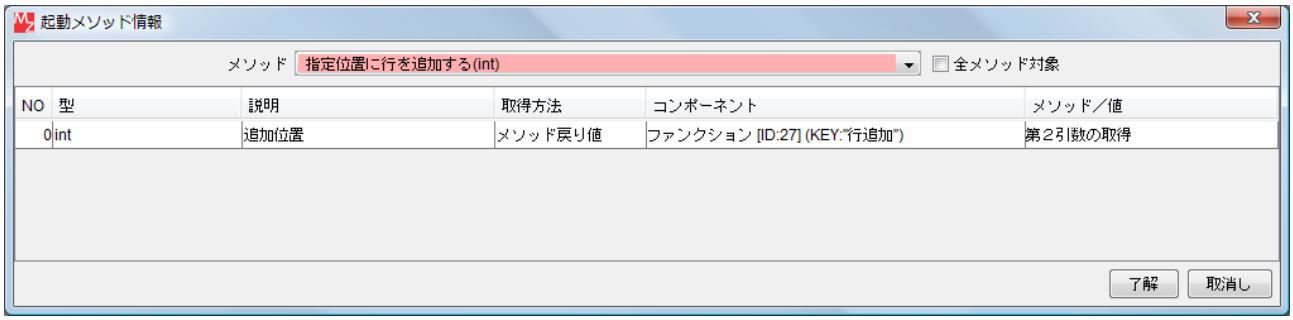
動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの行位置に行を追加し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで行追加の処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に追加行位置を指定してメソッド「ファンクションの呼び出し（2 引数)」を実行すると、結果のテーブルデータが戻り値として得られるように定義しています。



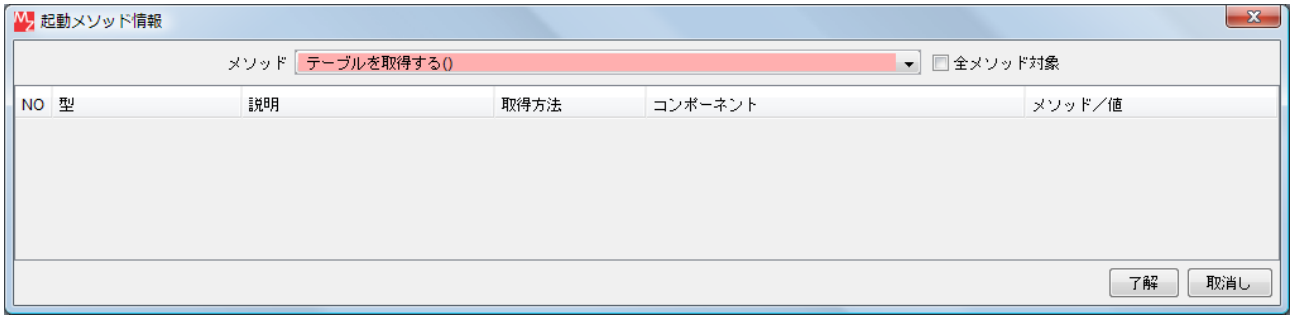
①の接続情報：処理対象のテーブルデータを設定する。



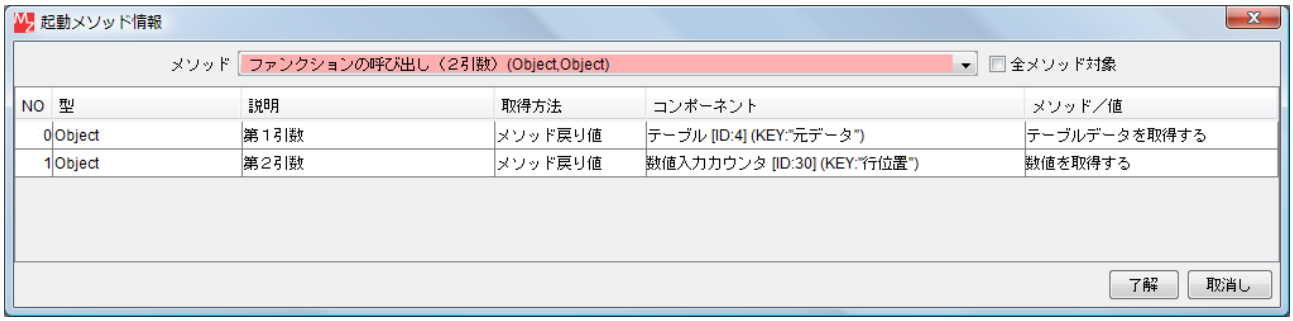
②の接続情報：追加する行位置（行インデックス）を指定して行を追加する。



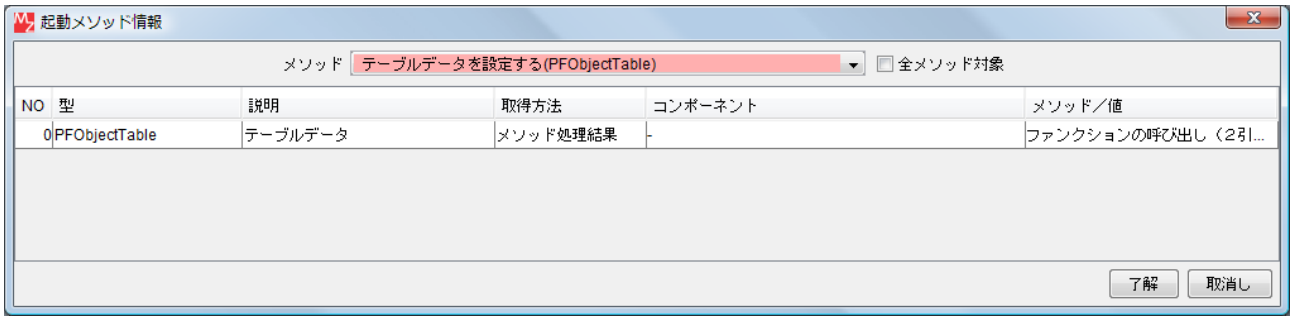
③の接続情報：ファンクションの戻り値としてテーブルデータを返す。



④の接続情報：定義した行追加処理を実行する。



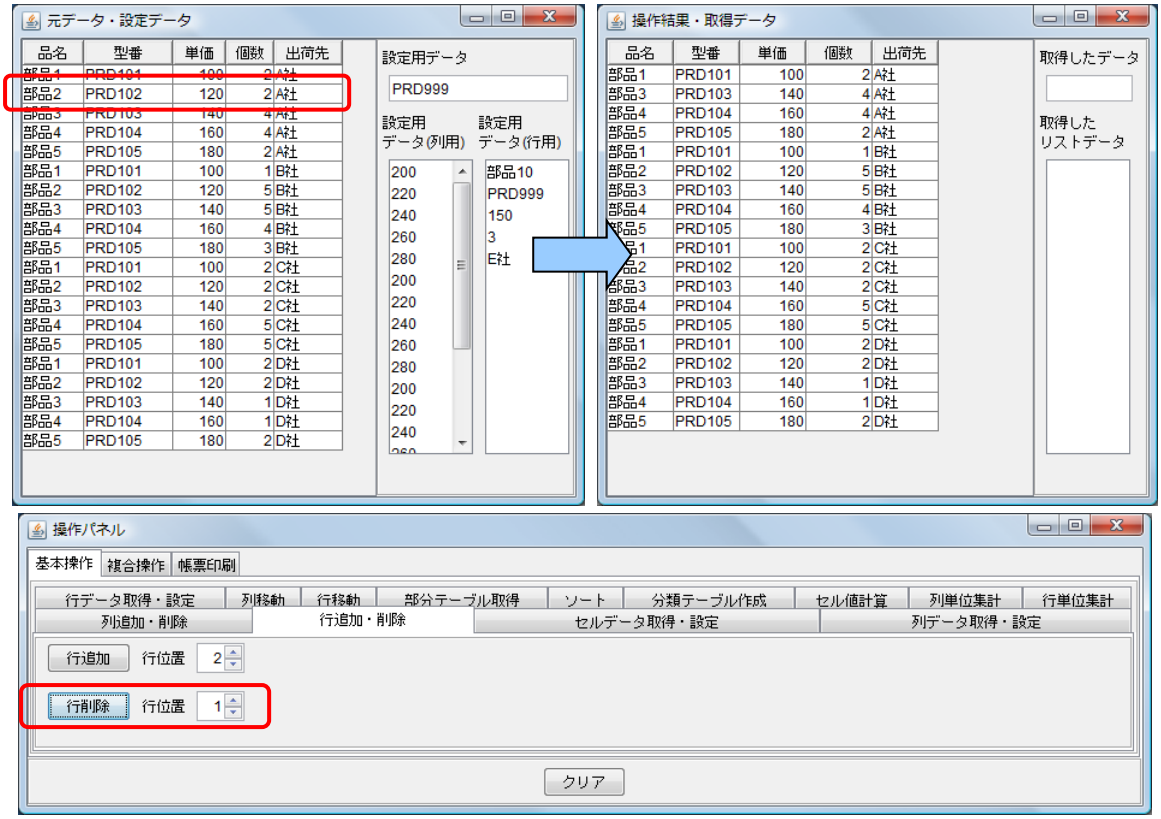
⑤の接続情報：結果をテーブルに設定する。



2.4 行の削除

動作イメージ

以下の図のように、テーブルデータの行を削除するための処理手順について説明します。



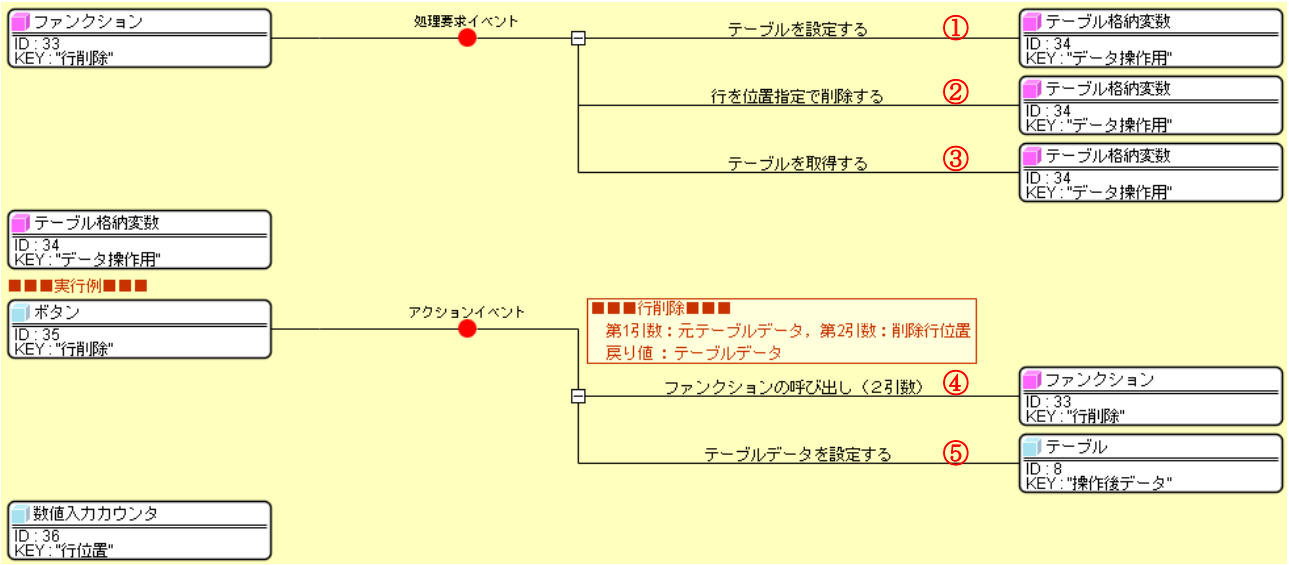
使用メソッド

テーブルデータの行を削除するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

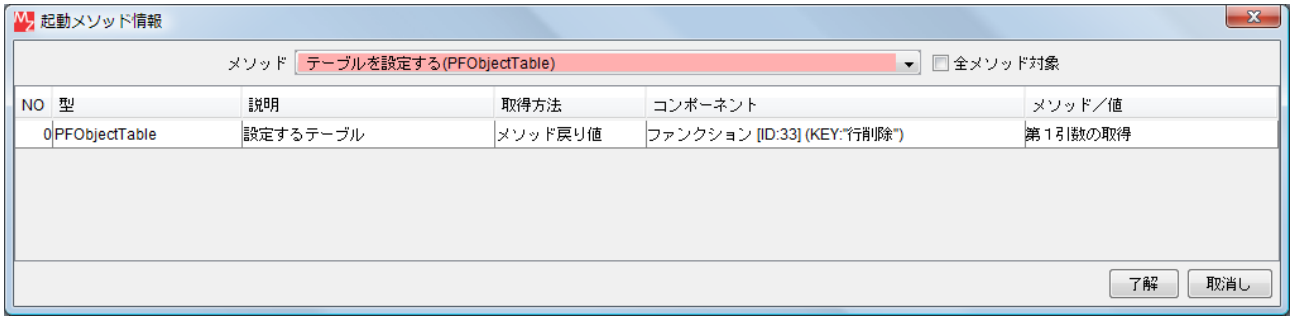
コンポーネント	■テーブル格納変数
メソッド	行を位置指定で削除する(int)
引数	第 1 引数：指定位置
戻り値	なし
行の位置を指定して、行を追加します。	

接続例

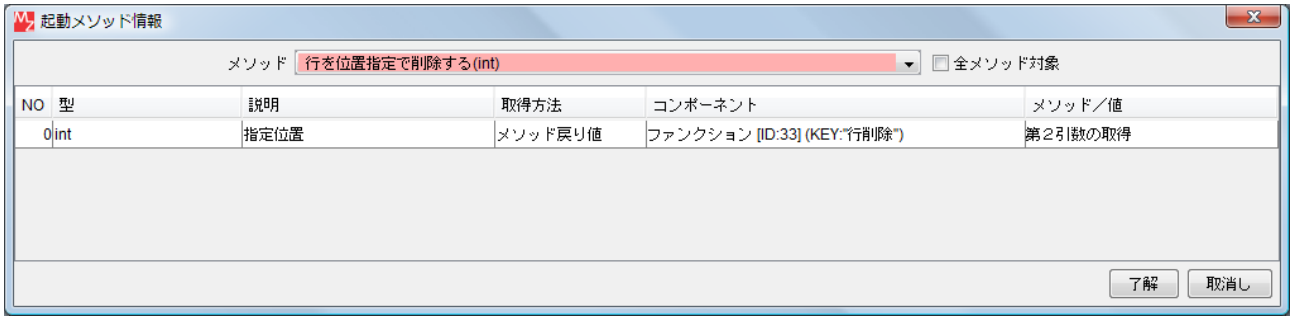
動作イメージのように、「数値入力カウンタ」で指定された位置の行を左側のテーブルデータから削除し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで行削除の処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に削除行位置を指定してメソッド「ファンクションの呼び出し（2 引数)」を実行すると、結果のテーブルデータが戻り値として得られるように定義しています。



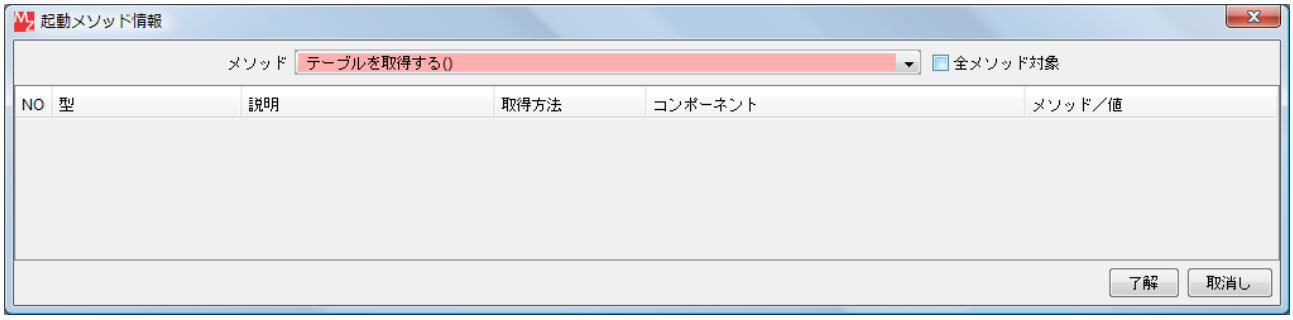
①の接続情報：処理対象のテーブルデータを設定する。



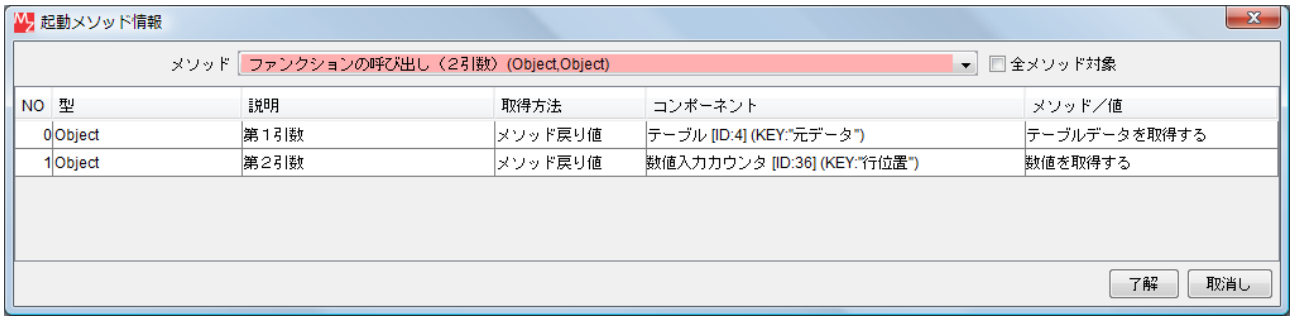
②の接続情報：削除する行位置（行インデックス）を指定して行を削除する。



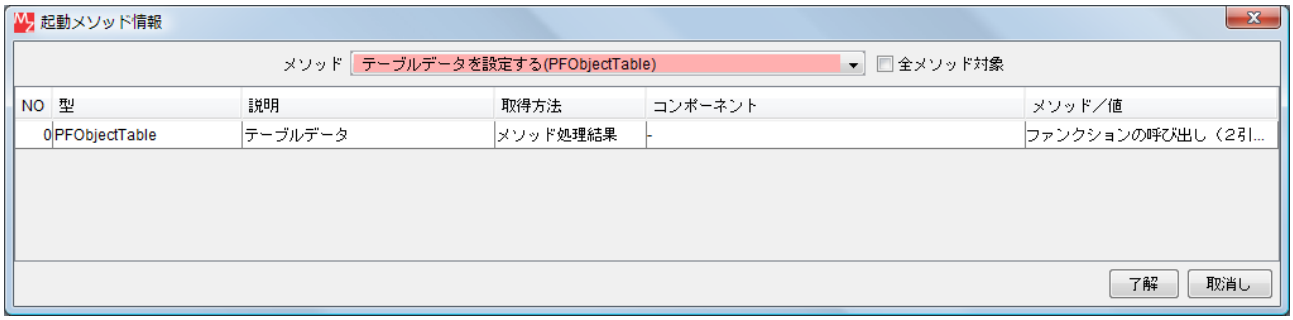
③の接続情報：ファンクションの戻り値としてテーブルデータを返す。



④の接続情報：定義した行削除処理を実行する。



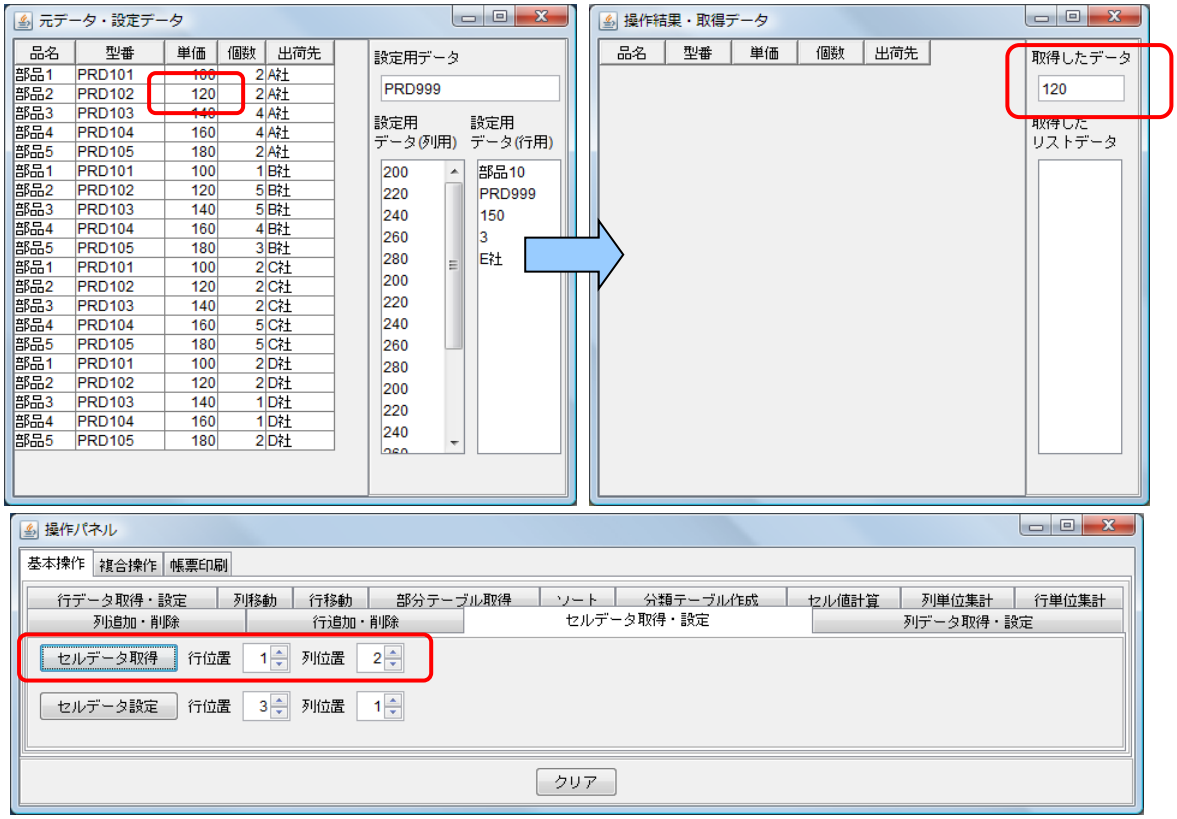
⑤の接続情報：結果をテーブルに設定する。



2.5 セルデータの取得

動作イメージ

以下の図のように、テーブルデータからセルデータを取得するための処理手順について説明します。



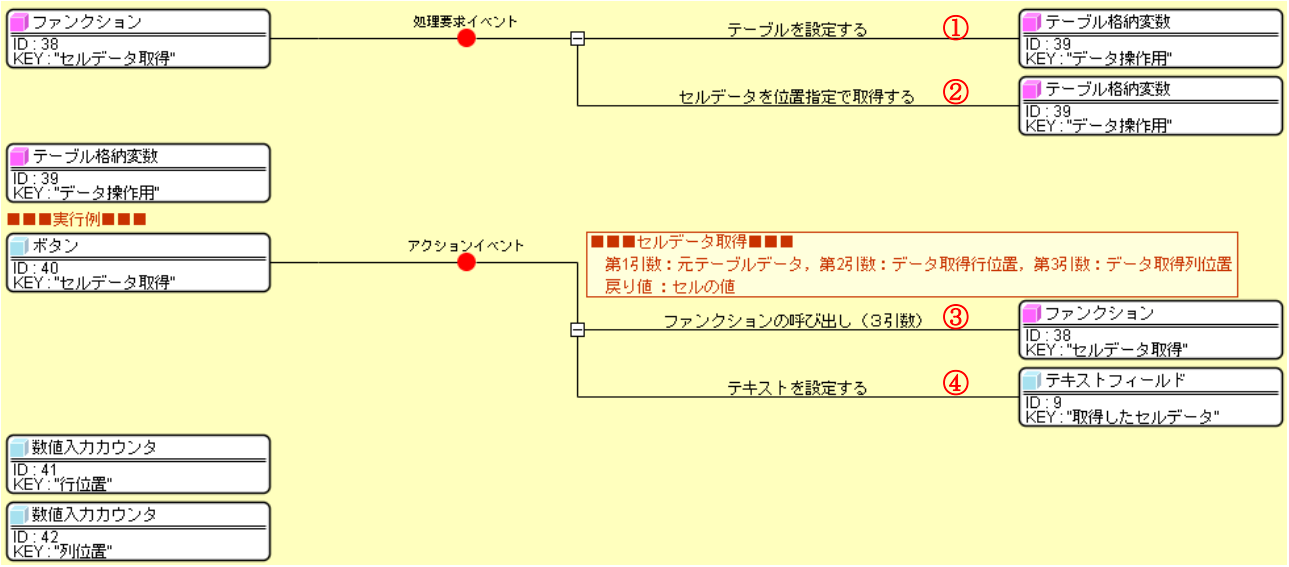
使用メソッド

テーブルデータからセルデータを取得するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

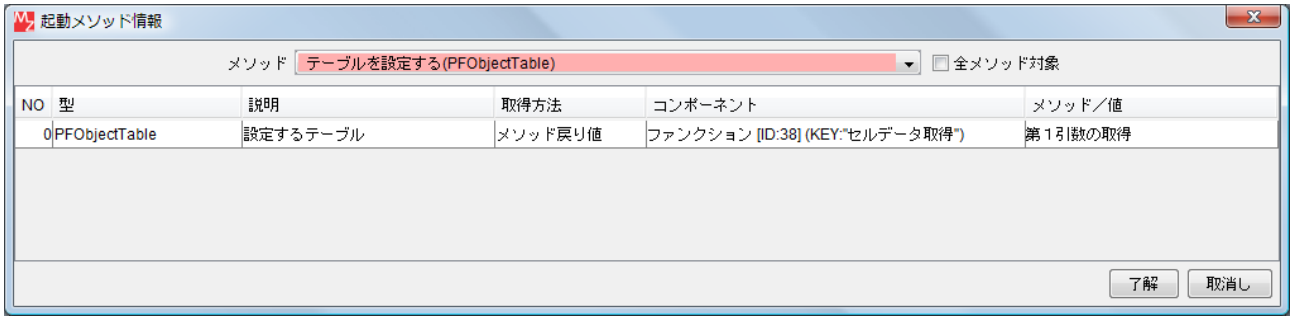
コンポーネント	■テーブル格納変数
メソッド	セルデータを位置指定で取得する(int, int)
引数	第 1 引数：行位置 第 2 引数：列位置
戻り値	Object
行位置および列位置を指定してセルデータを取得します。	

接続例

動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの行位置・列位置から、その位置のセルの値を取得し「テキストフィールド」に設定する処理手順を以下のように定義します。ここでセルデータ取得処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に行位置、第 3 引数に列位置を指定してメソッド「ファンクションの呼び出し（3 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。

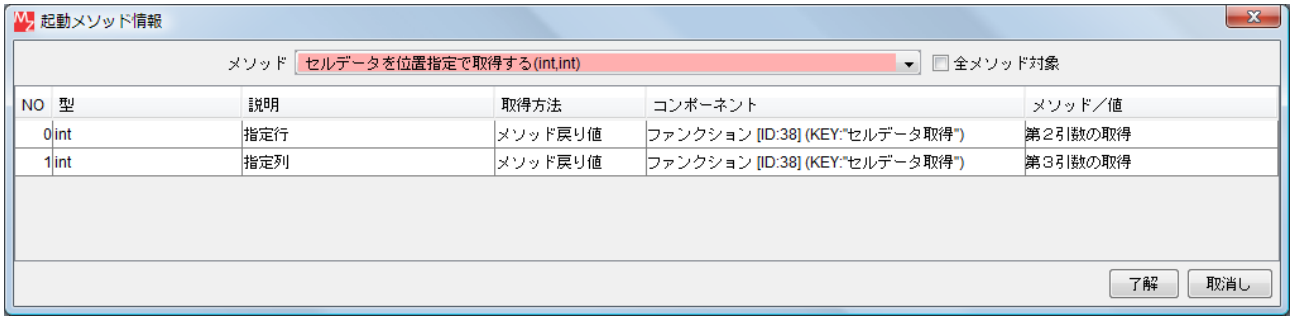


①の接続情報：処理対象のテーブルデータを設定する。



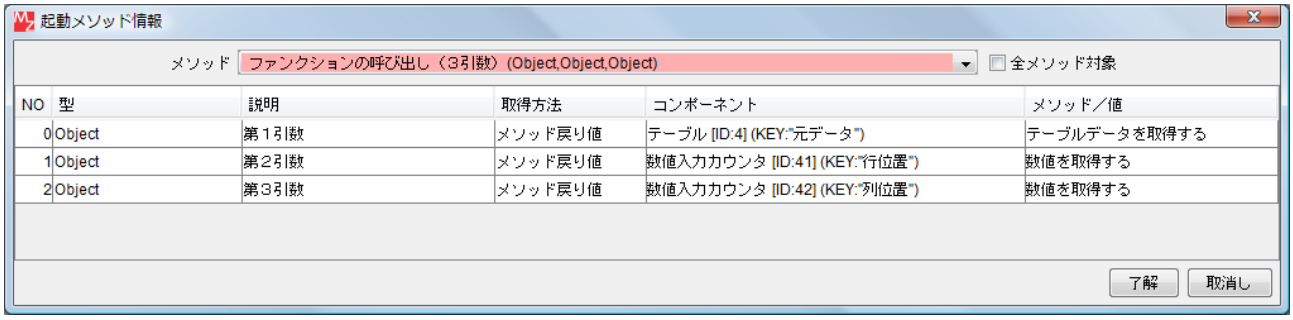
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	設定するテーブル	メソッド戻り値	ファンクション [ID:38] (KEY:"セルデータ取得")	第１引数の取得

②の接続情報：行位置、列位置を指定してセルデータを取得し、ファンクションの戻り値として返す。



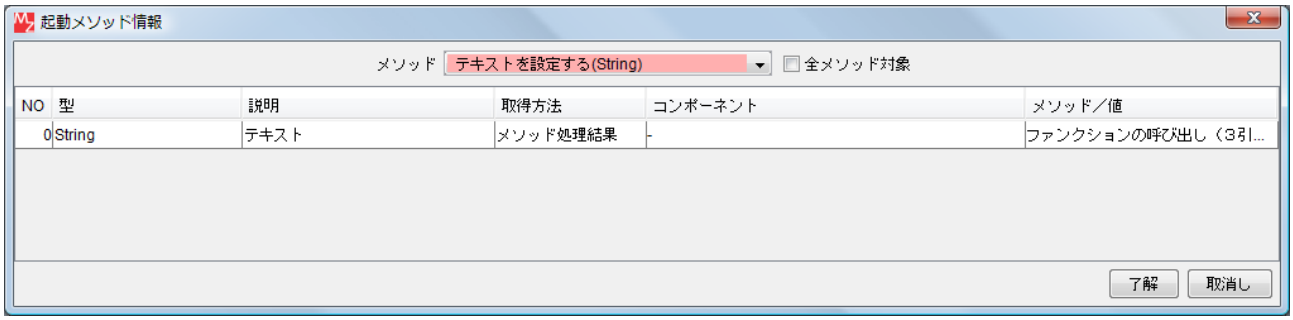
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	指定行	メソッド戻り値	ファンクション [ID:38] (KEY:"セルデータ取得")	第２引数の取得
1	int	指定列	メソッド戻り値	ファンクション [ID:38] (KEY:"セルデータ取得")	第３引数の取得

③の接続情報：定義したセルデータ取得処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第１引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第２引数	メソッド戻り値	数値入力カウンタ [ID:41] (KEY:"行位置")	数値を取得する
2	Object	第３引数	メソッド戻り値	数値入力カウンタ [ID:42] (KEY:"列位置")	数値を取得する

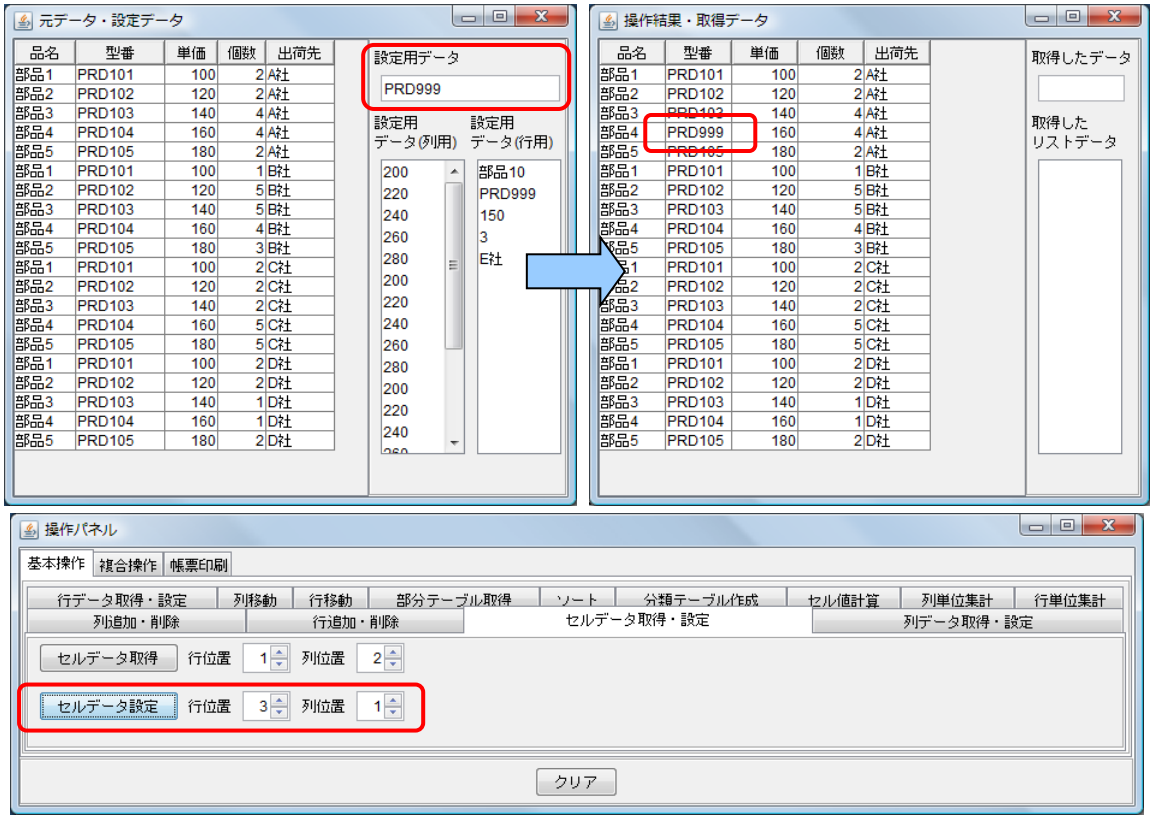
④の接続情報：結果をテキストフィールドに設定する。



2.6 セルデータの設定

動作イメージ

以下の図のように、テーブルデータのセルにデータを設定するための処理手順について説明します。



使用メソッド

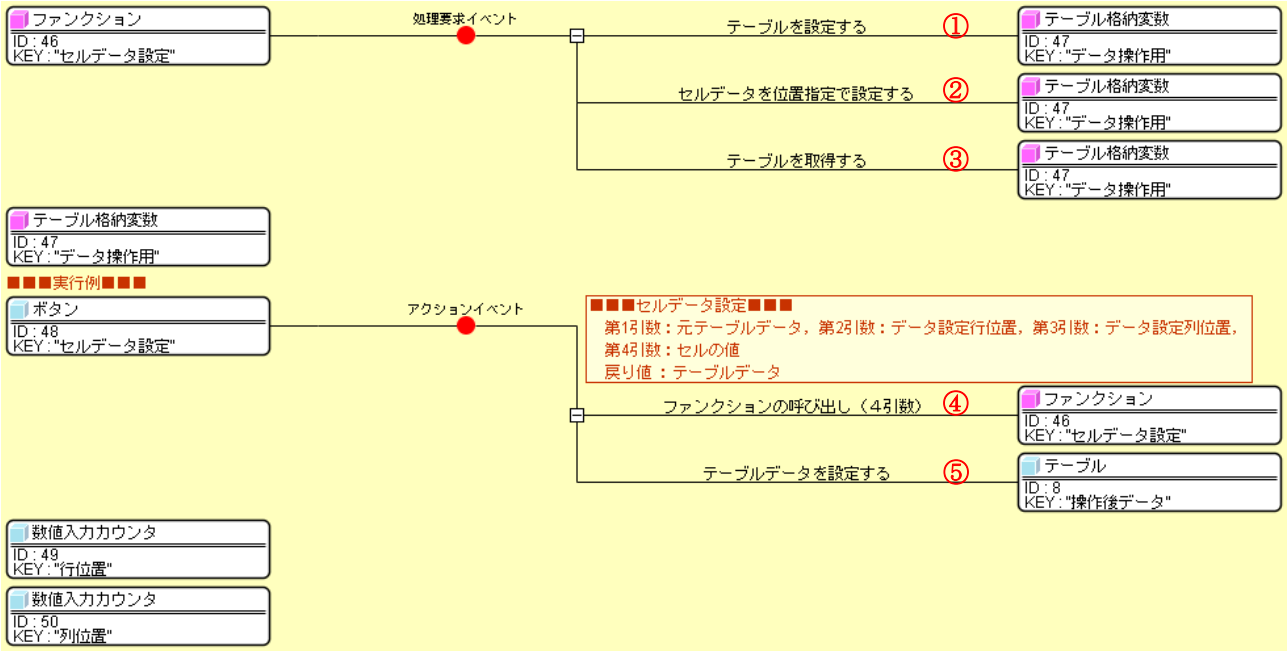
テーブルデータのセルにデータを設定するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

コンポーネント	■テーブル格納変数
メソッド	セルデータを位置指定で設定する(int, int, Object)
引数	第 1 引数：行位置 第 2 引数：列位置 第 3 引数：データ
戻り値	なし
行位置および列位置を指定してセルデータを設定します。	

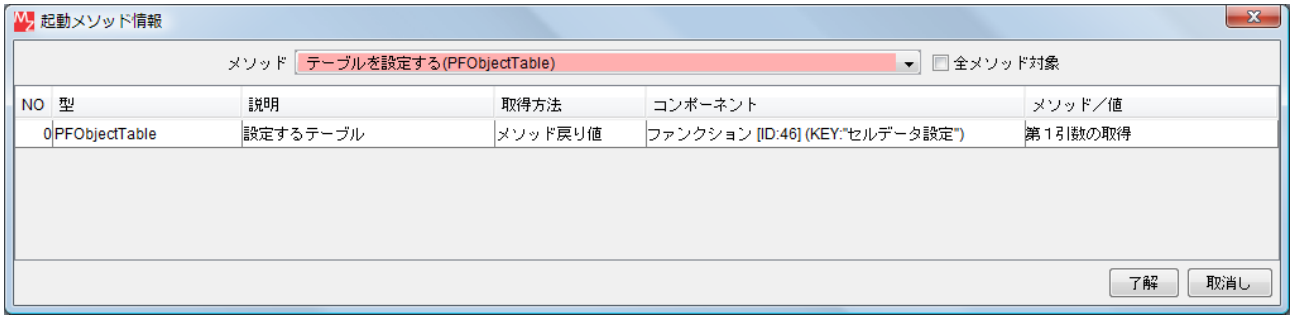
接続例

動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの行位置・列位置のセルに値を設定し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここでセルデータ設定処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に行位置、第 3 引数に列位置、第 4 引数に設定データを指定してメソッド「ファン

クションの呼び出し（4 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。

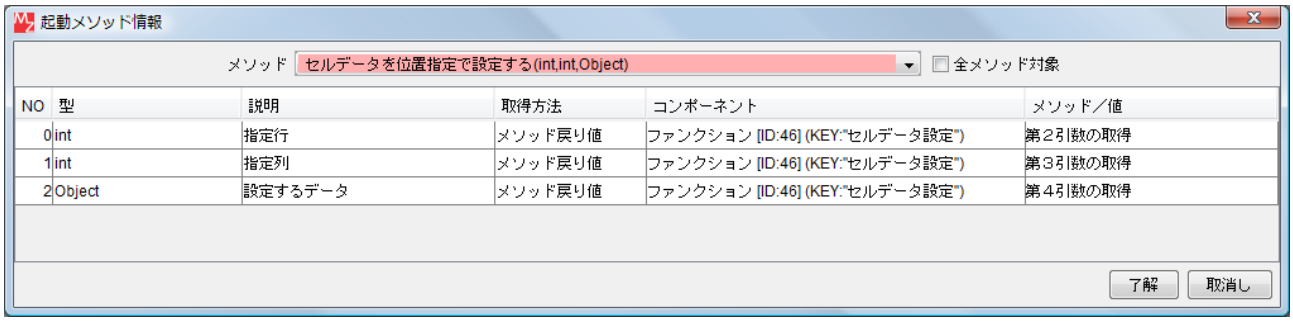


①の接続情報：処理対象のテーブルデータを設定する。



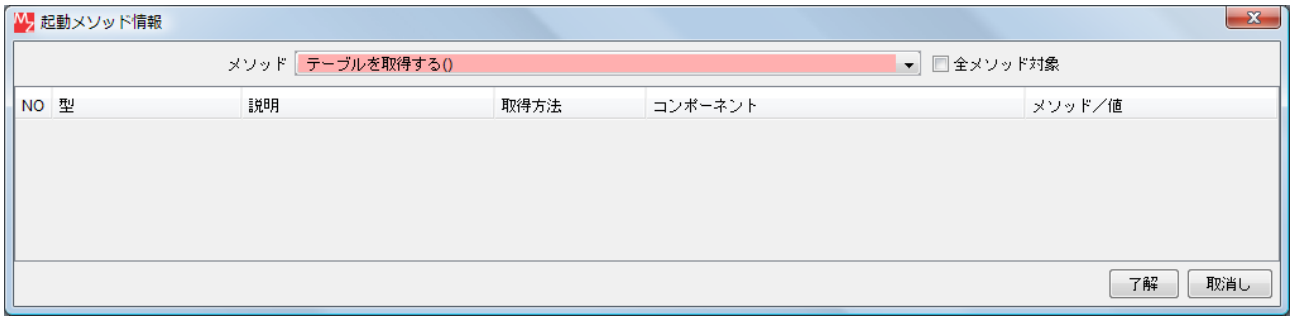
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	設定するテーブル	メソッド戻り値	ファンクション [ID:46] (KEY:"セルデータ設定")	第１引数の取得

②の接続情報：行位置、列位置を指定してセルデータを設定する。

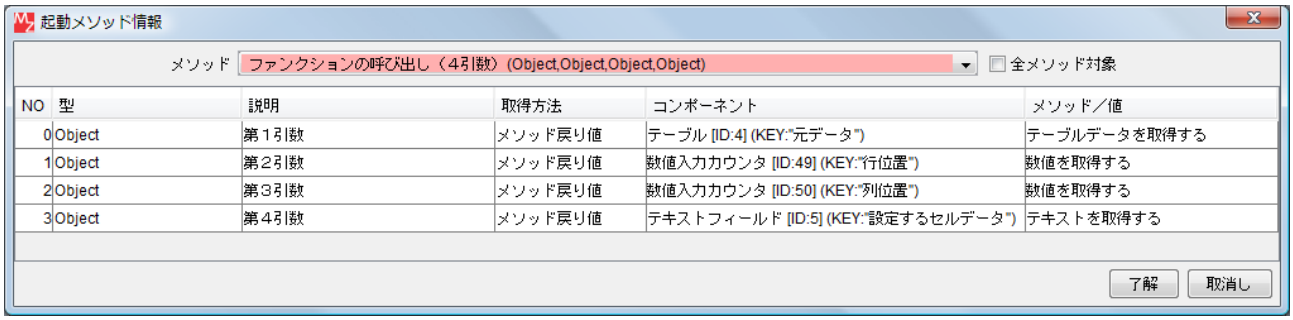


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	指定行	メソッド戻り値	ファンクション [ID:46] (KEY:"セルデータ設定")	第２引数の取得
1	int	指定列	メソッド戻り値	ファンクション [ID:46] (KEY:"セルデータ設定")	第３引数の取得
2	Object	設定するデータ	メソッド戻り値	ファンクション [ID:46] (KEY:"セルデータ設定")	第４引数の取得

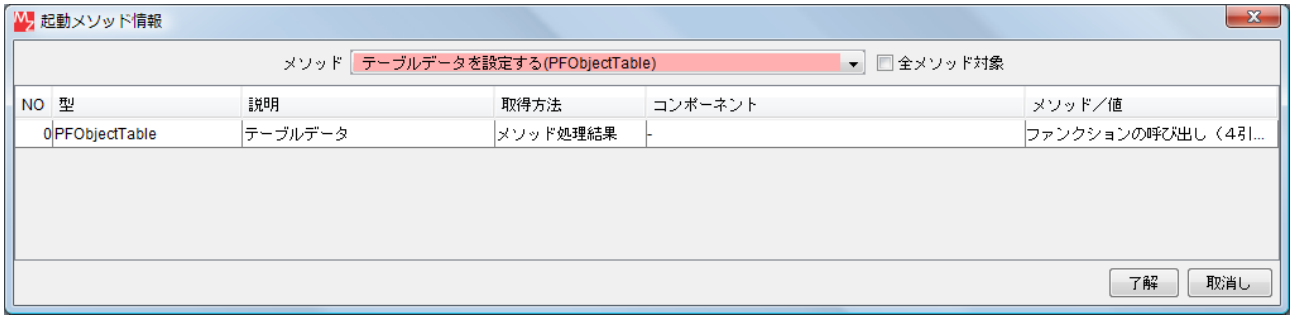
③の接続情報：ファンクションの戻り値としてテーブルデータを返す。



④の接続情報：定義したセルデータ設定処理を実行する。



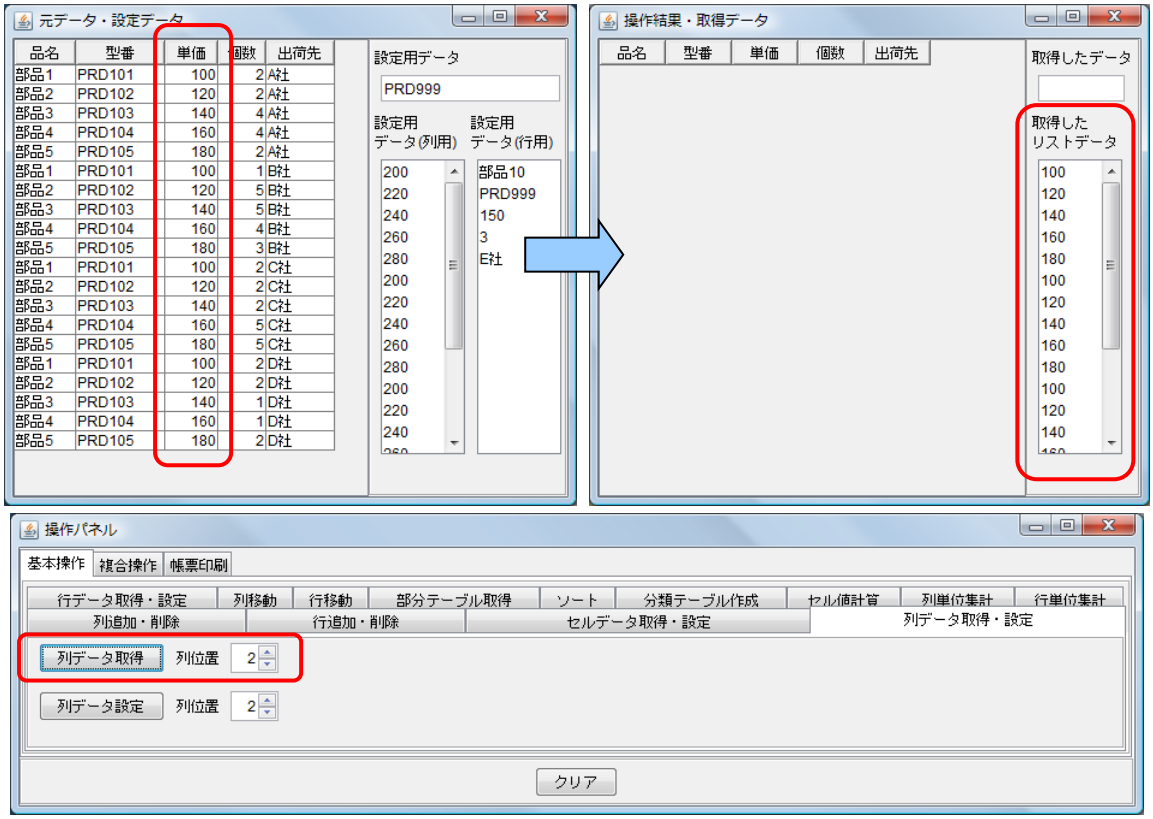
⑤の接続情報：結果をテーブルに設定する。



2.7 列データの取得

動作イメージ

以下の図のように、テーブルデータから列データを取得するための処理手順について説明します。



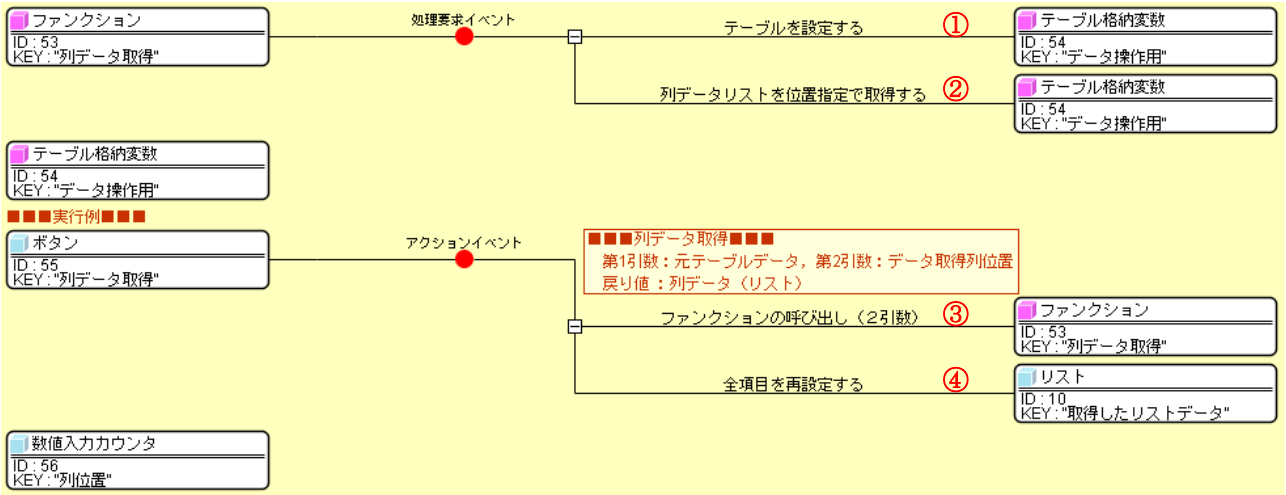
使用メソッド

テーブルデータから列データを取得するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

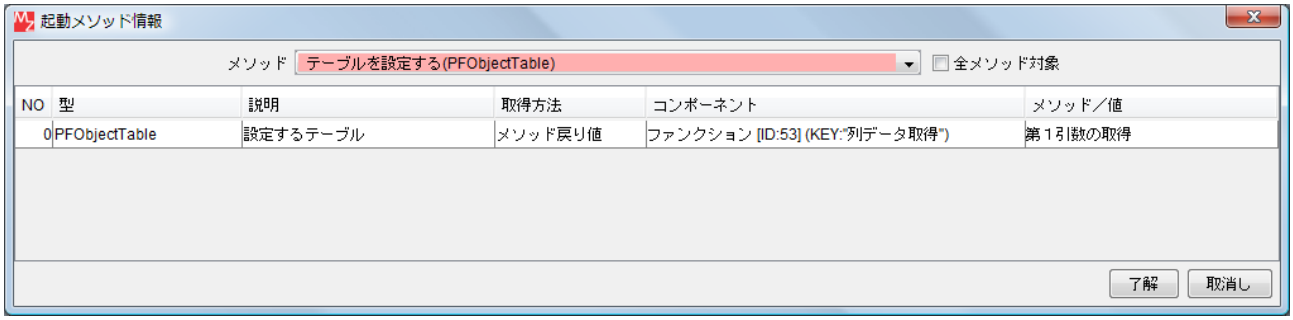
コンポーネント	■テーブル格納変数
メソッド	列データリストを位置指定で取得する(int)
引数	第 1 引数：列位置
戻り値	PFObjectList
列位置を指定して列データ（リスト）を取得します。	

接続例

動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの列位置から、その位置の列データをリストとして取得し「リスト」に設定する処理手順を以下のように定義します。ここで列データ取得処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に列位置を指定してメソッド「ファンクションの呼び出し（2 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。

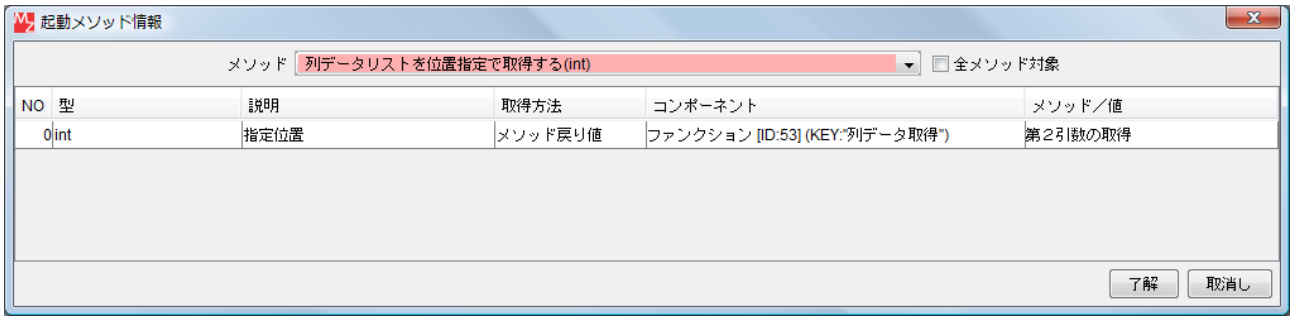


①の接続情報：処理対象のテーブルデータを設定する。



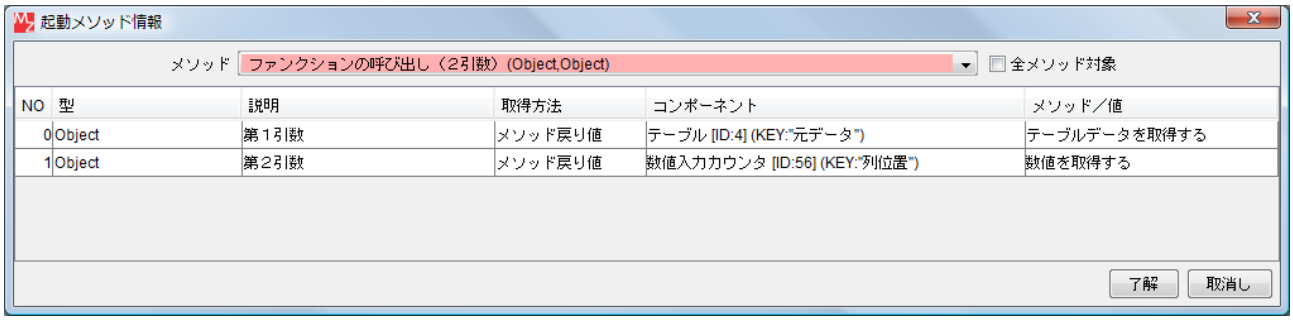
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	設定するテーブル	メソッド戻り値	ファンクション [ID:53] (KEY:"列データ取得")	第１引数の取得

②の接続情報：列位置を指定して列データを取得し、ファンクションの戻り値として返す。



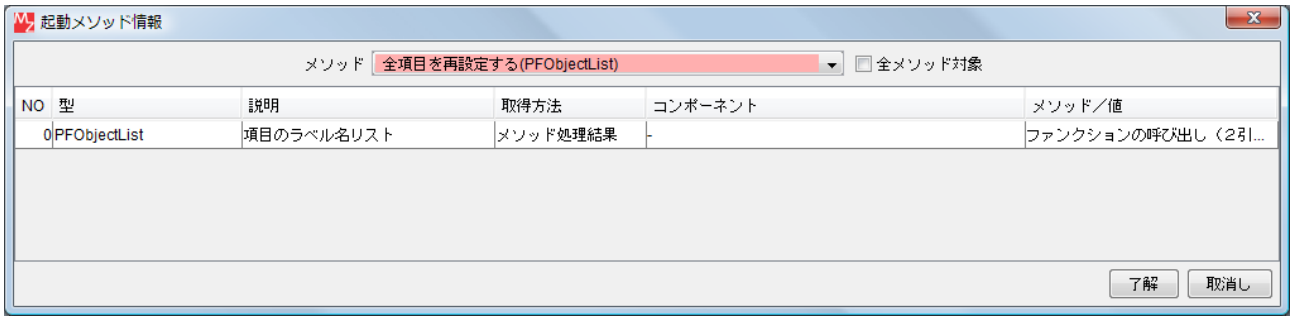
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	指定位置	メソッド戻り値	ファンクション [ID:53] (KEY:"列データ取得")	第２引数の取得

③の接続情報：定義した列データ取得処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第１引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第２引数	メソッド戻り値	数値入力カウンタ [ID:56] (KEY:"列位置")	数値を取得する

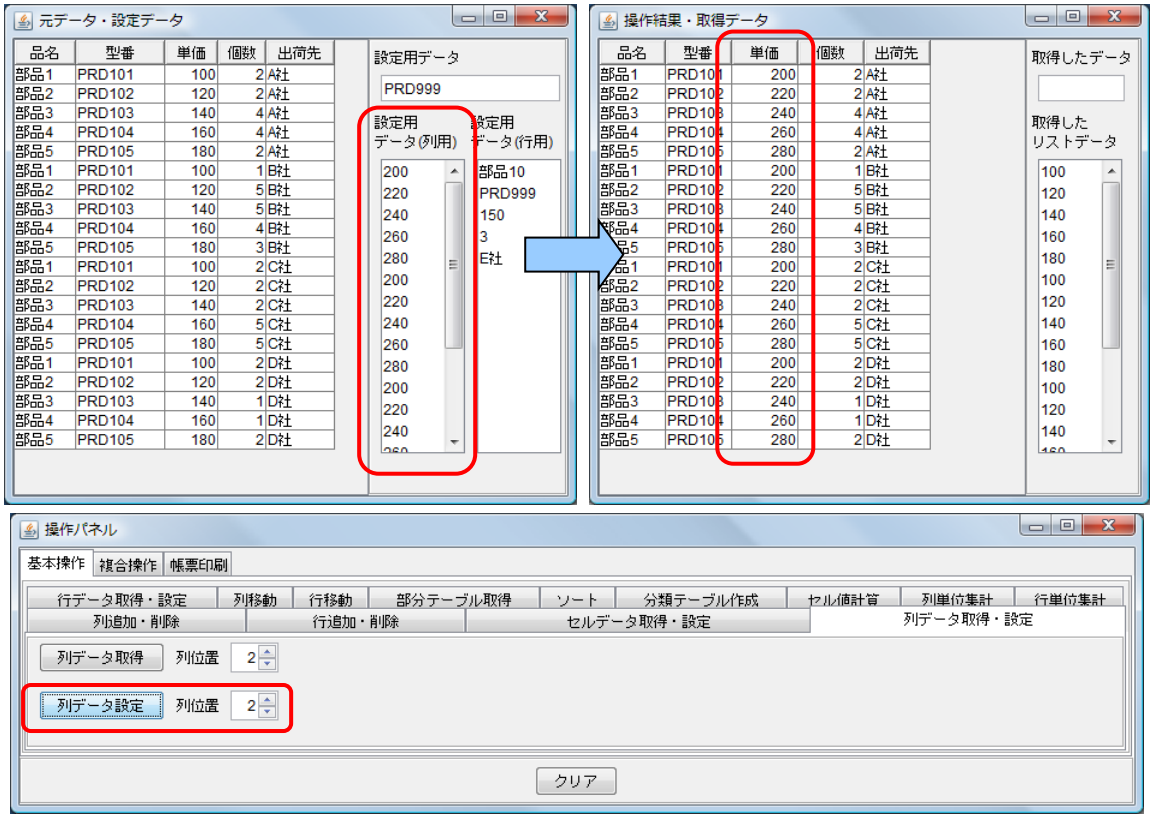
④の接続情報：結果をリストに設定する。



2.8 列データの設定

動作イメージ

以下の図のように、テーブルデータに列データを設定するための処理手順について説明します。



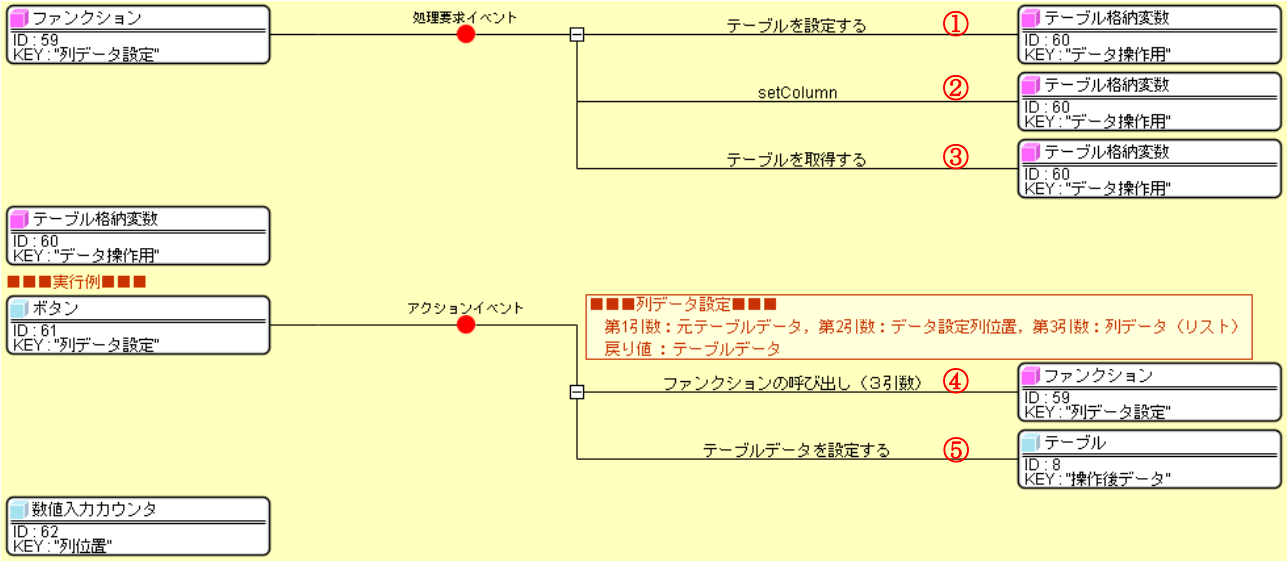
使用メソッド

テーブルデータに列データを設定するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

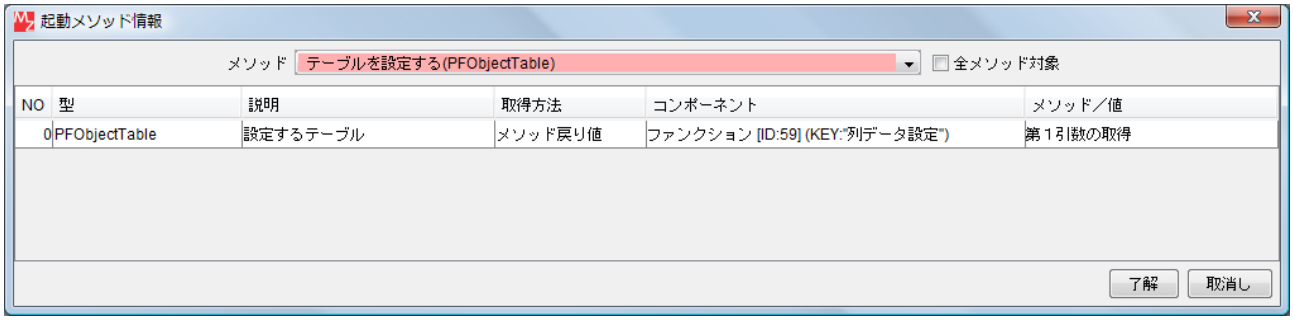
コンポーネント	■テーブル格納変数
メソッド	setColumn(int, PFObjectList)
引数	第 1 引数：列位置 第 2 引数：列データ（リスト）
戻り値	なし
列位置を指定して列データを設定します。	

接続例

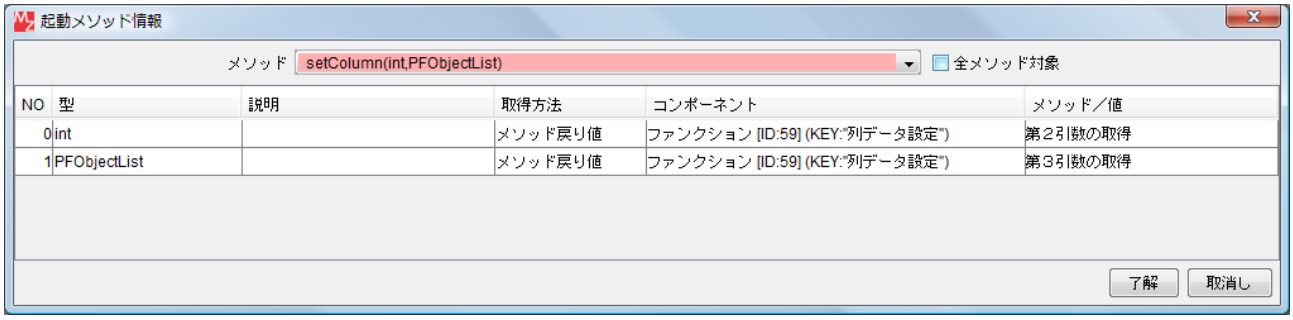
動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの列位置に列データを設定し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで列データ設定処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に列位置、第 3 引数に設定データを指定してメソッド「ファンクションの呼び出し（3 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。



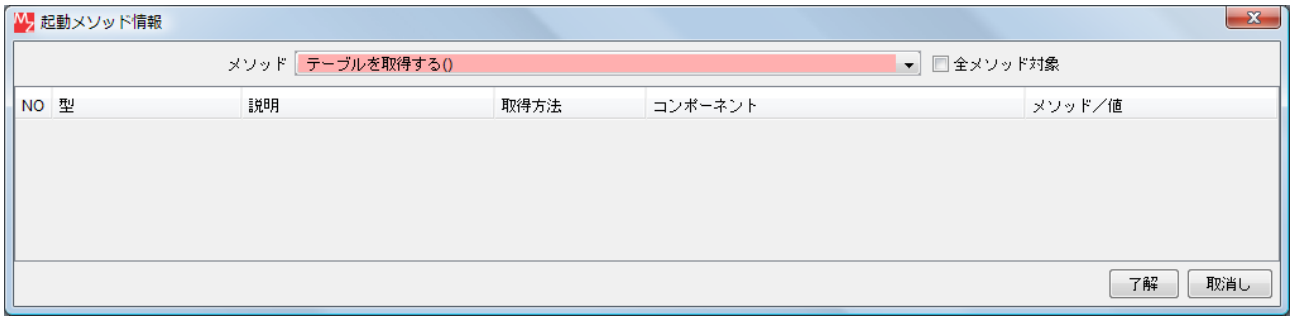
①の接続情報：処理対象のテーブルデータを設定する。



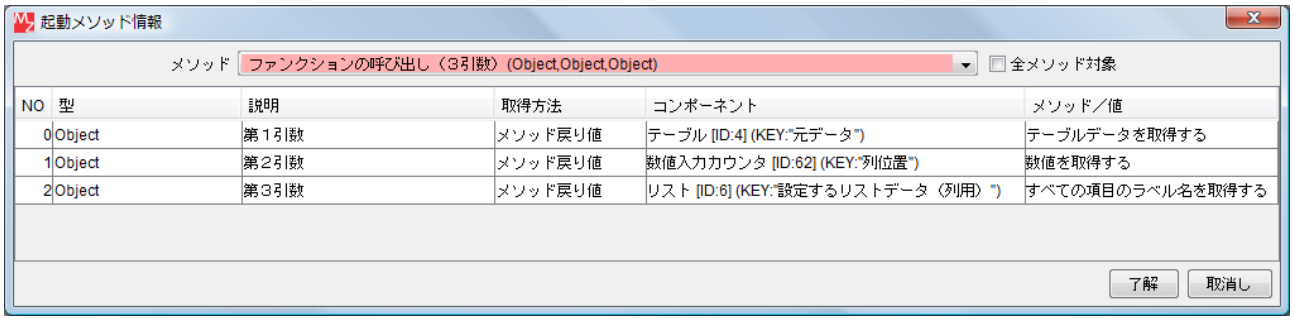
②の接続情報：列位置を指定して列データを設定する。



③の接続情報：ファンクションの戻り値としてテーブルデータを返す。

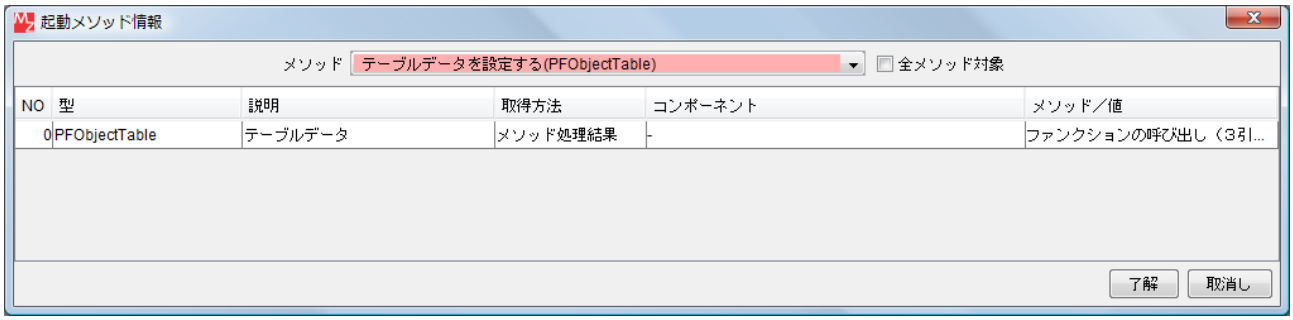


④の接続情報：定義した列データ設定処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	メソッド戻り値	数値入力カウンタ [ID:62] (KEY:"列位置")	数値を取得する
2	Object	第3引数	メソッド戻り値	リスト [ID:6] (KEY:"設定するリストデータ（列用）")	すべての項目のラベル名を取得する

⑤の接続情報：結果をテーブルに設定する。

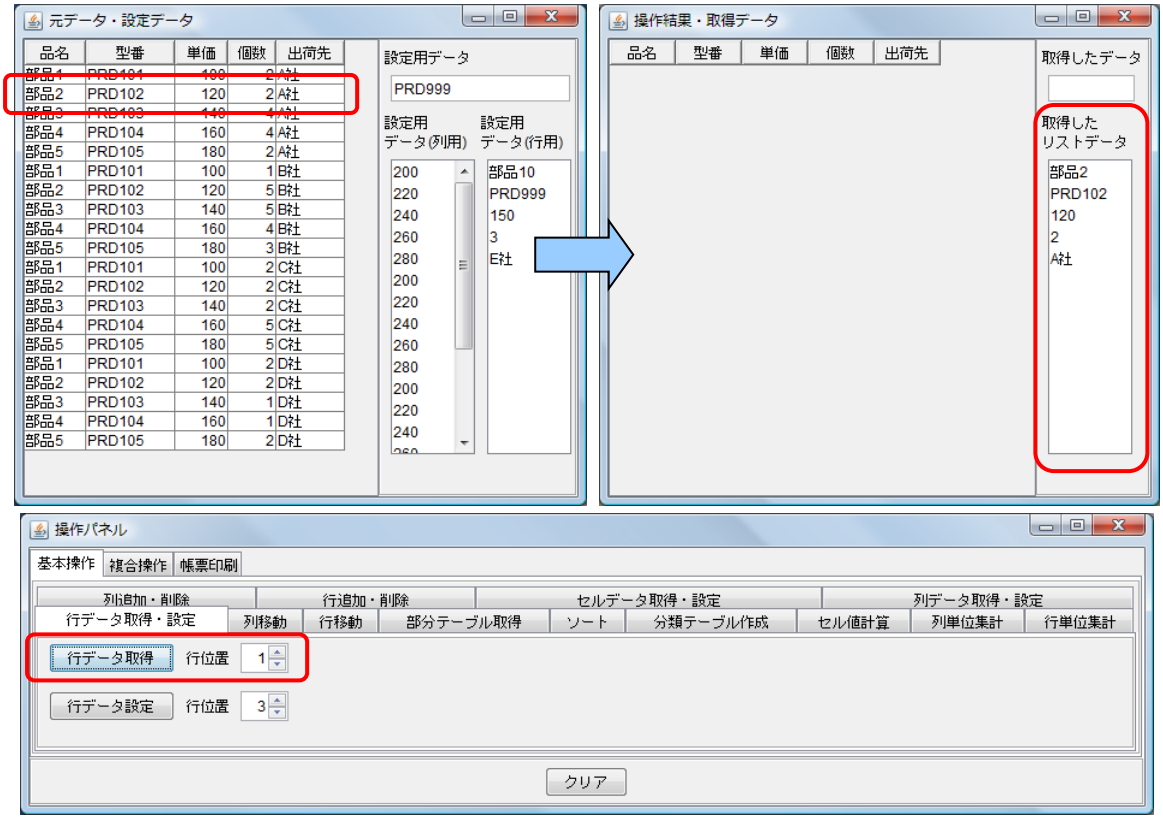


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（3引...

2.9 行データの取得

動作イメージ

以下の図のように、テーブルデータから行データを取得するための処理手順について説明します。



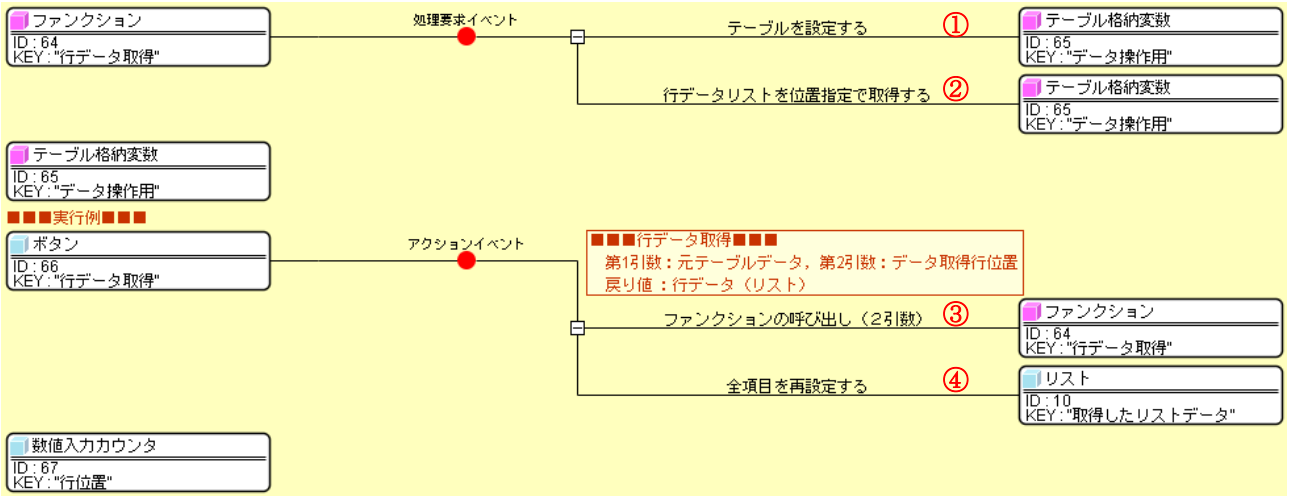
使用メソッド

テーブルデータから行データを取得するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

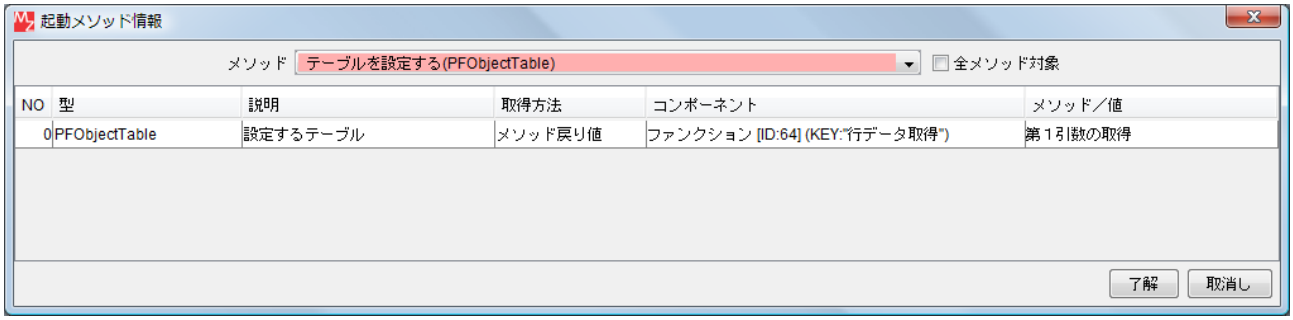
コンポーネント	■テーブル格納変数
メソッド	行データリストを位置指定で取得する(int)
引数	第 1 引数：行位置
戻り値	PFObjectList
行位置を指定して行データ（リスト）を取得します。	

接続例

動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの行位置から、その位置の行データをリストとして取得し「リスト」に設定する処理手順を以下のように定義します。ここで行データ取得処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に行位置を指定してメソッド「ファンクションの呼び出し（2 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。

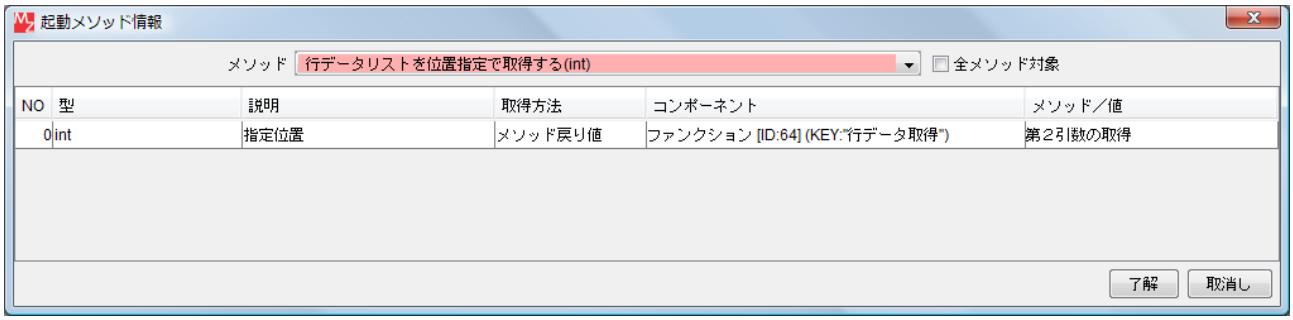


①の接続情報：処理対象のテーブルデータを設定する。



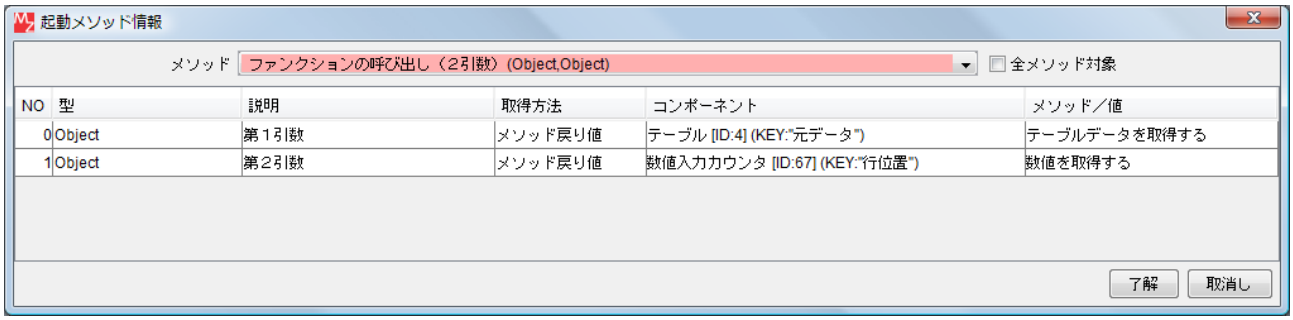
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	設定するテーブル	メソッド戻り値	ファンクション [ID:64] (KEY:"行データ取得")	第１引数の取得

②の接続情報：行位置を指定して行データを取得し、ファンクションの戻り値として返す。



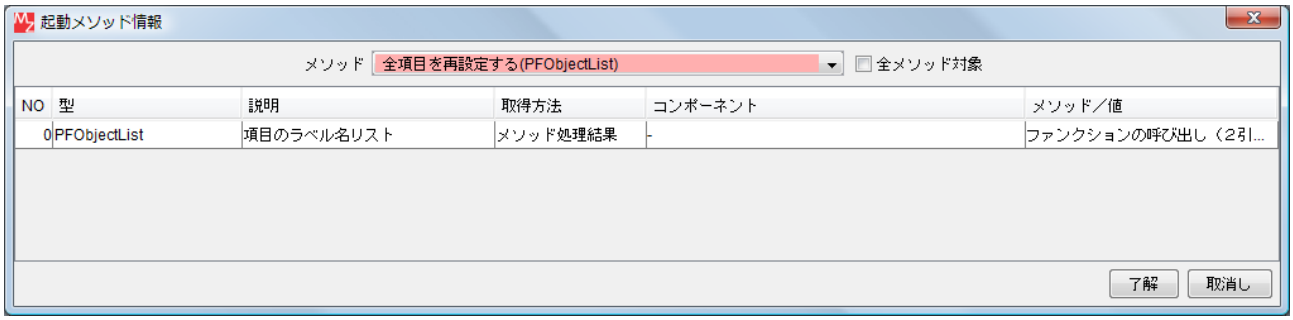
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	指定位置	メソッド戻り値	ファンクション [ID:64] (KEY:"行データ取得")	第２引数の取得

③の接続情報：定義した行データ取得処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第１引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第２引数	メソッド戻り値	数値入力カウンタ [ID:67] (KEY:"行位置")	数値を取得する

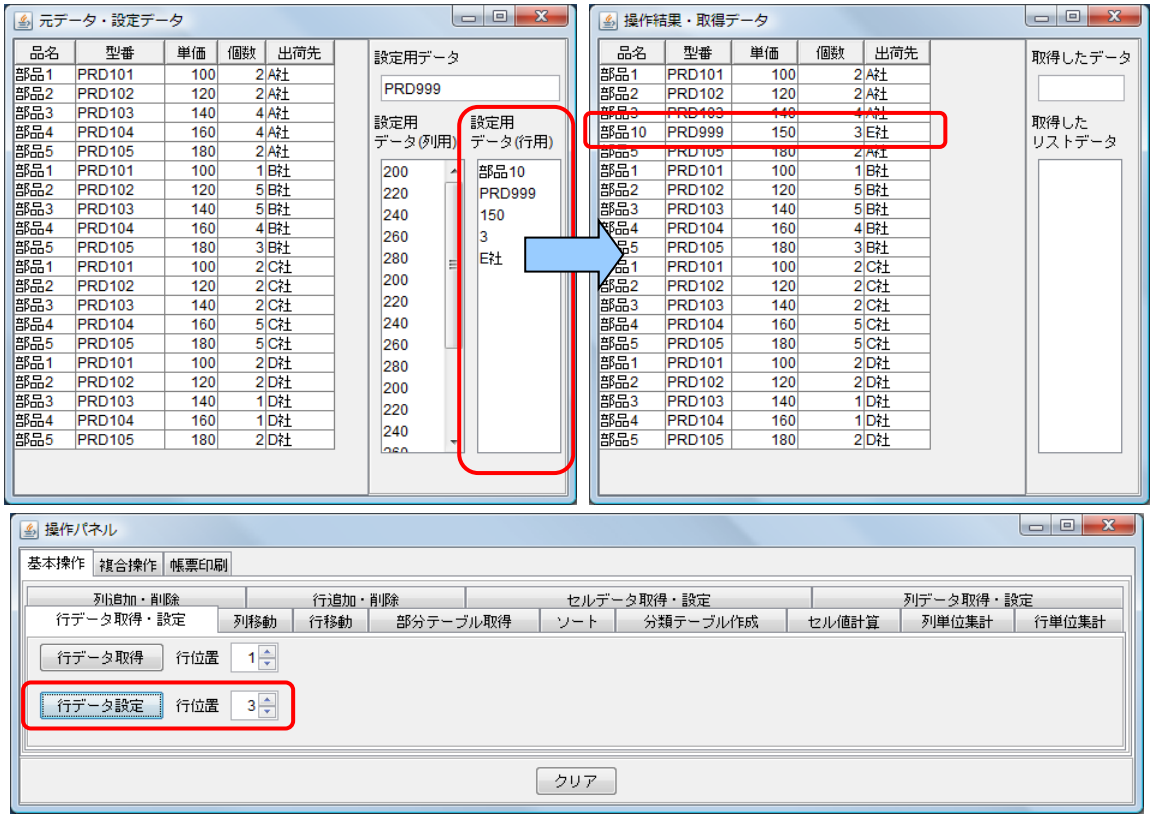
④の接続情報：結果をリストに設定する。



2.10 行データの設定

動作イメージ

以下の図のように、テーブルデータに行データを設定するための処理手順について説明します。



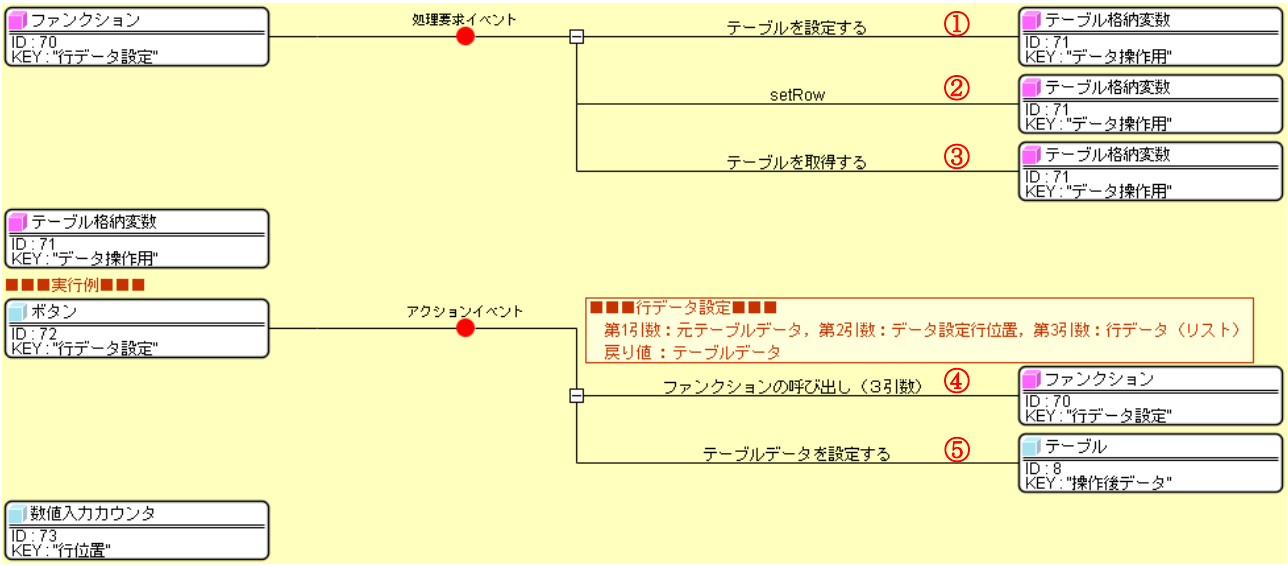
使用メソッド

テーブルデータに行データを設定するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

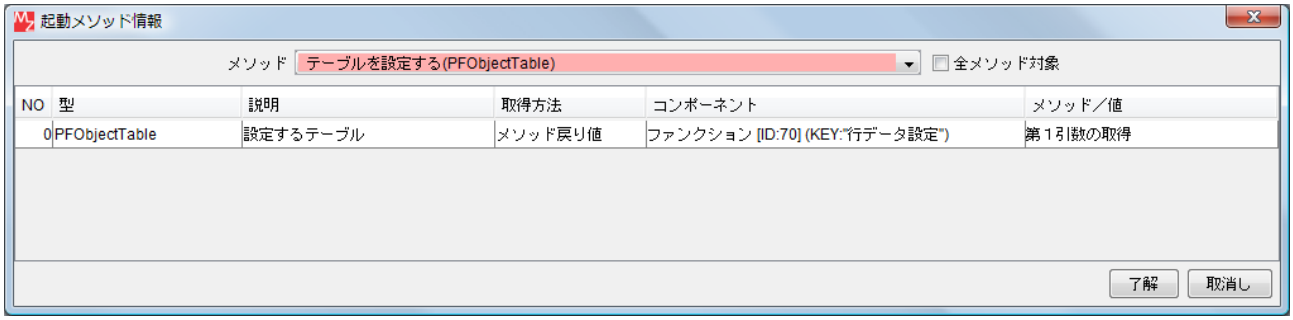
コンポーネント	■テーブル格納変数
メソッド	setRow(int, PFObjectList)
引数	第 1 引数：行位置 第 2 引数：行データ（リスト）
戻り値	なし
行位置を指定して行データを設定します。	

接続例

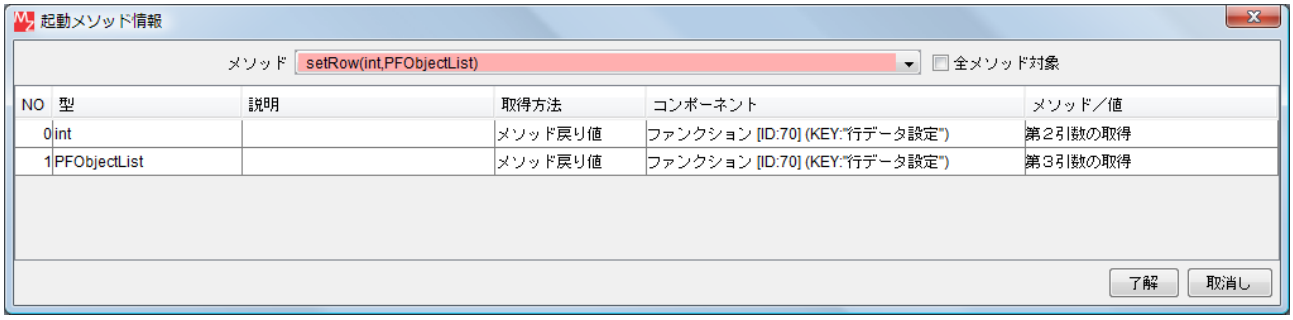
動作イメージのように、「数値入力カウンタ」で指定された左側テーブルデータの行位置に行データを設定し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで行データ設定処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に行位置、第 3 引数に設定データを指定してメソッド「ファンクションの呼び出し（3 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。



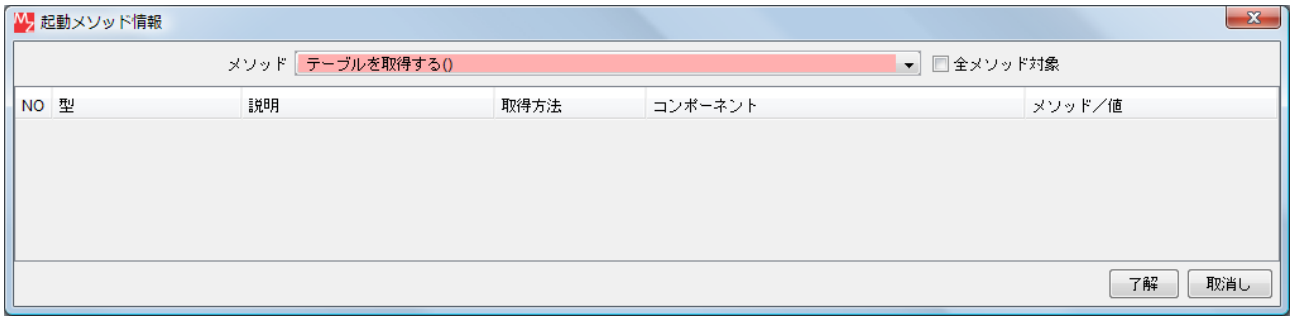
①の接続情報：処理対象のテーブルデータを設定する。



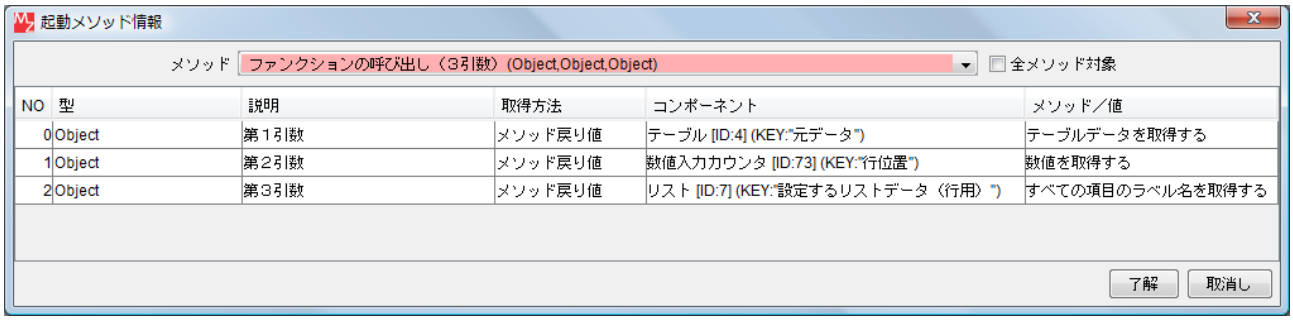
②の接続情報：行位置を指定して行データを設定する。



③の接続情報：ファンクションの戻り値としてテーブルデータを返す。

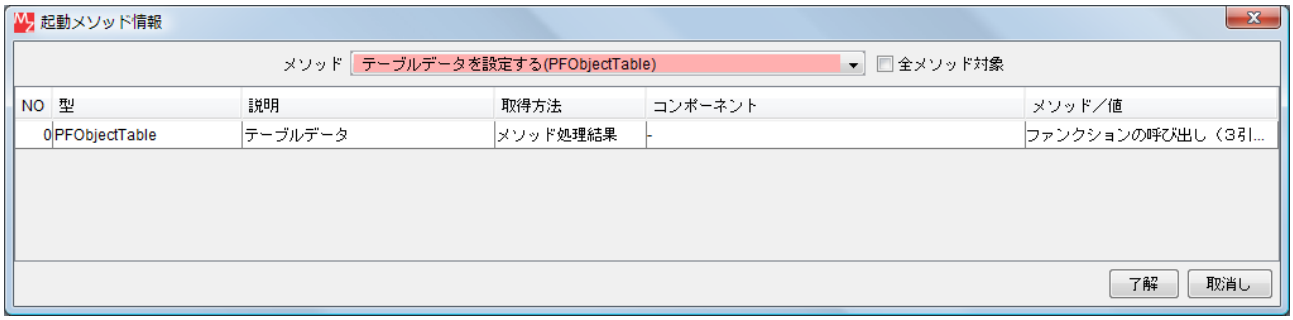


④の接続情報：定義した行データ設定処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	メソッド戻り値	数値入力カウンタ [ID:73] (KEY:"行位置")	数値を取得する
2	Object	第3引数	メソッド戻り値	リスト [ID:7] (KEY:"設定するリストデータ（行用）")	すべての項目のラベル名を取得する

⑤の接続情報：結果をテーブルに設定する。

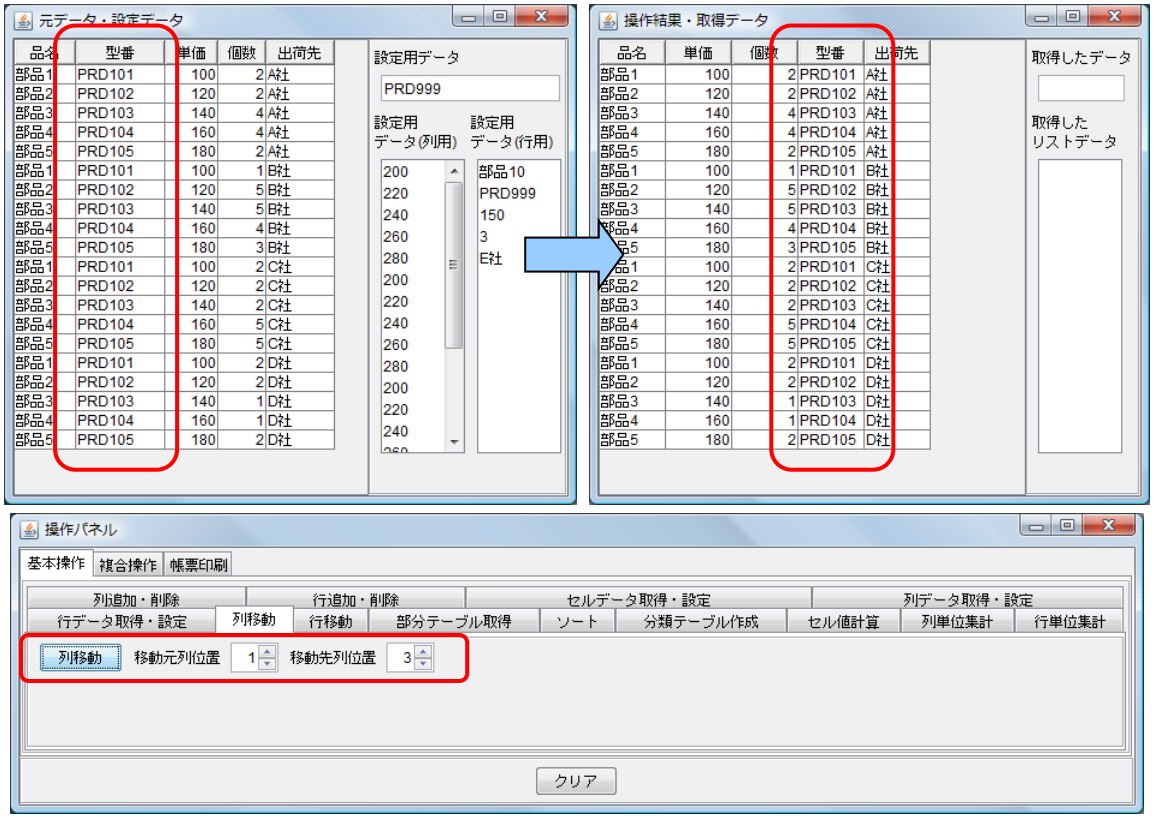


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（3引...

2.11 列の移動

動作イメージ

以下の図のように、テーブルデータの列を移動するための処理手順について説明します。



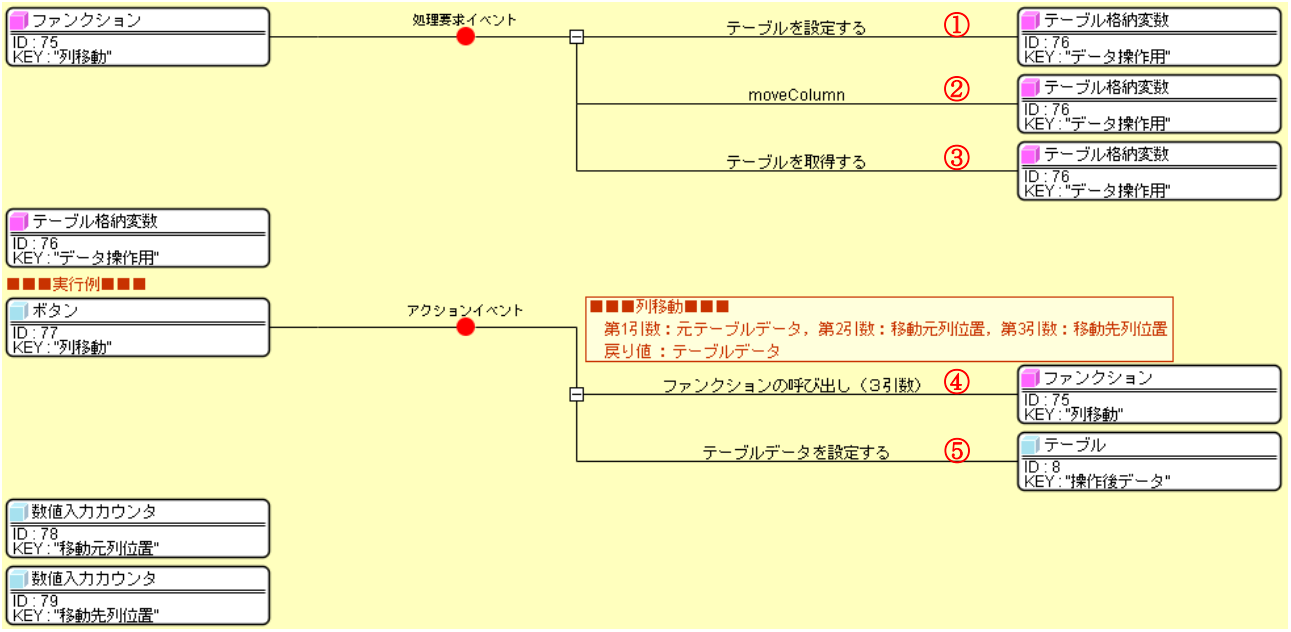
使用メソッド

テーブルデータの列を移動するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

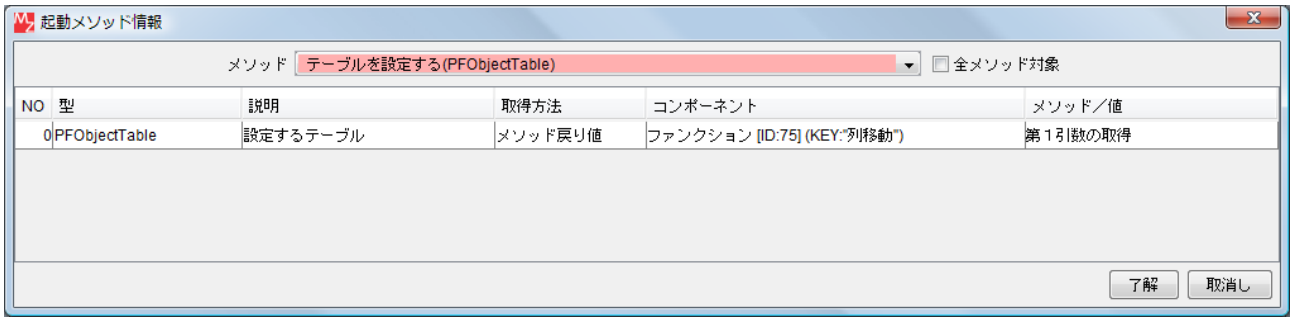
コンポーネント	■テーブル格納変数
メソッド	moveColumn(int, int)
引数	第 1 引数：移動元列位置 第 2 引数：移動先列位置
戻り値	なし
移動元列位置および移動先列位置を指定して列を移動します。	

接続例

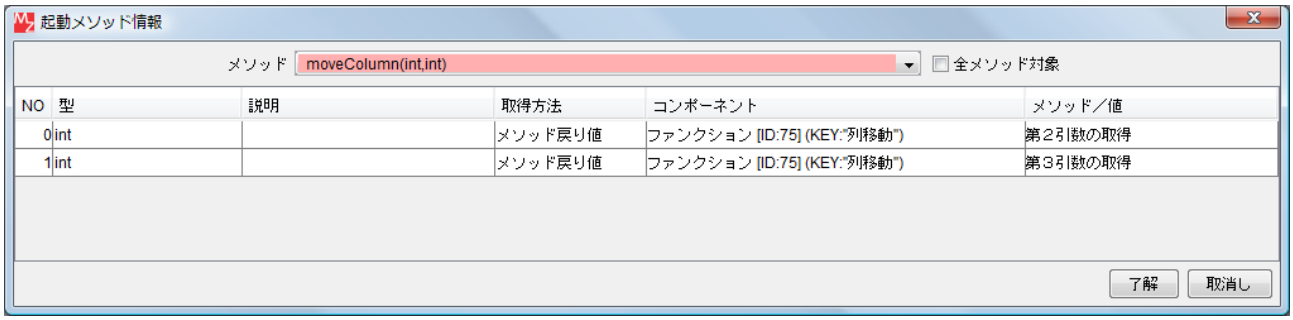
動作イメージのように、「数値入力カウンタ」で指定された列位置に従い左側テーブルデータの列データを移動させ、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで列移動処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に移動元列位置、第 3 引数に移動先列位置を指定してメソッド「ファンクションの呼び出し（3 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。



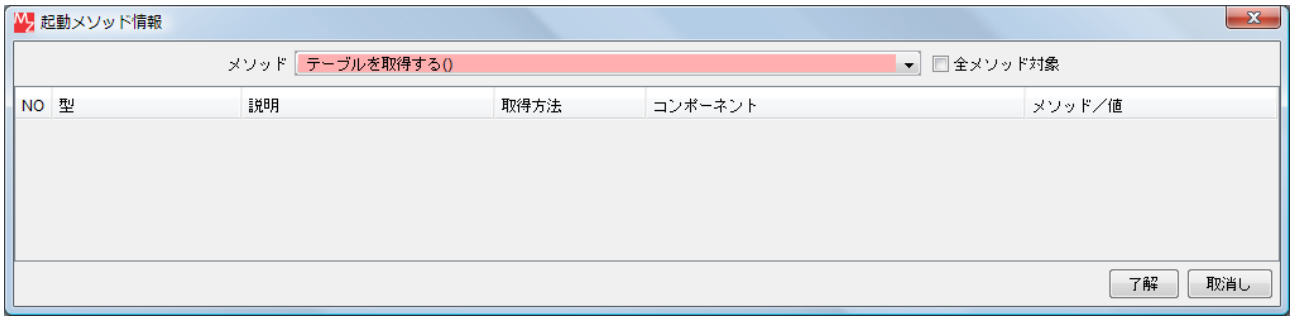
①の接続情報：処理対象のテーブルデータを設定する。



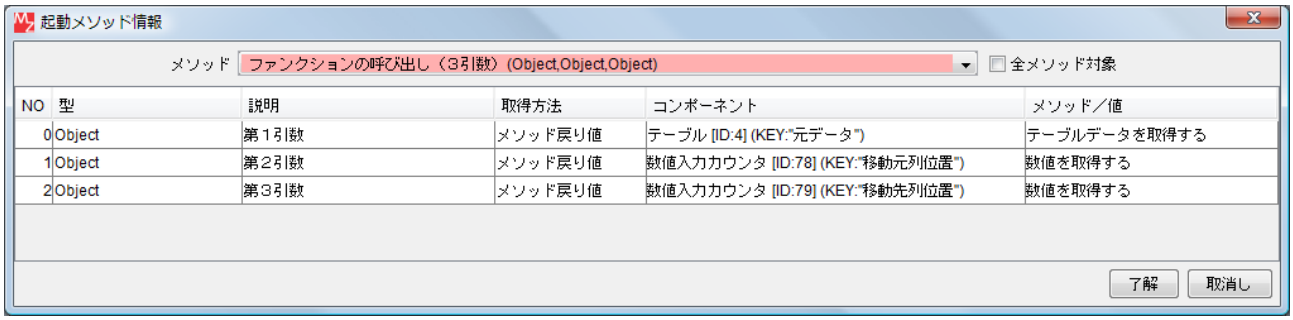
②の接続情報：列位置を指定して列を移動する。



③の接続情報：ファンクションの戻り値としてテーブルデータを返す。

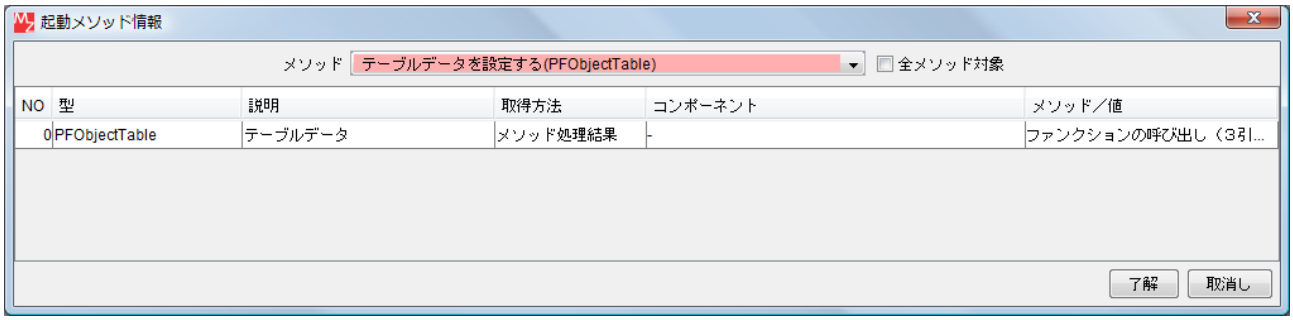


④の接続情報：定義した列移動処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	メソッド戻り値	数値入力カウンタ [ID:78] (KEY:"移動元列位置")	数値を取得する
2	Object	第3引数	メソッド戻り値	数値入力カウンタ [ID:79] (KEY:"移動先列位置")	数値を取得する

⑤の接続情報：結果をテーブルに設定する。

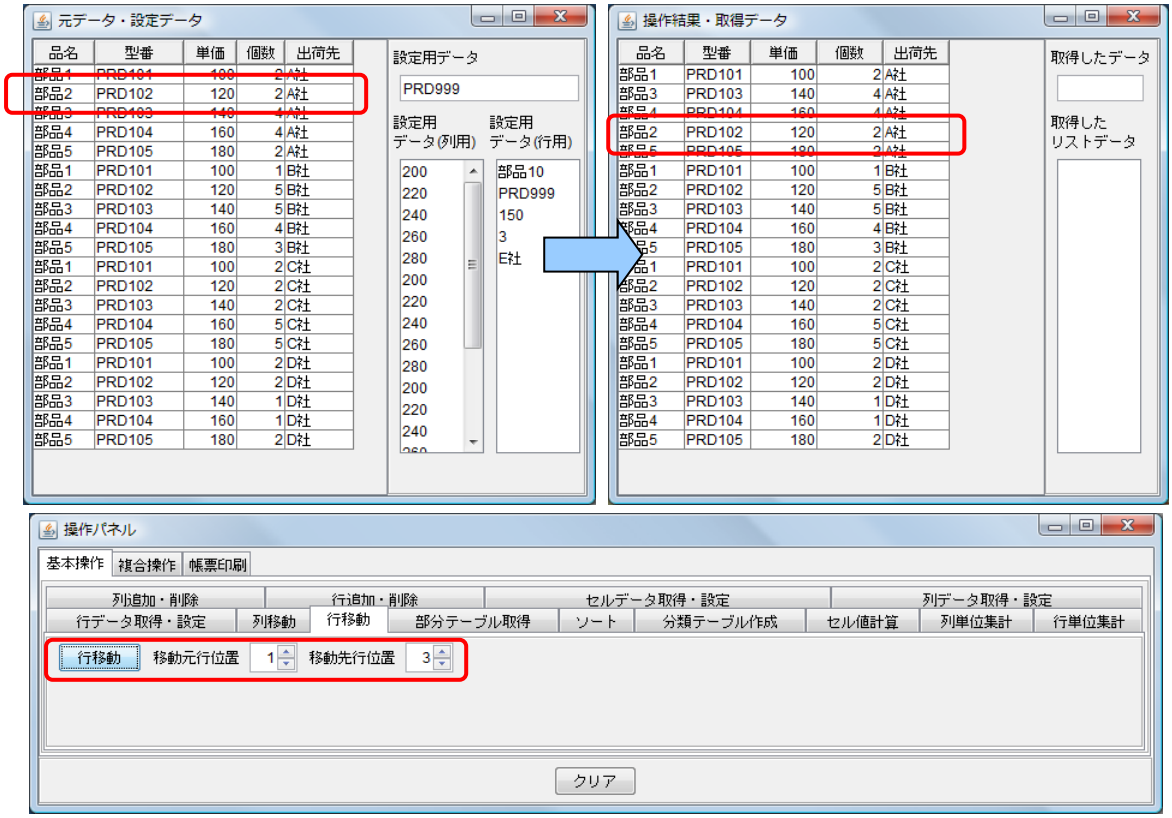


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（3引...

2.12 行の移動

動作イメージ

以下の図のように、テーブルデータの行を移動するための処理手順について説明します。



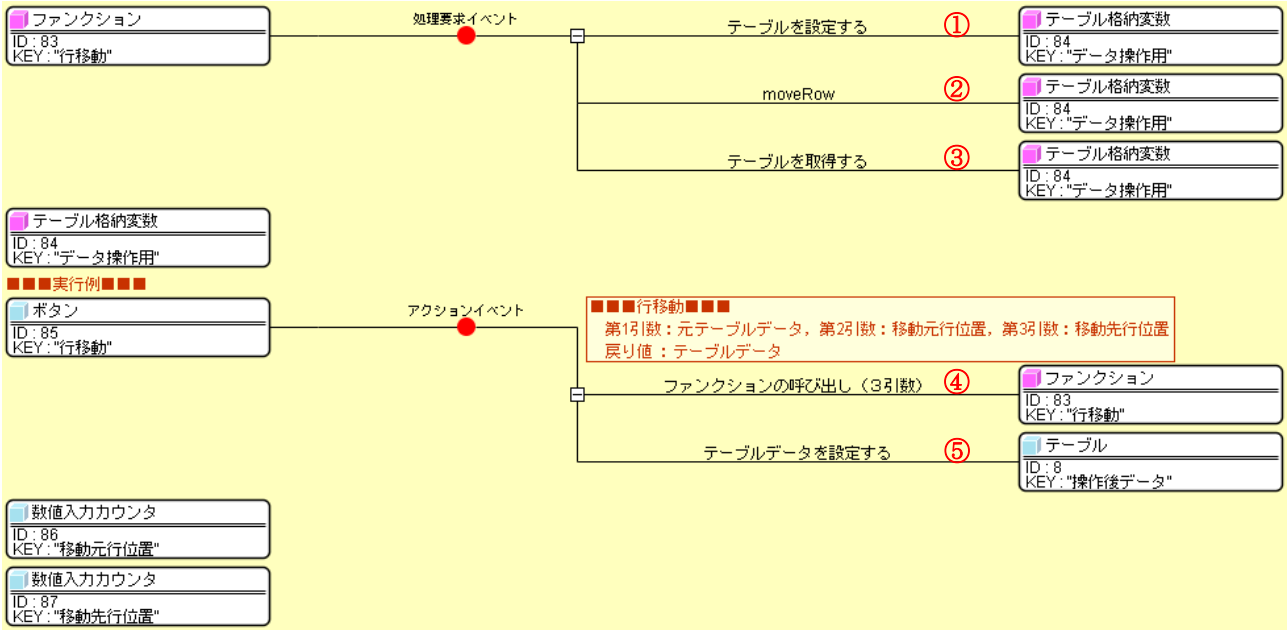
使用メソッド

テーブルデータの行を移動するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

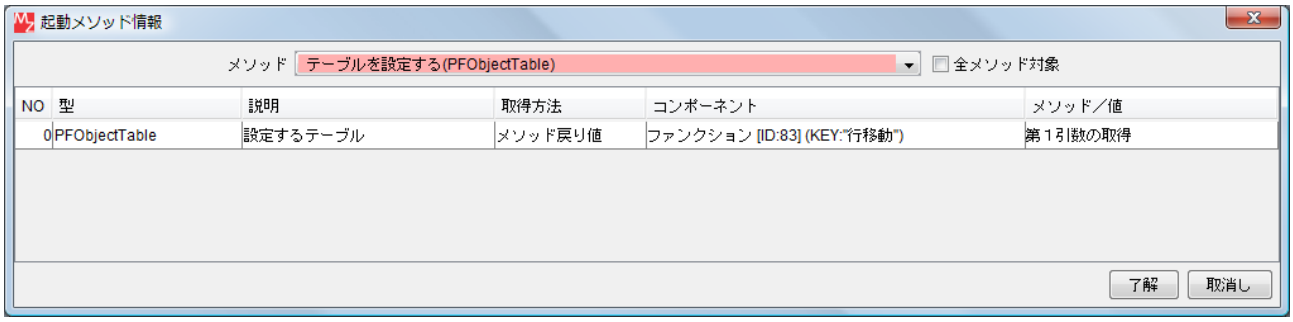
コンポーネント	■テーブル格納変数
メソッド	moveRow(int, int)
引数	第 1 引数：移動元行位置 第 2 引数：移動先行位置
戻り値	なし
移動元行位置および移動先行位置を指定して行を移動します。	

接続例

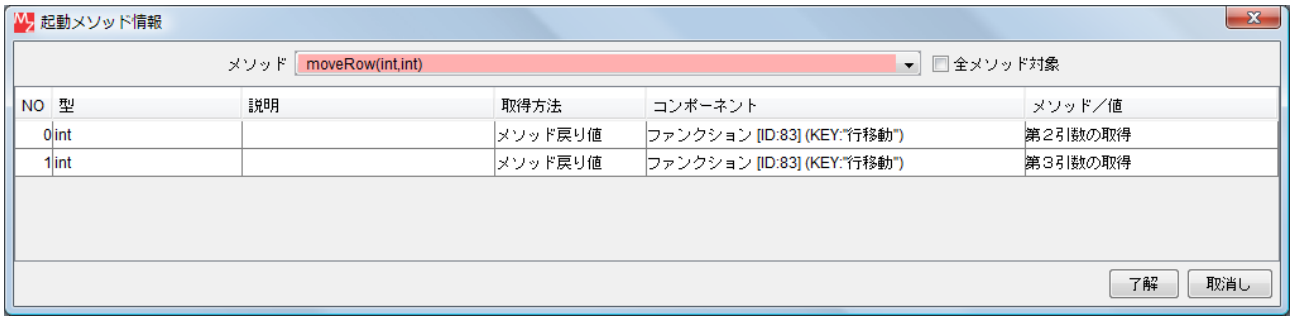
動作イメージのように、「数値入力カウンタ」で指定された行位置に従い左側テーブルデータの行データを移動させ、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで行移動処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に移動元行位置、第 3 引数に移動先行位置を指定してメソッド「ファンクションの呼び出し（3 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。



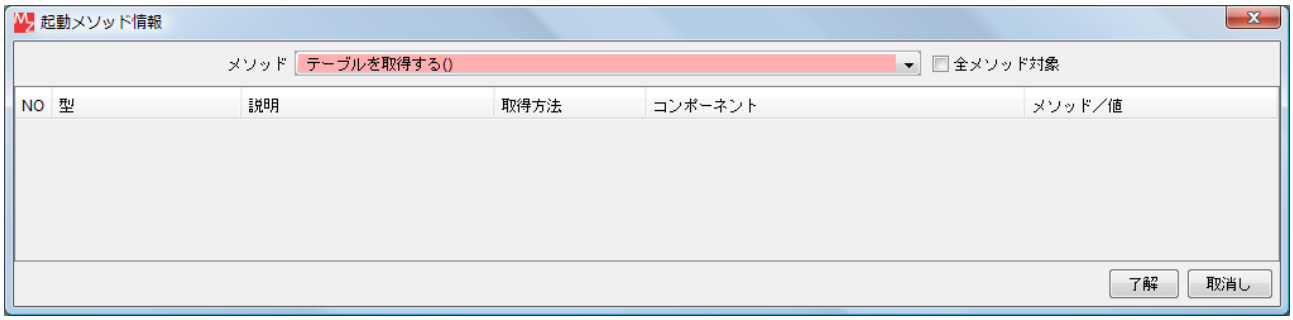
①の接続情報：処理対象のテーブルデータを設定する。



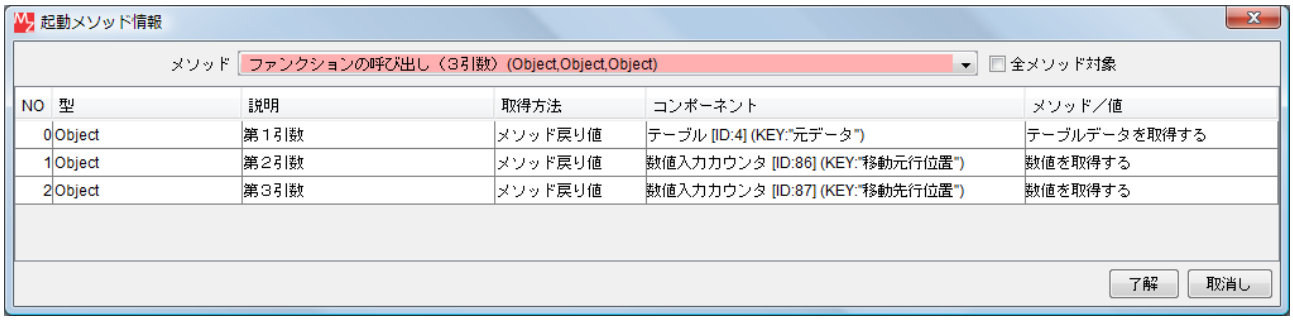
②の接続情報：行位置を指定して行を移動する。



③の接続情報：ファンクションの戻り値としてテーブルデータを返す。

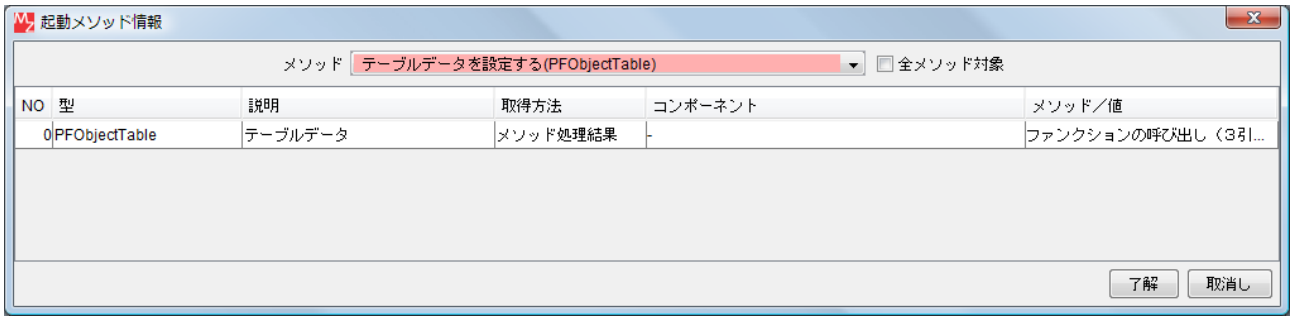


④の接続情報：定義した行移動処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	メソッド戻り値	数値入力カウンタ [ID:86] (KEY:"移動元行位置")	数値を取得する
2	Object	第3引数	メソッド戻り値	数値入力カウンタ [ID:87] (KEY:"移動先行位置")	数値を取得する

⑤の接続情報：結果をテーブルに設定する。

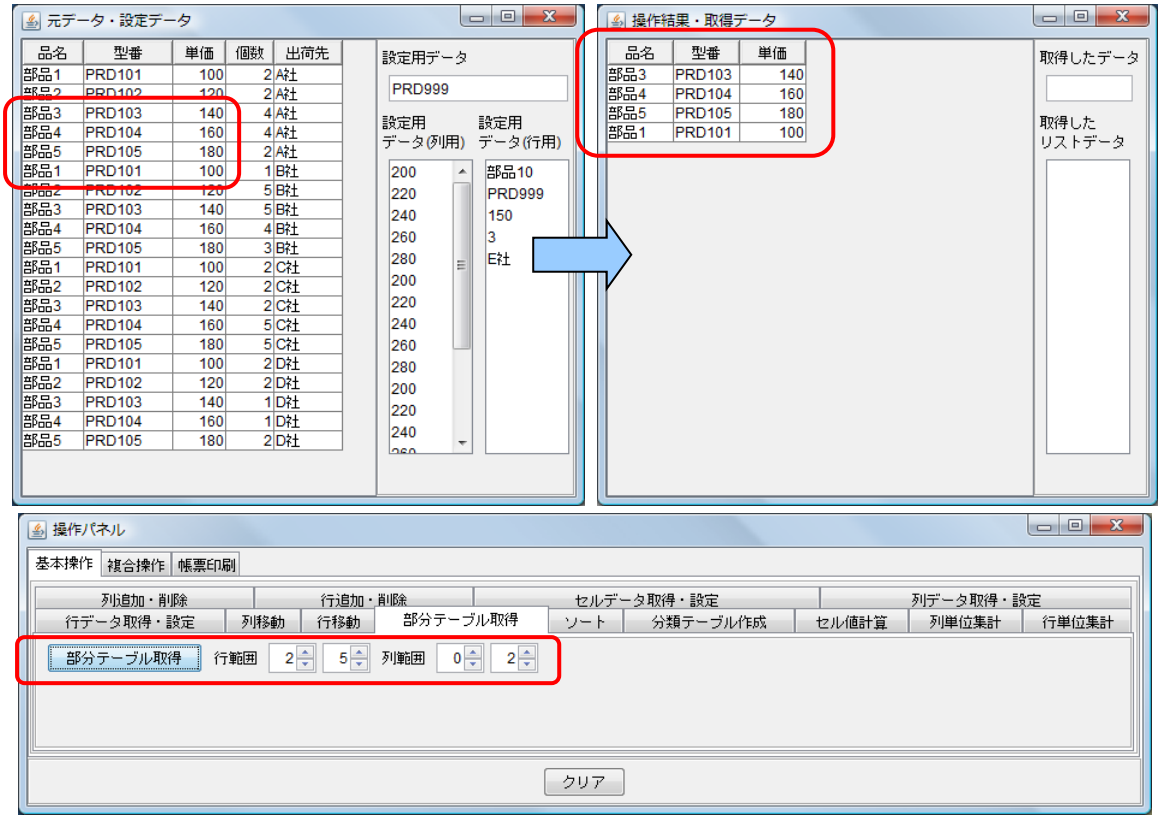


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（3引...

2.13 部分テーブルの取得

動作イメージ

以下の図のように、テーブルデータから部分テーブルデータを取得するための処理手順について説明します。



使用メソッド

テーブルデータから部分テーブルデータを取得するためには、次のコンポーネントのメソッドを順に実行します。

①選択範囲を指定する

コンポーネント	■テーブルサブセットフィルタ
メソッド	選択対象の単一範囲を設定する(int, int, int, int)
引数	第 1 引数：開始行位置 第 2 引数：終了行位置 第 3 引数：開始列位置 第 4 引数：終了列位置
戻り値	なし
行位置および列位置の範囲を指定して選択範囲を設定します。	

②選択範囲のテーブルデータを取得する

コンポーネント	■テーブルサブセットフィルタ
メソッド	選択データを取得する()
引数	なし
戻り値	PFObjectTable
選択範囲のテーブルデータを取得します。	

①に類似する処理を行うものとして、選択範囲の指定の仕方の違いにより次のメソッドがあります。

メソッド	選択対象列を設定する(int)
引数	第１引数：列位置
戻り値	なし
列位置を指定して選択範囲を設定します。	

メソッド	選択対象列を設定する(int, int)
引数	第１引数：開始列位置 第２引数：終了列位置
戻り値	なし
列範囲を指定して選択範囲を設定します。	

メソッド	選択対象列を設定する(PFObjectList)
引数	第１引数：列位置のリスト
戻り値	なし
列位置のリスト（整数を表現する文字列のリスト、整数のリスト、連続していなくても可能）を指定して選択範囲を設定します。	

メソッド	全列を選択する()
引数	なし
戻り値	なし
全列を選択範囲に設定します。	

メソッド	選択対象行を設定する(int)
引数	第１引数：行位置
戻り値	なし
行位置を指定して選択範囲を設定します。	

メソッド	選択対象行を設定する(int, int)
引数	第１引数：開始行位置 第２引数：終了行位置
戻り値	なし
行範囲を指定して選択範囲を設定します。	

メソッド	選択対象行を設定する(PFObjectList)
引数	第１引数：行位置のリスト
戻り値	なし
行位置のリスト（整数を表現する文字列のリスト、整数のリスト、連続していなくても可能）を指定して選択範囲を設定します。	

メソッド	全行を選択する()
引数	なし
戻り値	なし
全行を選択範囲に設定します。	

メソッド	列データの日時で行を選択する(int, Date, boolean, Date, boolean)
引数	第１引数：列位置 第２引数：開始日時 第３引数：開始日時を含むかどうか 第４引数：終了日時 第５引数：終了日時を含むかどうか
戻り値	なし
日付列型の位置および日時の範囲を指定して選択範囲を設定します。	

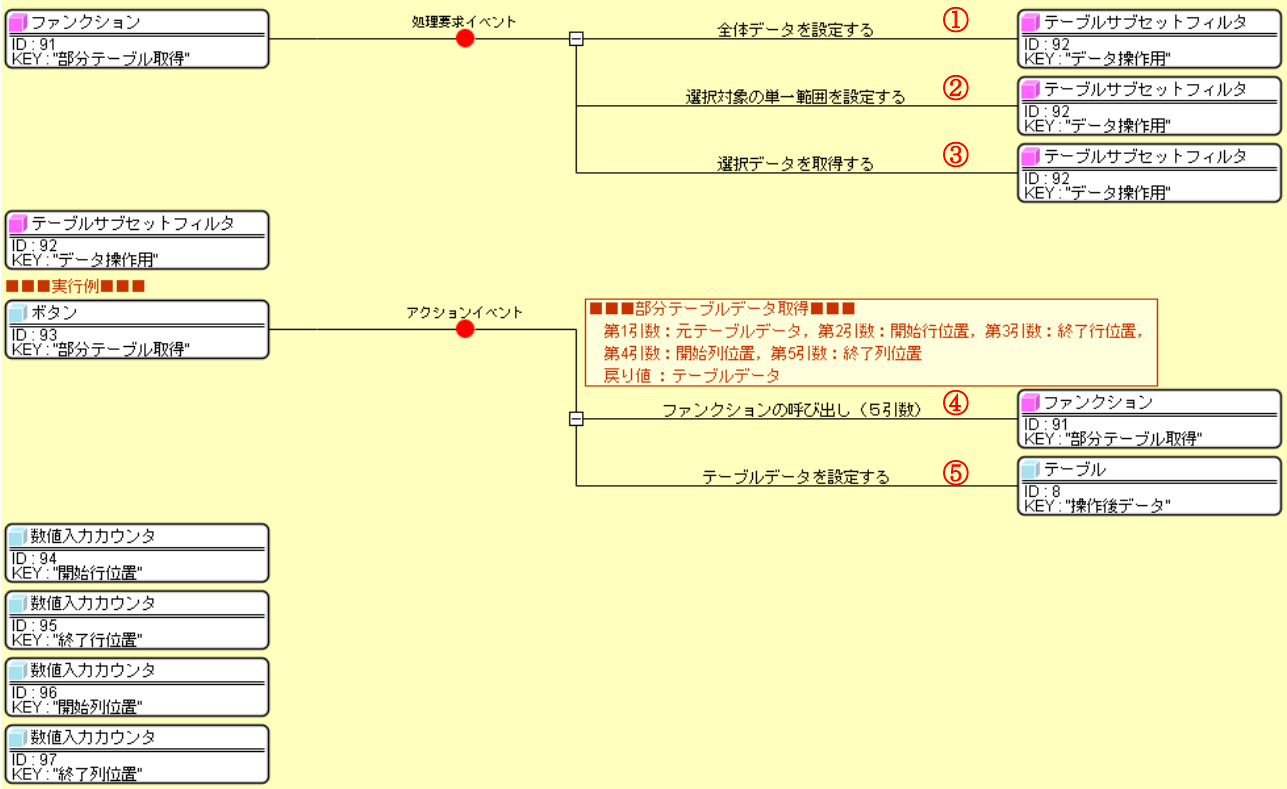
メソッド	列データの数値で行を選択する(int, Number, boolean, Number, boolean)
引数	第１引数：列位置 第２引数：下限値 第３引数：下限値を含むかどうか 第４引数：上限値 第５引数：上限値を含むかどうか
戻り値	なし
数値列型の位置および数値の範囲を指定して選択範囲を設定します。	

メソッド	列データの正規表現で行を選択する(int, String)
引数	第１引数：列位置 第２引数：正規表現文字列
戻り値	なし
文字列型の列の位置および正規表現を指定して、正規表現に合致する文字列を含む行を選択範囲に設定します。	

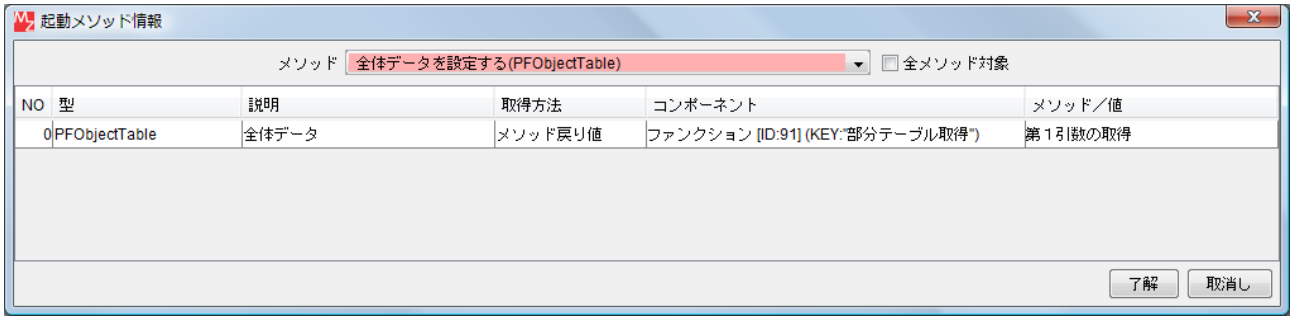
メソッド	列データの文字列で行を選択する(int, String)
引数	第１引数：列位置 第２引数：文字列
戻り値	なし
文字列型の列の位置および文字列を指定して、一致する文字列を含む行を選択範囲に設定します。	

接続例

動作イメージのように、「数値入力カウンタ」で指定された選択範囲に対して左側テーブルデータから部分テーブルデータを取得し「テーブル」に設定する処理手順を以下のように定義します。ここで部分テーブルデータ取得処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に開始行位置、第 3 引数に終了行位置、第 4 引数に開始列位置、第 5 引数に終了列位置を指定してメソッド「ファンクションの呼び出し（5 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。



①の接続情報：処理対象のテーブルデータを設定する。



②の接続情報：選択範囲を指定する。

起動メソッド情報

メソッド 選択対象の単一範囲を設定する(int,int,int,int) 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	int	開始行インデックス	メソッド戻り値	ファンクション [ID:91] (KEY:"部分テーブル取得")	第2引数の取得
1	int	終了行インデックス	メソッド戻り値	ファンクション [ID:91] (KEY:"部分テーブル取得")	第3引数の取得
2	int	開始列インデックス	メソッド戻り値	ファンクション [ID:91] (KEY:"部分テーブル取得")	第4引数の取得
3	int	終了列インデックス	メソッド戻り値	ファンクション [ID:91] (KEY:"部分テーブル取得")	第5引数の取得

了解 取消し

③の接続情報：選択範囲のデータを取得し、ファンクションの戻り値として返す。

起動メソッド情報

メソッド 選択データを取得する() 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

了解 取消し

④の接続情報：定義した部分テーブルデータ取得処理を実行する。

起動メソッド情報

メソッド ファンクションの呼び出し（5引数）(Object,Object,Object,Object,Object) 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	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	メソッド戻り値	数値入力カウンタ [ID:94] (KEY:"開始行位置")	数値を取得する
2	Object	第3引数	メソッド戻り値	数値入力カウンタ [ID:95] (KEY:"終了行位置")	数値を取得する
3	Object	第4引数	メソッド戻り値	数値入力カウンタ [ID:96] (KEY:"開始列位置")	数値を取得する
4	Object	第5引数	メソッド戻り値	数値入力カウンタ [ID:97] (KEY:"終了列位置")	数値を取得する

了解 取消し

⑤の接続情報：結果をテーブルに設定する。

起動メソッド情報

メソッド テーブルデータを設定する(PFObjectTable) 全メソッド対象

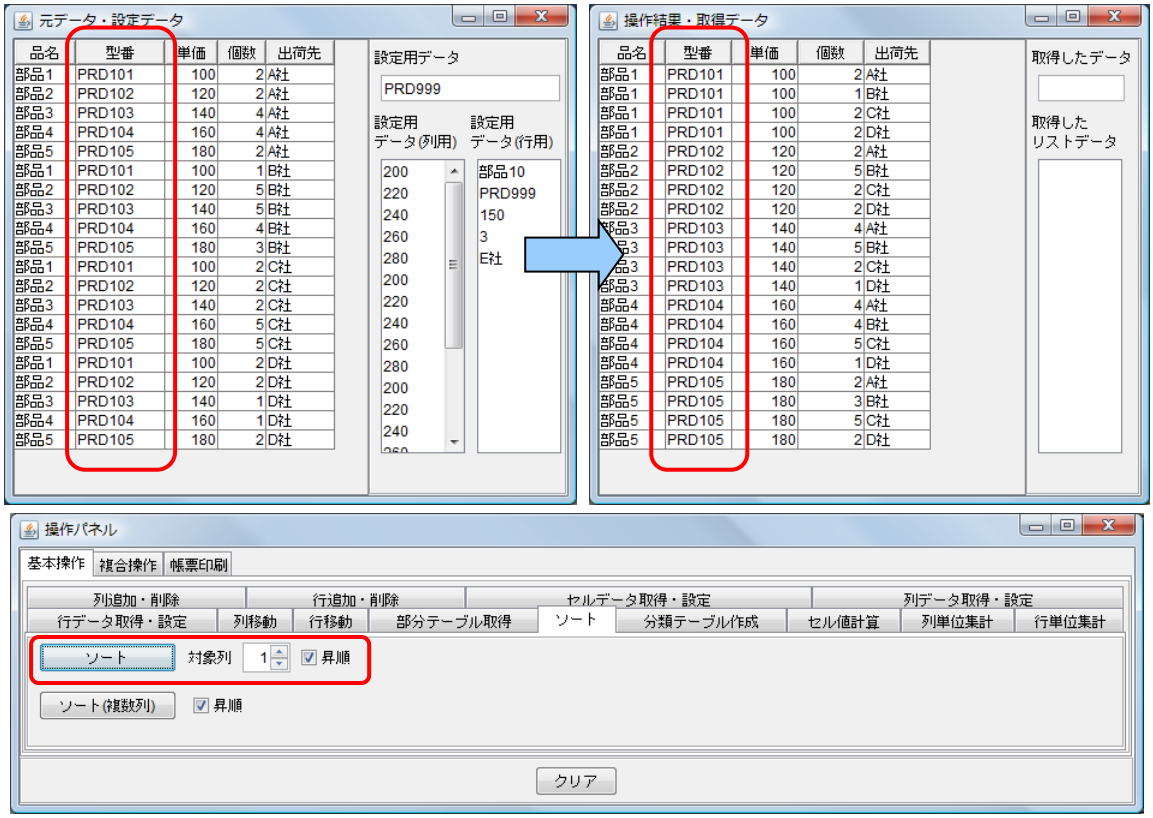
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（5引...

了解 取消し

2.14 並べ替え（ソート、一列）

動作イメージ

以下の図のように、テーブルデータを指定列についてソートする処理手順について説明します。



使用メソッド

テーブルデータを指定列についてソートするために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

コンポーネント	■テーブル格納変数
メソッド	列を指定して行をソートする(int, boolean)
引数	第 1 引数：ソート列位置 第 2 引数：整列順序（true：昇順, flase：降順）
戻り値	なし
ソート列位置および整列順序を指定して、文字列として行を並べ替えます。	

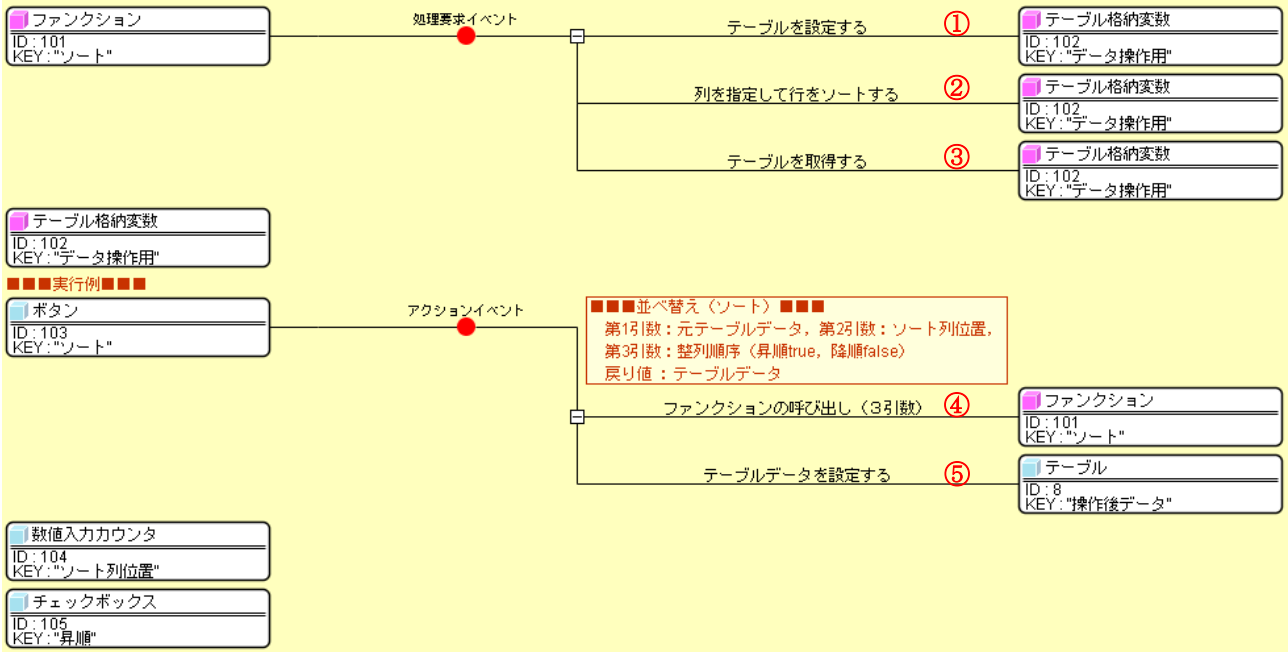
類似する処理行うメソッドとして次のメソッドがあります。

メソッド	列を指定して日時で行をソートする(int, boolean)
引数	第 1 引数：ソート列位置 第 2 引数：整列順序（true：昇順, flase：降順）
戻り値	なし
ソート列位置および整列順序を指定して、日時として行を並べ替えます。	

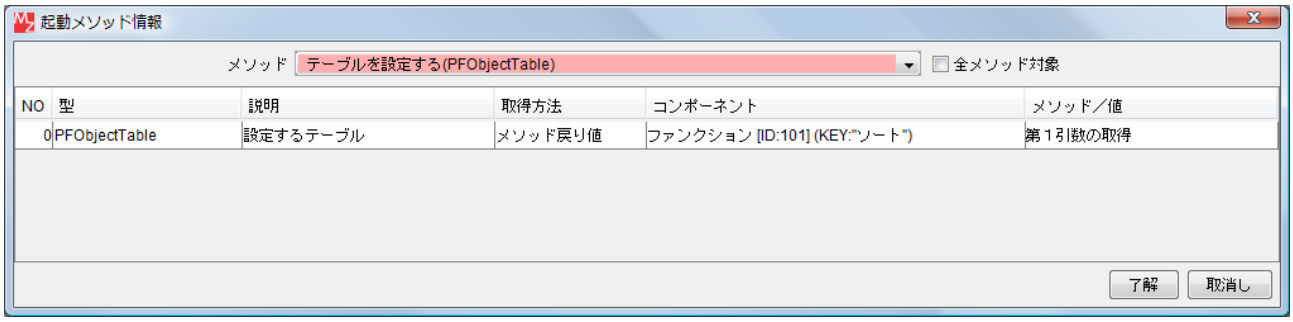
メソッド	列を指定して数値で行をソートする(int, boolean)
引数	第 1 引数：ソート列位置 第 2 引数：整列順序（true：昇順, flase：降順）
戻り値	なし
ソート列位置および整列順序を指定して、数値として行を並べ替えます。	

接続例

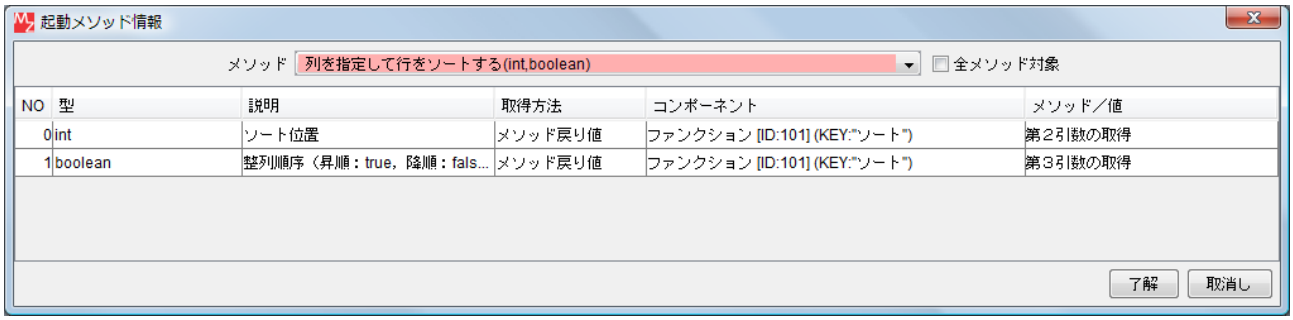
動作イメージのように、「数値入力カウンタ」で指定された列位置および「チェックボックス」で指定された整列順序に従い左側テーブルデータをソートし、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここでソート処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数にソート列位置、第 3 引数に整列順序を指定してメソッド「ファンクションの呼び出し（3 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。



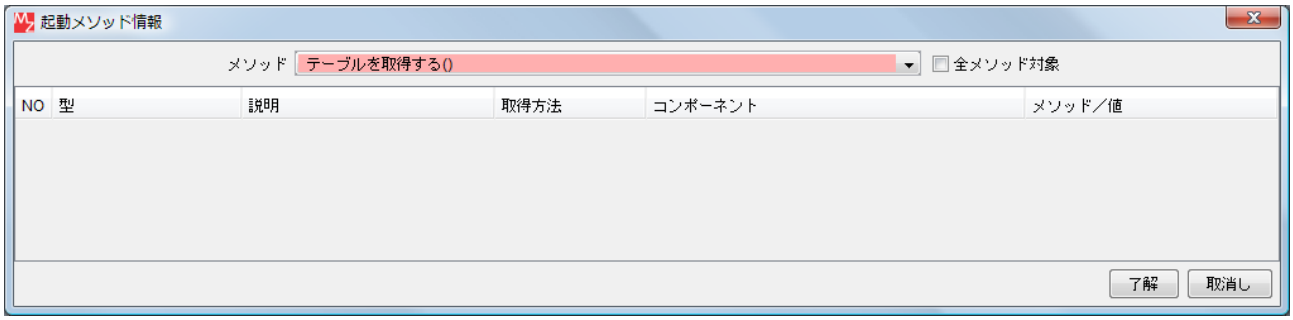
①の接続情報：処理対象のテーブルデータを設定する。



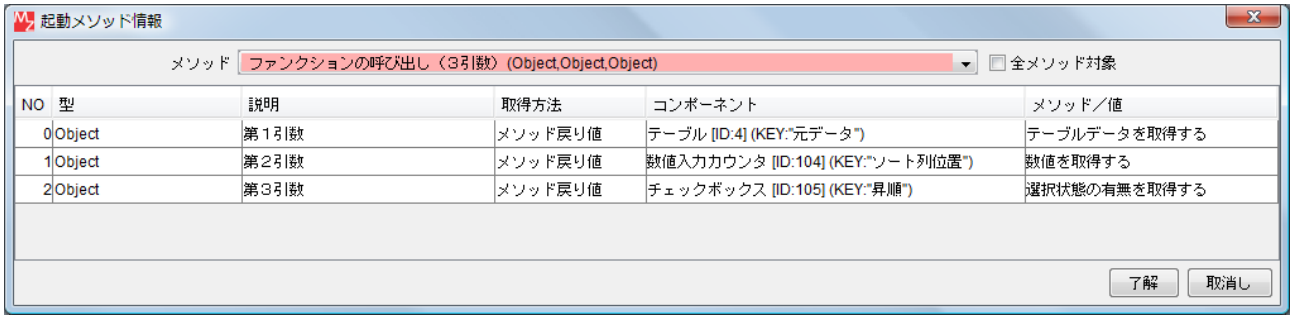
②の接続情報：ソート列位置および整列順序を指定してソートする。



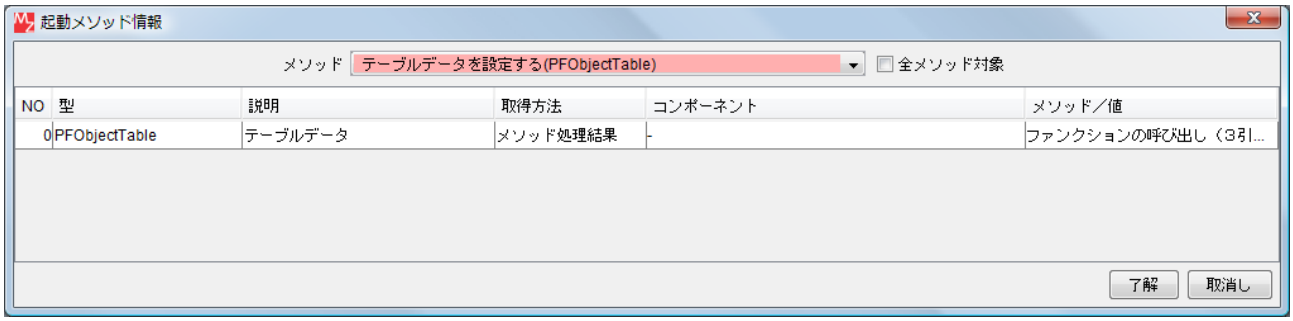
③の接続情報：ファンクションの戻り値としてテーブルデータを返す。



④の接続情報：定義したソート処理を実行する。



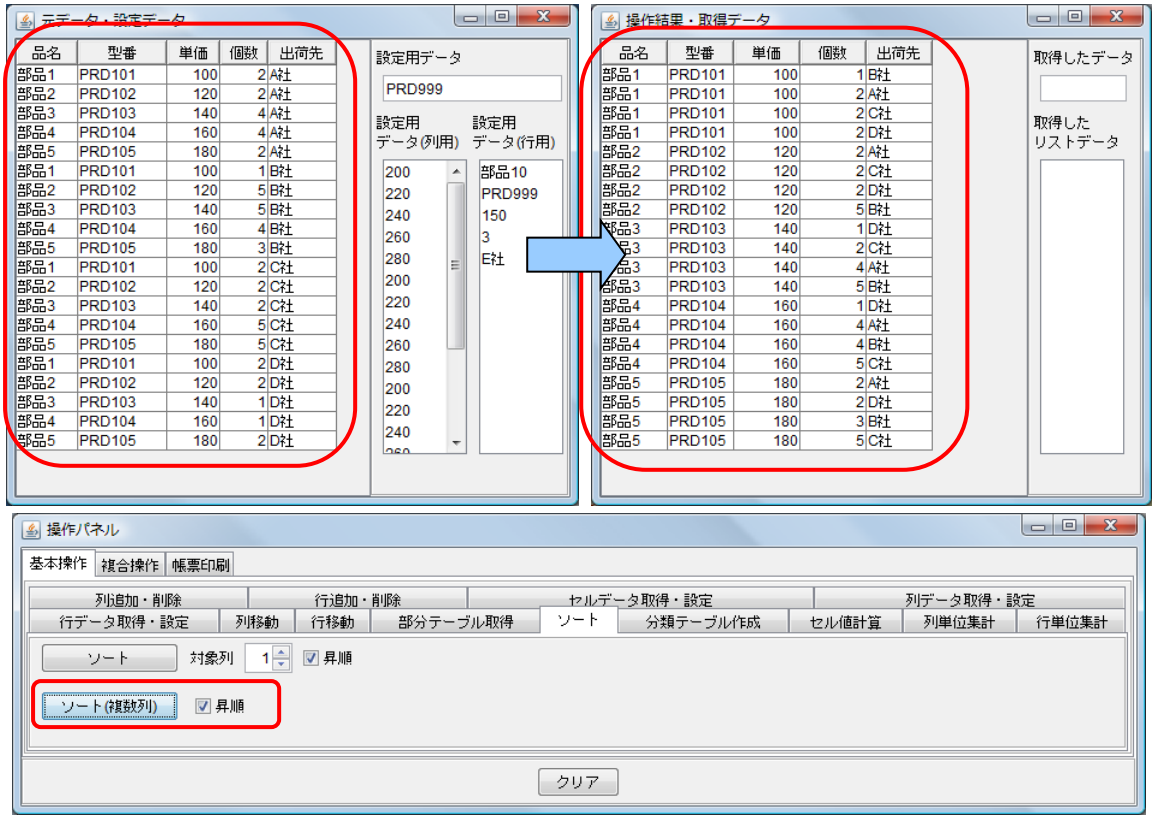
⑤の接続情報：結果をテーブルに設定する。



2.15 並べ替え（ソート、複数列）

動作イメージ

以下の図のように、テーブルデータの複数列を対象として左側の列ほど優先的にソートする処理手順について説明します。また、列のデータ型に応じてソートに用いるメソッドを切り替えます。



使用メソッド

テーブルデータの複数列を対象として左側の列ほど優先的にソートするためには、テーブルの右側から順にソートを実行する必要があります。また、列のデータ型によるソートメソッドの切り替えは、データ型による条件分岐を定義することにより実現できます。このような処理のために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

①テーブルデータの列について繰り返し処理を実行する

コンポーネント	■繰り返し処理(FOR)
メソッド	繰り返し処理を実行する(int, boolean, int, boolean, int)
引数	第１引数：開始値 第２引数：開始値を含めるかどうか 第３引数：終了値 第４引数：終了値を含めるかどうか 第５引数：増減値
戻り値	なし
テーブルデータの列について繰り返し処理を実行します。	

②ソートを実行する：数値型、日付型以外の場合

コンポーネント	■テーブル格納変数
メソッド	列を指定して行をソートする(int, boolean)
引数	第 1 引数：ソート列位置 第 2 引数：整列順序（true：昇順, flase：降順）
戻り値	なし
ソート列位置および整列順序を指定して、文字列として行を並べ替えます。	

日付型の場合

メソッド	列を指定して日時で行をソートする(int, boolean)
引数	第 1 引数：ソート列位置 第 2 引数：整列順序（true：昇順, flase：降順）
戻り値	なし
ソート列位置および整列順序を指定して、日時として行を並べ替えます。	

数値型（本章では整数(BigInteger)、実数(BigDecimal)）の場合

メソッド	列を指定して数値で行をソートする(int, boolean)
引数	第 1 引数：ソート列位置 第 2 引数：整列順序（true：昇順, flase：降順）
戻り値	なし
ソート列位置および整列順序を指定して、数値として行を並べ替えます。	

②列型の判定

コンポーネント	■テーブル格納変数
メソッド	列型を位置指定で取得する(int)
引数	第 1 引数：列位置
戻り値	Class
指定列のデータ型を取得します。	

コンポーネント	■ユーティリティ起動
メソッド	クラスからクラス名取得(Class)
引数	第 1 引数：クラス
戻り値	String
データ型からデータ型名（文字列）を取得します。	

コンポーネント	■オブジェクト分岐
メソッド	オブジェクトの一致判定を実行する(Object)
引数	第 1 引数：オブジェクト
戻り値	なし
オブジェクト（文字列も可）の一致判定により条件分岐を行います。	

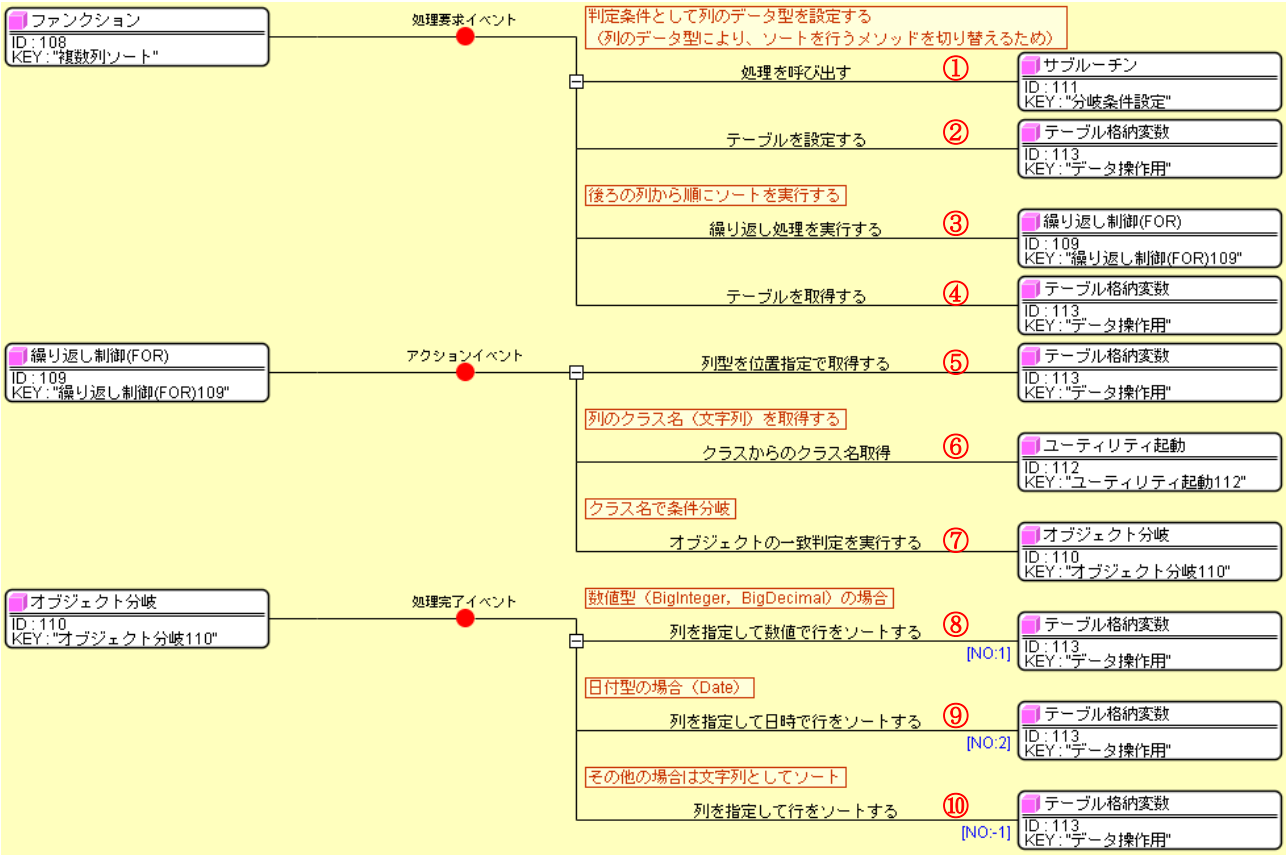
③分岐条件の定義

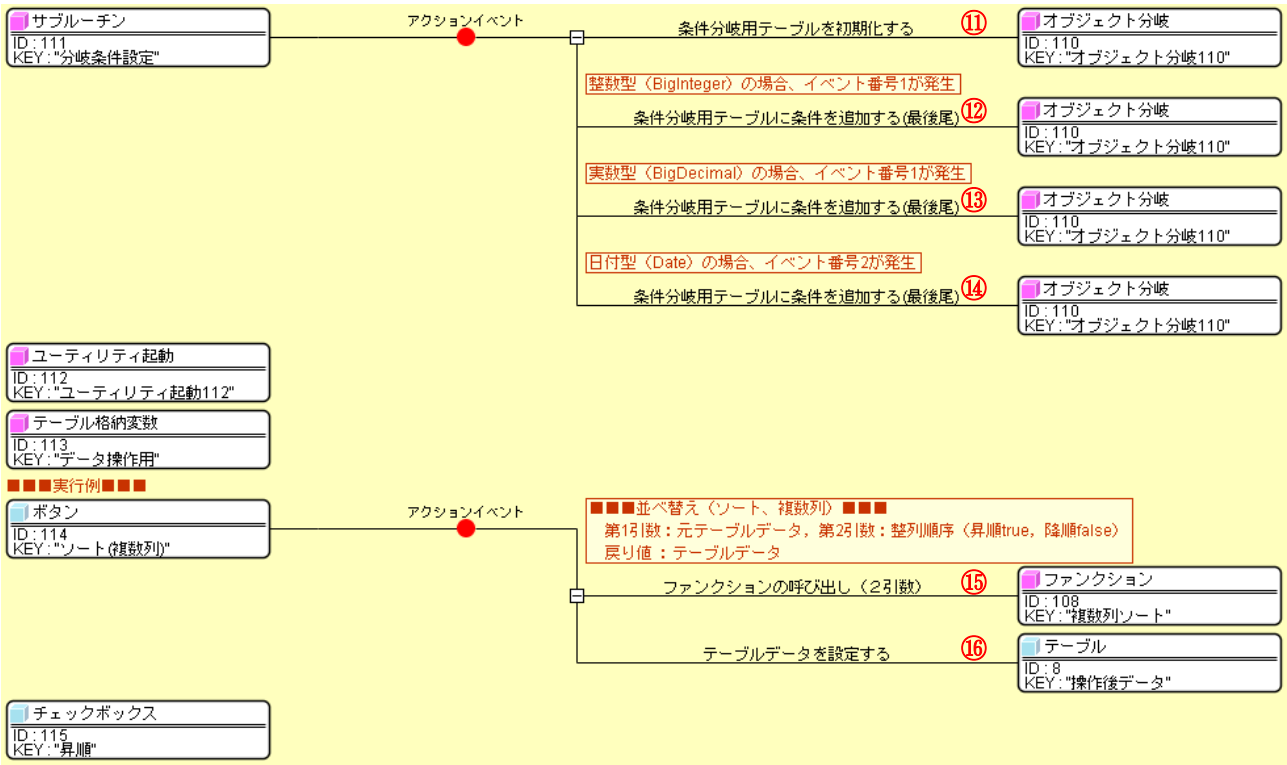
コンポーネント	■オブジェクト分岐
メソッド	条件分岐用テーブルを初期化する()
引数	なし
戻り値	なし
分岐条件を初期化します。	

メソッド	条件分岐用テーブルに条件を追加する(int, Object)
引数	第 1 引数:条件が一致した際に発生する処理完了イベントのイベント番号 第 2 引数：一致判定に用いるオブジェクト（文字列も可）
戻り値	なし
分岐条件を追加します。	

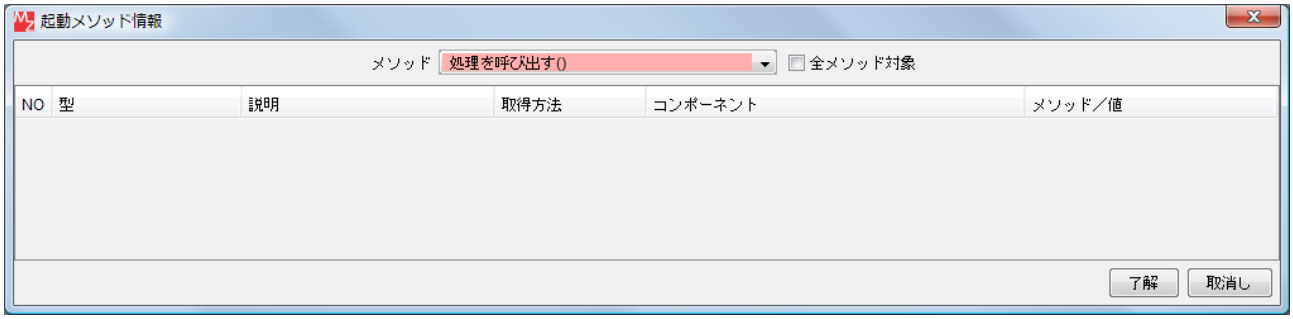
接続例

動作イメージのように､｢チェックボックス｣で指定された整列順序に従い左側テーブルデータをソートし、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここでソート処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に整列順序を指定してメソッド「ファンクションの呼び出し（2 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。

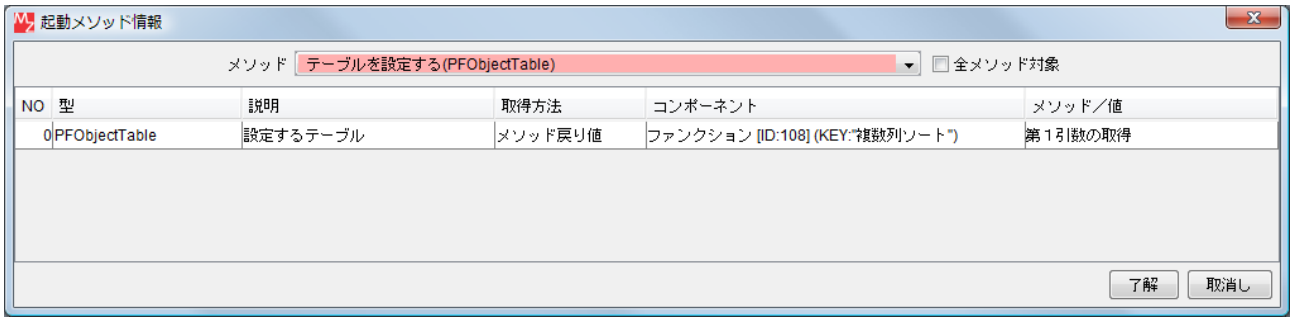




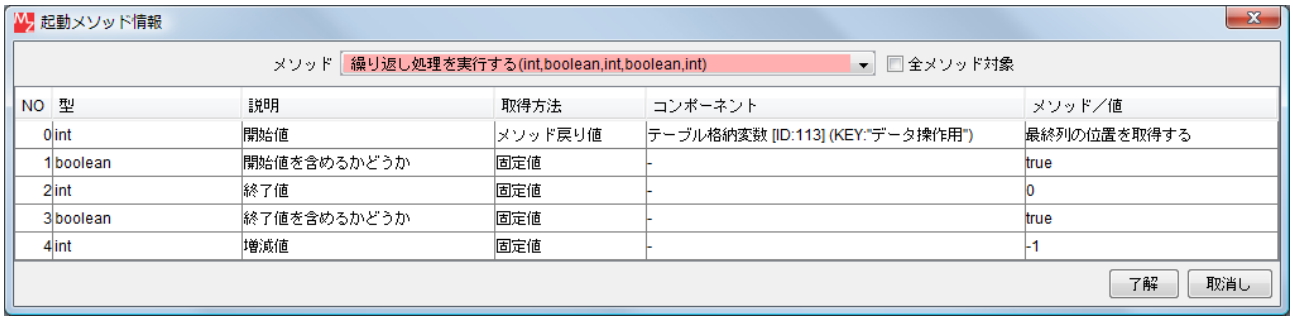
①の接続情報：分岐条件を設定する。



②の接続情報：処理対象のテーブルデータを設定する。

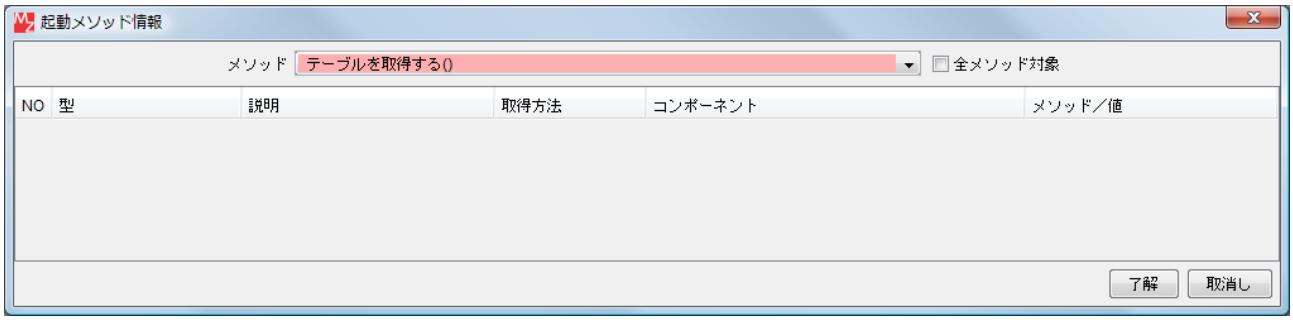


③の接続情報：列について繰り返し処理を実行する。



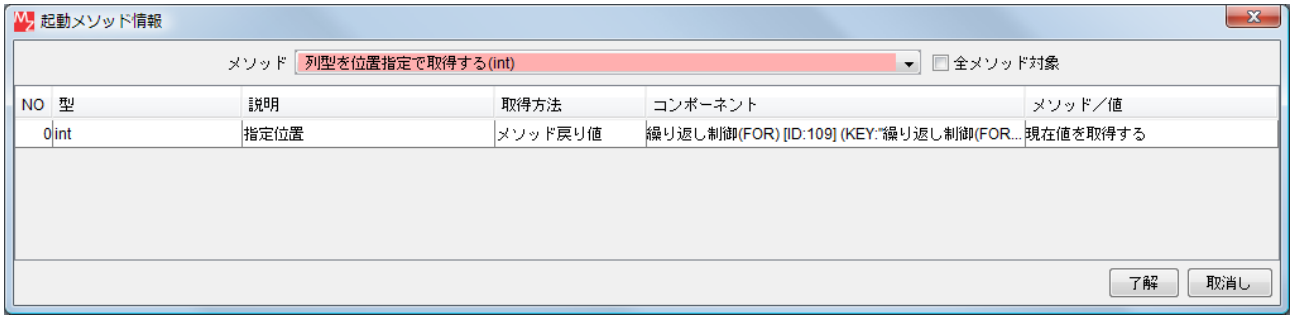
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	開始値	メソッド戻り値	テーブル格納変数 [ID:113] (KEY:"データ操作用")	最終列の位置を取得する
1	boolean	開始値を含めるかどうか	固定値	-	true
2	int	終了値	固定値	-	0
3	boolean	終了値を含めるかどうか	固定値	-	true
4	int	増減値	固定値	-	-1

④の接続情報：ファンクションの戻り値としてテーブルデータを返す。



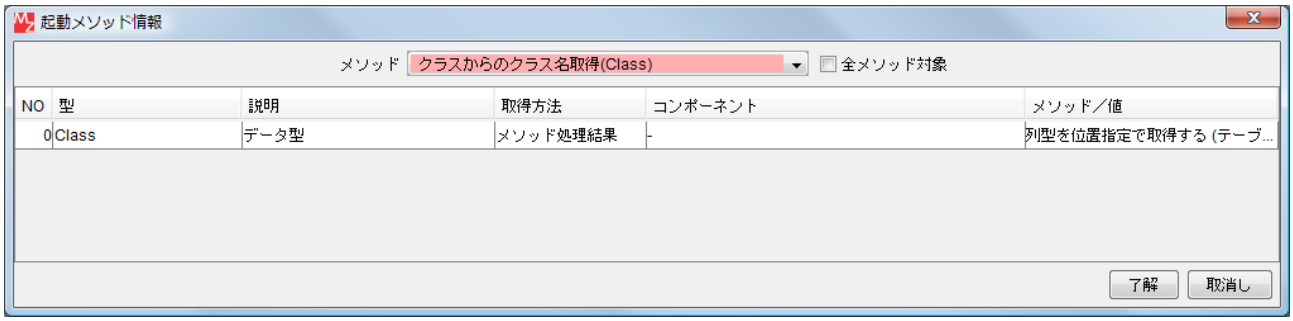
NO	型	説明	取得方法	コンポーネント	メソッド／値

⑤の接続情報：列のデータ型を取得する。



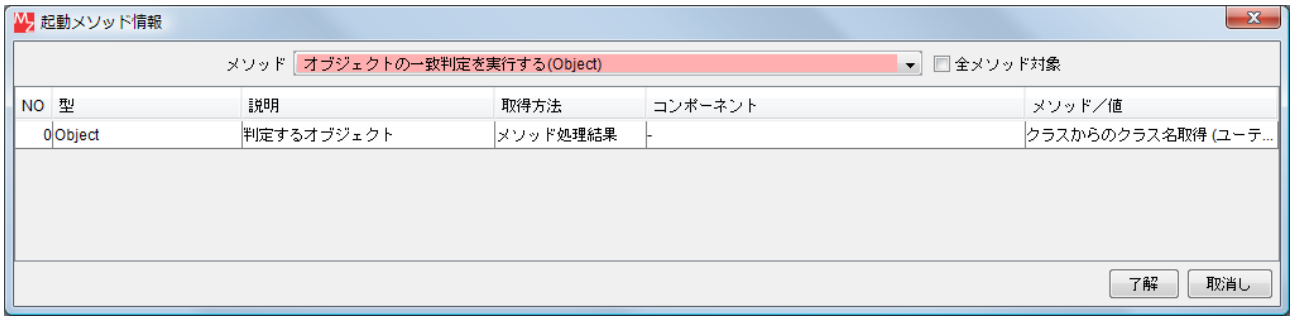
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	指定位置	メソッド戻り値	繰り返し制御(FOR) [ID:109] (KEY:"繰り返し制御(FOR...	現在値を取得する

⑥の接続情報：クラス（データ型）からクラス名の文字列に変換する。

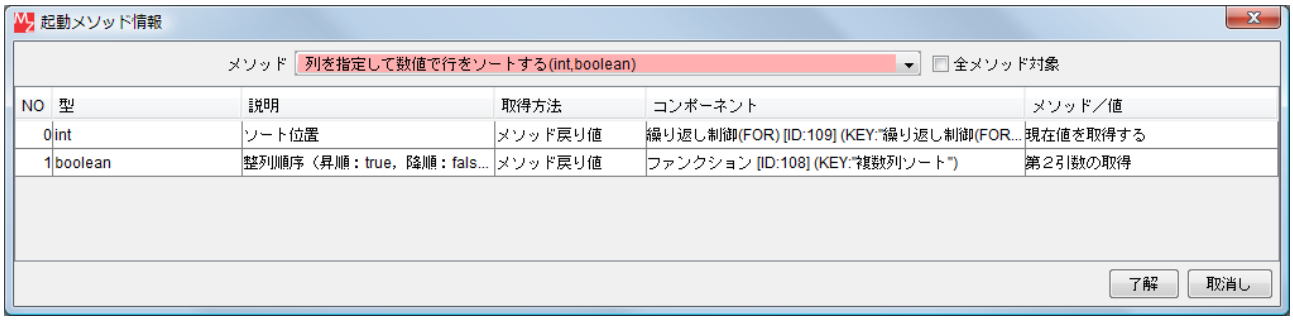


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Class	データ型	メソッド処理結果	-	列型を位置指定で取得する (テーブ...

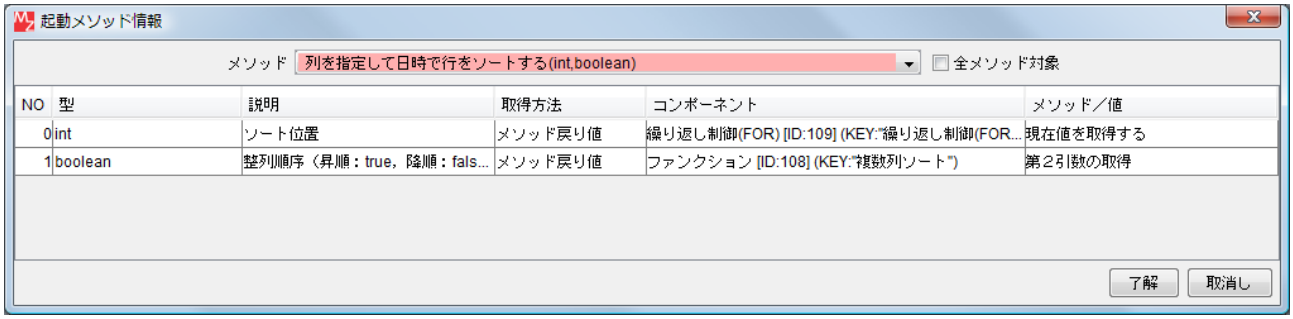
⑦の接続情報：クラス名で条件分岐を行う。



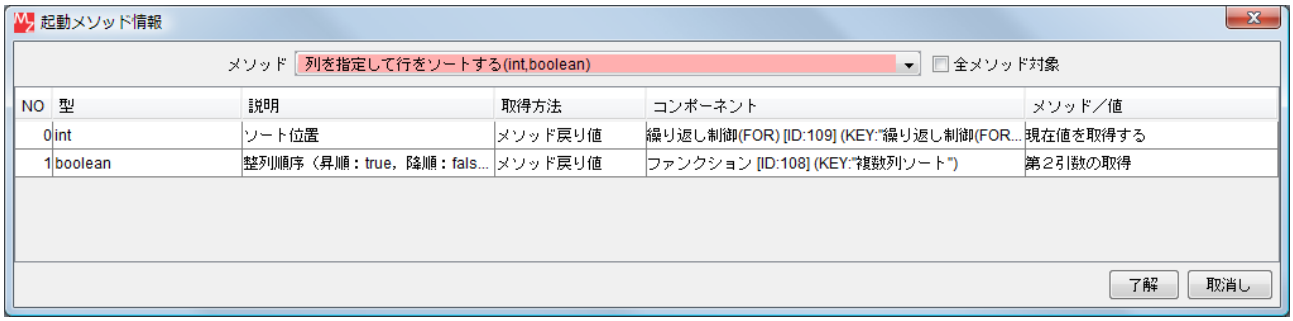
⑧の接続情報：数値型の場合、数値型としてソートを実行する。



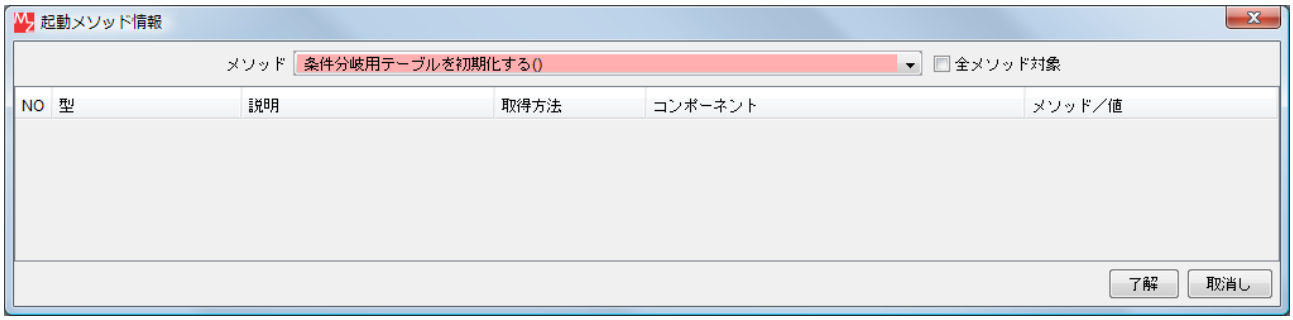
⑨の接続情報：日付型の場合、日付型としてソートを実行する。



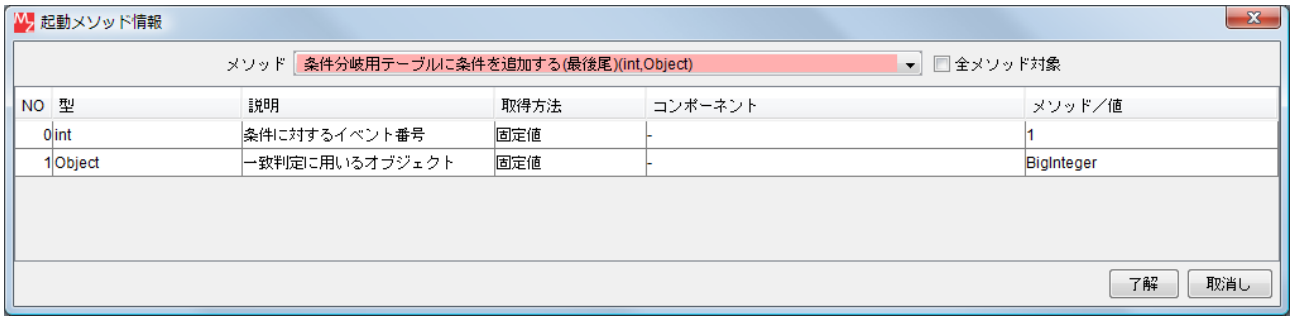
⑩の接続情報：その他の場合は文字列型としてソートを実行する。



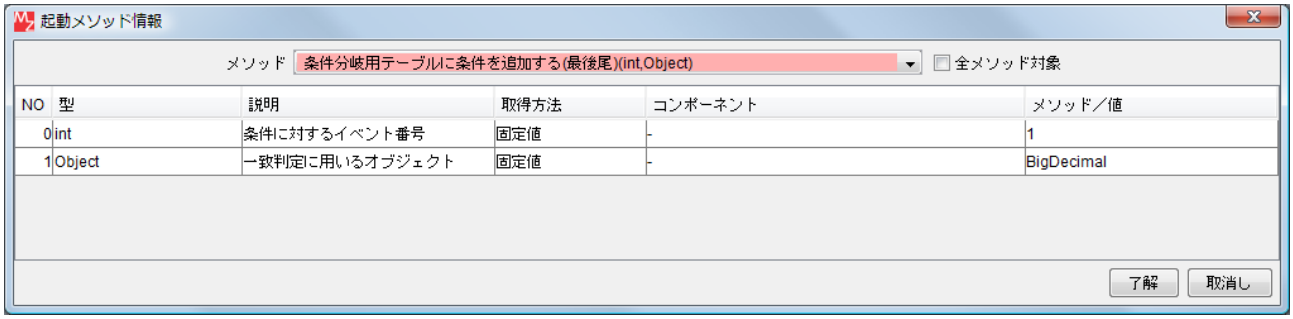
⑪の接続情報：分岐条件を初期化する。



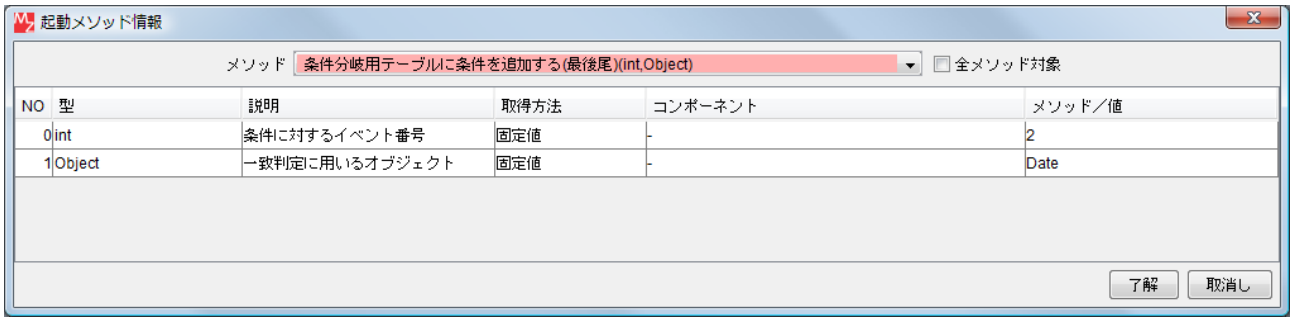
⑫の接続情報：分岐条件を追加する（整数型の場合）。



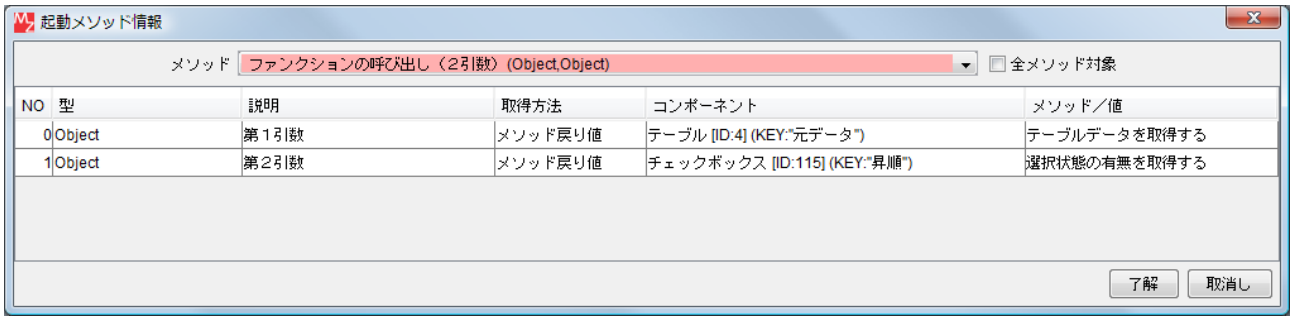
⑬の接続情報：分岐条件を追加する（実数型の場合）。



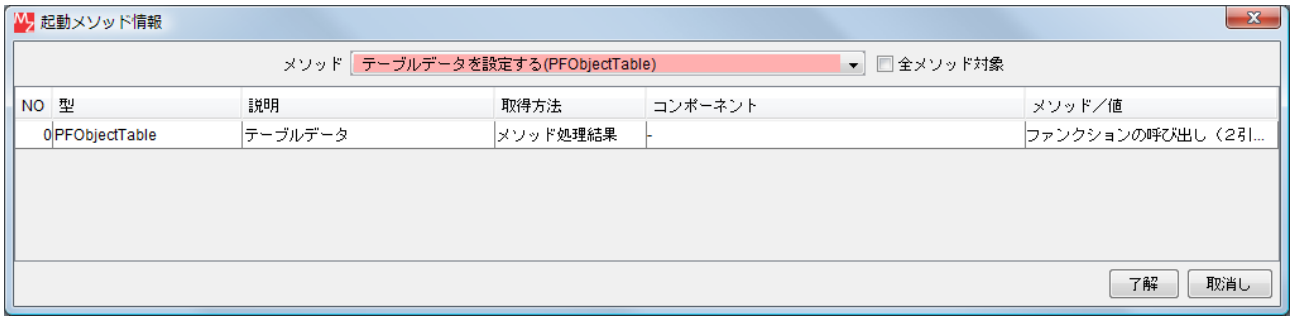
⑭の接続情報：分岐条件を追加する（日付型の場合）。



⑮の接続情報：定義したソート処理を実行する。



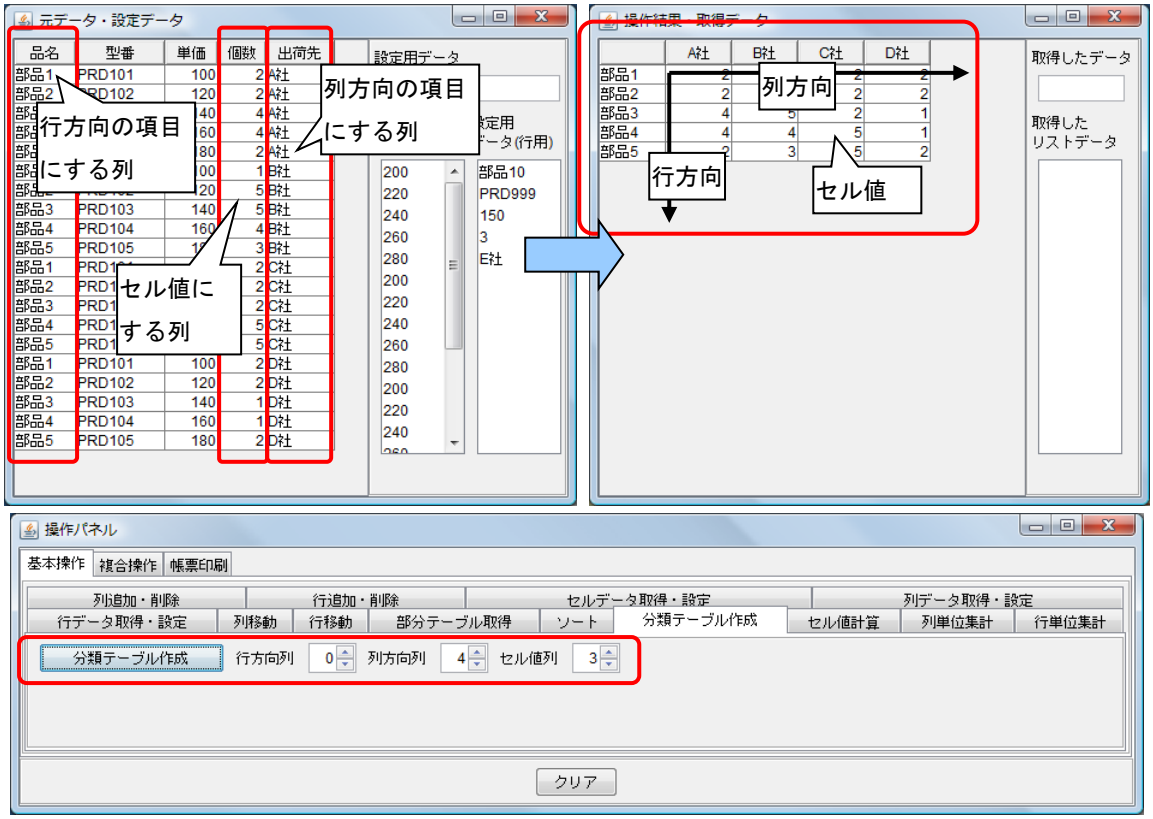
⑯の接続情報：結果をテーブルに設定する。



2.16 分類テーブルの作成

動作イメージ

以下の図のように、テーブルデータから列指定（行方向の項目にする列、列方向の項目にする列、セル値にする列）の情報に基づいて分類テーブルを生成する処理手順について説明します。



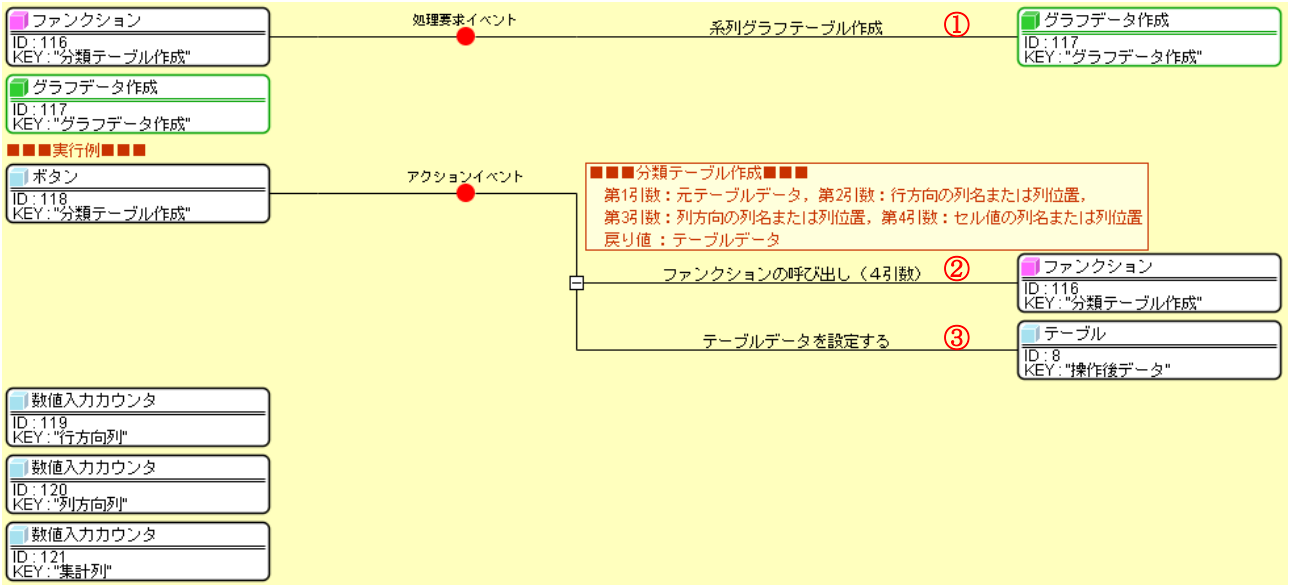
使用メソッド

テーブルデータから列指定（行方向の項目にする列、列方向の項目にする列、セル値にする列）の情報に基づいて分類テーブルを生成するために使用するコンポーネント、基本メソッドおよび設定する引数は次のとおりです。

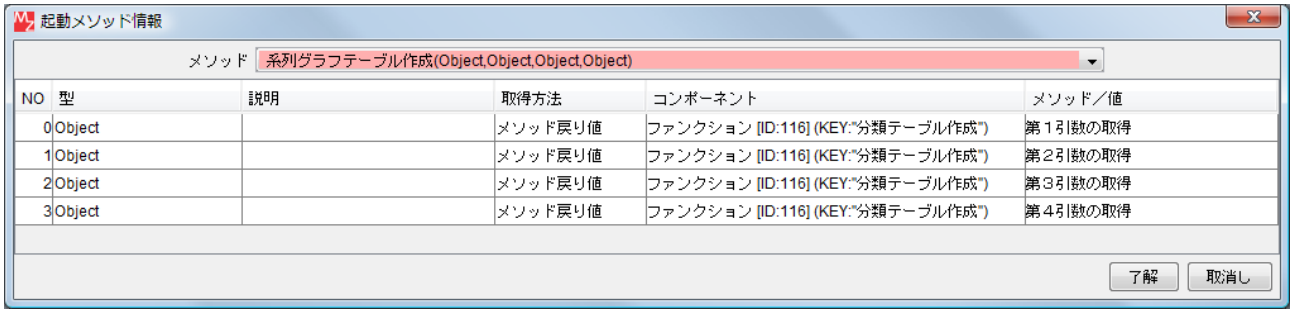
コンポーネント	■グラフデータ作成（複合コンポーネント）
メソッド	系列グラフテーブル作成(Object, Object, Object, Object)
引数	第 1 引数：元テーブルデータ 第 2 引数：行方向の列名または列位置 第 3 引数：列方向の列名または列位置 第 4 引数：セル値の列名または列位置
戻り値	PFObjectTable
列情報を指定して、分類されたテーブルを生成します。	

接続例

動作イメージのように、「数値入力カウンタ」で指定された列情報（**行方向の列位置、列方向の列位置、集計対象の列位置**）に従い左側テーブルデータから分類テーブルを生成し、その結果を右側の「テーブル」に設定する処理手順を以下のように定義します。ここで分類テーブル生成処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に行方向の列位置、第 3 引数：列方向の列位置、第 4 引数：集計対象の列位置を指定してメソッド「ファンクションの呼び出し（4 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。

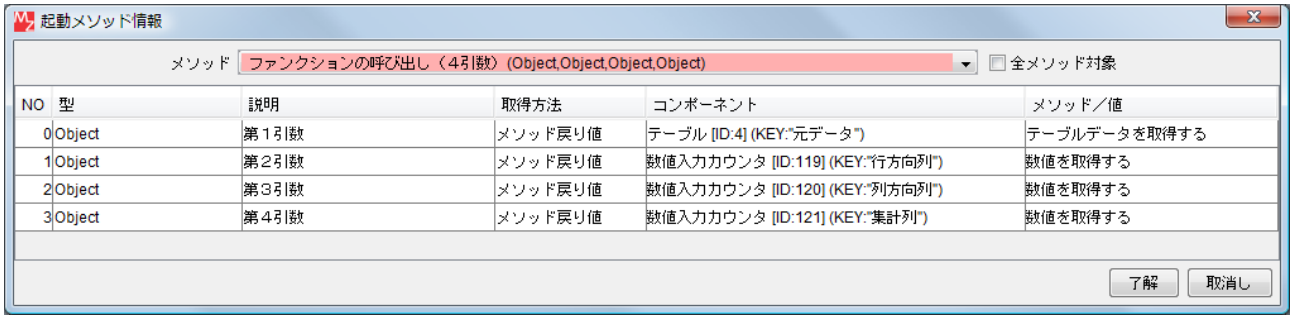


①の接続情報：列情報を指定し、分類テーブルを生成する。



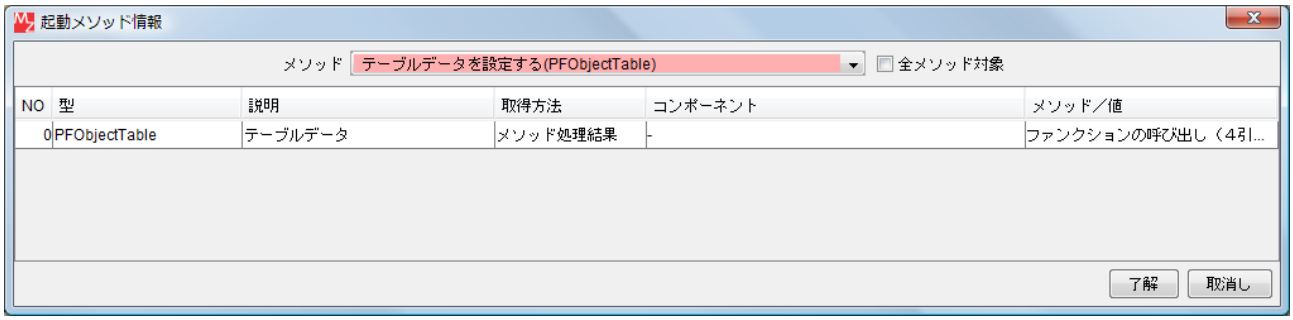
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object		メソッド戻り値	ファンクション [ID:116] (KEY:"分類テーブル作成")	第１引数の取得
1	Object		メソッド戻り値	ファンクション [ID:116] (KEY:"分類テーブル作成")	第２引数の取得
2	Object		メソッド戻り値	ファンクション [ID:116] (KEY:"分類テーブル作成")	第３引数の取得
3	Object		メソッド戻り値	ファンクション [ID:116] (KEY:"分類テーブル作成")	第４引数の取得

②の接続情報：定義した分類テーブル生成処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第１引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第２引数	メソッド戻り値	数値入力カウンタ [ID:119] (KEY:"行方向列")	数値を取得する
2	Object	第３引数	メソッド戻り値	数値入力カウンタ [ID:120] (KEY:"列方向列")	数値を取得する
3	Object	第４引数	メソッド戻り値	数値入力カウンタ [ID:121] (KEY:"集計列")	数値を取得する

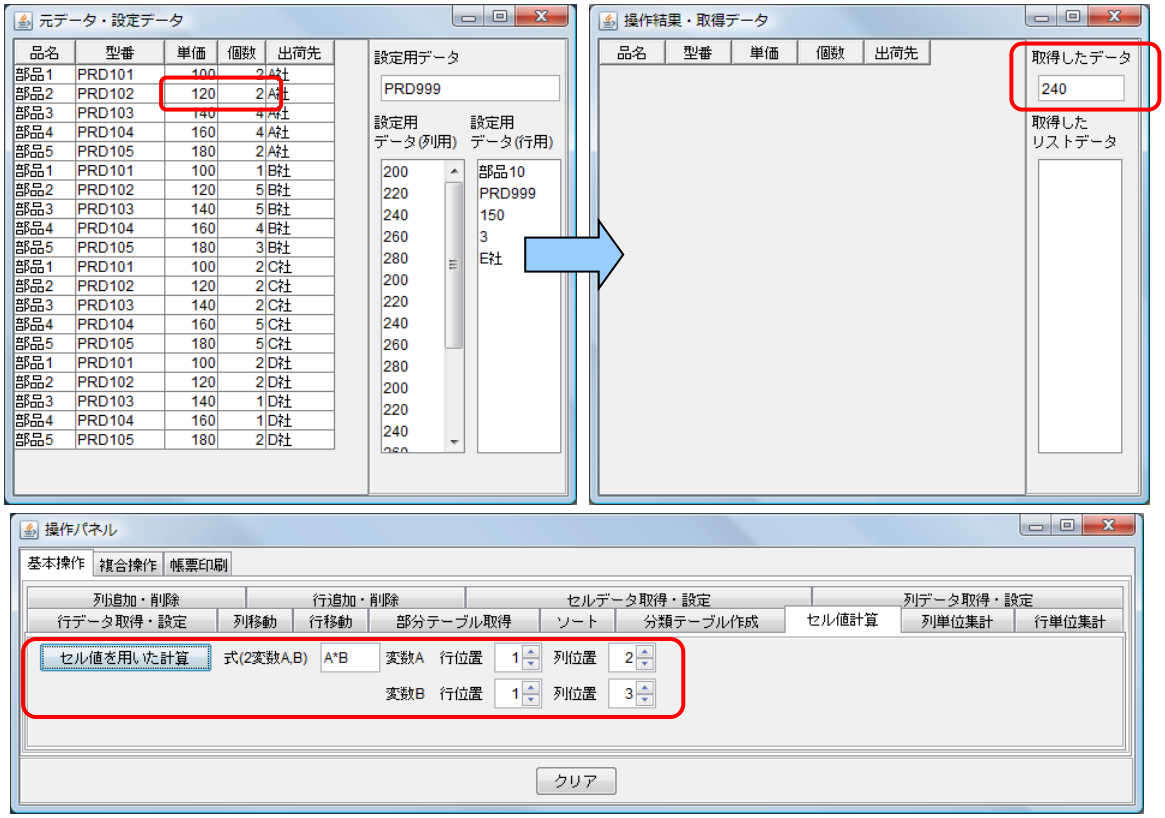
③の接続情報：結果をテーブルに設定する。



2.17 セル値を用いた計算

動作イメージ

以下の図のように、テーブルデータのセルの値を用いて計算を行う処理手順について説明します。



使用メソッド

テーブルデータのセルの値を用いて計算を行うためには、次のコンポーネントのメソッドを使用します。

①セルの値を取得する

コンポーネント	■テーブル格納変数
メソッド	セルデータを位置指定で取得する(int, int)
引数	第１引数：行位置 第２引数：列位置
戻り値	Object
行位置および列位置を指定してセルデータを取得します。	

②計算式の文字列を生成する

・計算式（例 A*B）を設定する

コンポーネント	■文字列格納変数
メソッド	文字列を設定する(String)
引数	第１引数：文字列
戻り値	なし
文字列（変数を含む計算式を表現する文字列）を設定します。	

・変数を数値で置き換える（代入）

メソッド	指定文字列と一致する最初の文字列を置換する(String, String)
引数	第 1 引数：置換前の文字列 第 2 引数：置換後の文字列
戻り値	なし
文字列を置き換えます（変数に数値を代入した計算式を表現する文字列を生成する）。	

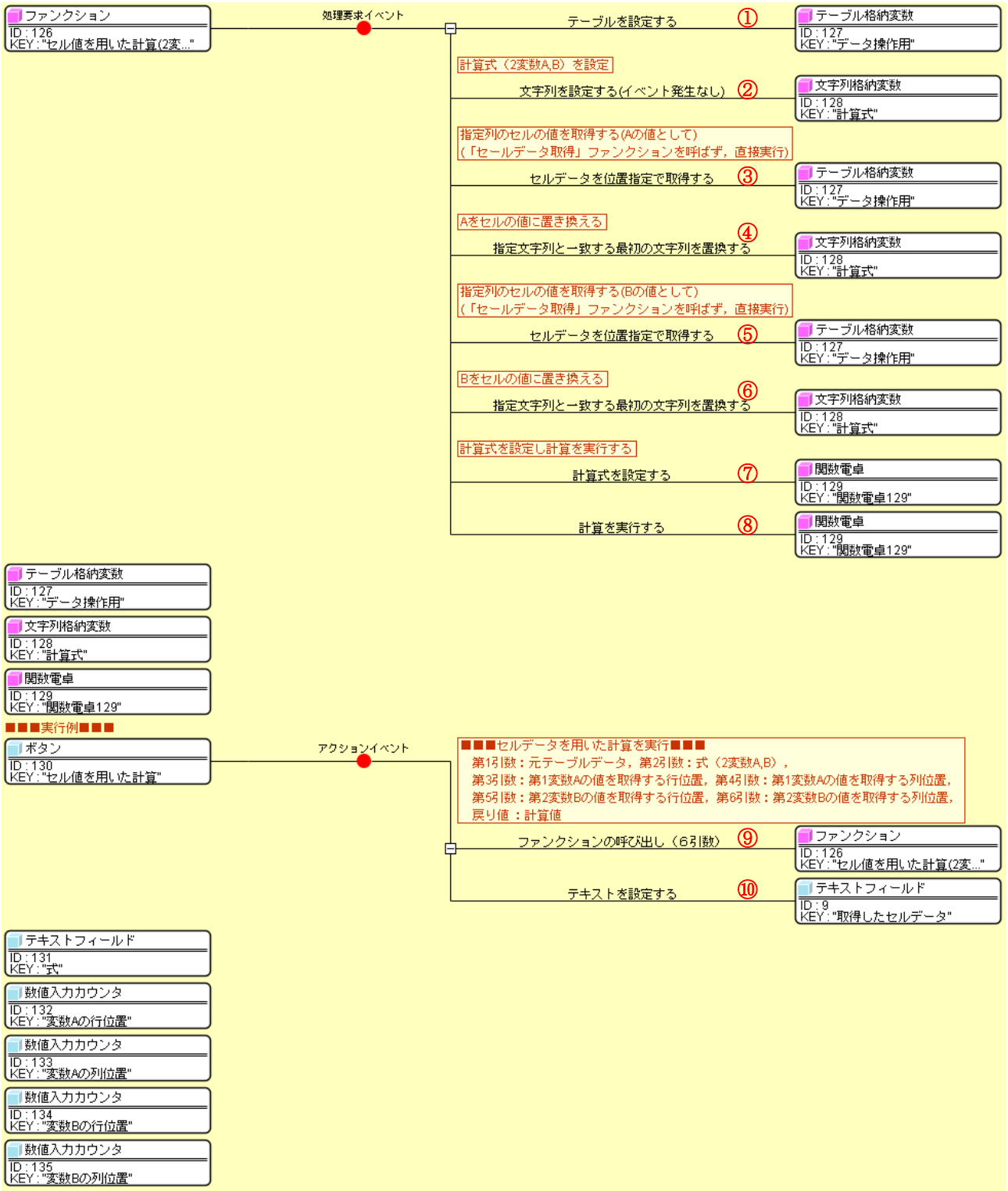
③計算式を設定し実行する

コンポーネント	■関数電卓
メソッド	計算式を設定する(String)
引数	第 1 引数：計算式をあらわす文字列
戻り値	なし
計算に当たり計算式をあらかじめ文字列で設定します。	

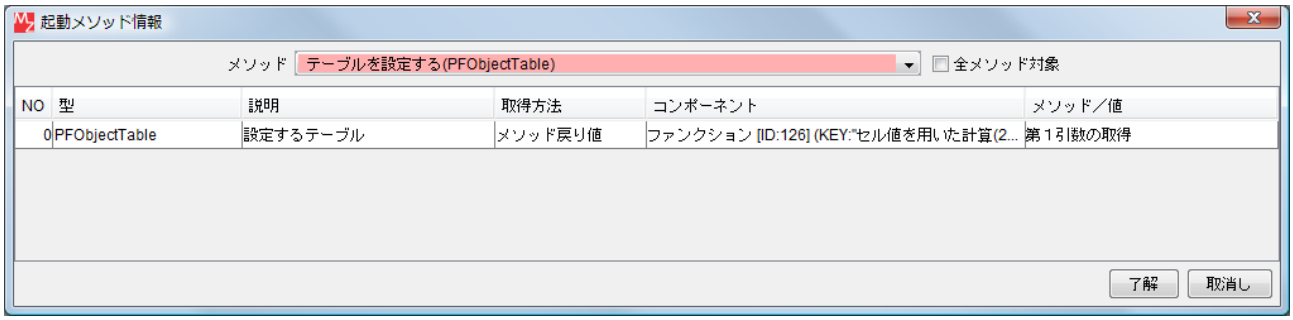
メソッド	計算を実行する()
引数	なし
戻り値	BigDecimal（数値）
計算を実行します。	

接続例

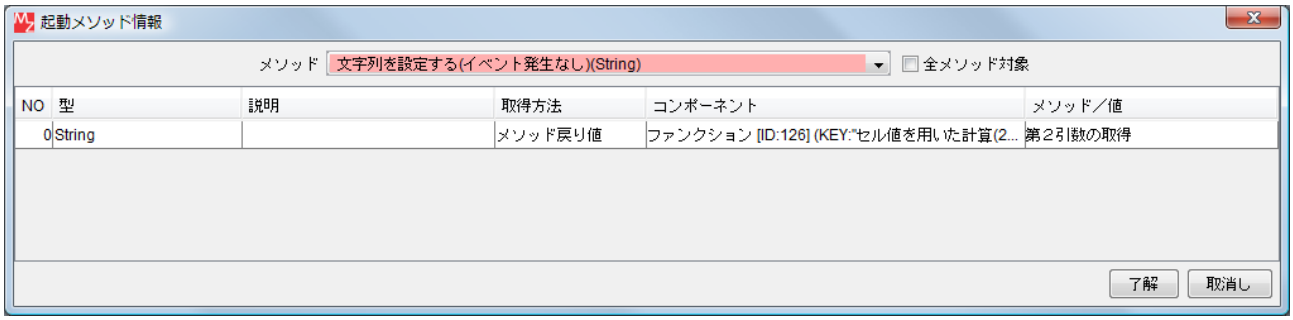
動作イメージのように、「数値入力カウンタ」で指定された位置のセルの値を左側テーブルデータから取得し、「テキストフィールド」で指定された計算式で計算を行った結果を「テキストフィールド」に設定する処理手順を以下のように定義します。ここでセル値の計算処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に式（2 変数 A,B）の文字列、第 3 引数に第 1 変数 A の値を取得する行位置、第 4 引数に第 1 変数 A の値を取得する列位置、第 5 引数に第 2 変数 B の値を取得する行位置、第 6 引数に第 2 変数 B の値を取得する列位置を指定してメソッド「ファンクションの呼び出し（6 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。


ファンクション
ID : 126
KEY : "セル値を用いた計算(2変..."
処理要求イベント
テーブルを設定する ①
テーブル格納変数
ID : 127
KEY : "データ操作用"
計算式（2変数A,B）を設定
文字列を設定する(イベント発生なし) ②
文字列格納変数
ID : 128
KEY : "計算式"
指定列のセルの値を取得する(Aの値として)
(「セールデータ取得」ファンクションを呼ばず，直接実行)
セルデータを位置指定で取得する ③
テーブル格納変数
ID : 127
KEY : "データ操作用"
Aをセルの値に置き換える
④
指定文字列と一致する最初の文字列を置換する
文字列格納変数
ID : 128
KEY : "計算式"
指定列のセルの値を取得する(Bの値として)
(「セールデータ取得」ファンクションを呼ばず，直接実行)
セルデータを位置指定で取得する ⑤
テーブル格納変数
ID : 127
KEY : "データ操作用"
Bをセルの値に置き換える
⑥
指定文字列と一致する最初の文字列を置換する
文字列格納変数
ID : 128
KEY : "計算式"
計算式を設定し計算を実行する
計算式を設定する ⑦
関数電卓
ID : 129
KEY : "関数電卓129"
計算を実行する ⑧
関数電卓
ID : 129
KEY : "関数電卓129"
テーブル格納変数
ID : 127
KEY : "データ操作用"
文字列格納変数
ID : 128
KEY : "計算式"
関数電卓
ID : 129
KEY : "関数電卓129"
■■■実行例■■■
ボタン
ID : 130
KEY : "セル値を用いた計算"
アクションイベント
■■■セルデータを用いた計算を実行■■■
第1引数：元テーブルデータ，第2引数：式（2変数A,B），
第3引数：第1変数Aの値を取得する行位置，第4引数：第1変数Aの値を取得する列位置，
第5引数：第2変数Bの値を取得する行位置，第6引数：第2変数Bの値を取得する列位置，
戻り値 ：計算値
ファンクションの呼び出し（6引数） ⑨
ファンクション
ID : 126
KEY : "セル値を用いた計算(2変..."
テキストを設定する ⑩
テキストフィールド
ID : 9
KEY : "取得したセルデータ"
テキストフィールド
ID : 131
KEY : "式"
数値入力カウンタ
ID : 132
KEY : "変数Aの行位置"
数値入力カウンタ
ID : 133
KEY : "変数Aの列位置"
数値入力カウンタ
ID : 134
KEY : "変数Bの行位置"
数値入力カウンタ
ID : 135
KEY : "変数Bの列位置"

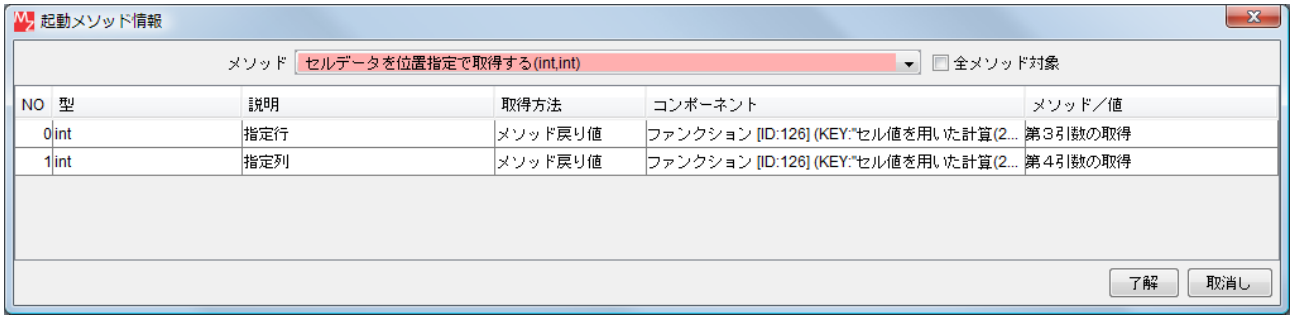
①の接続情報：処理対象のテーブルデータを設定する。



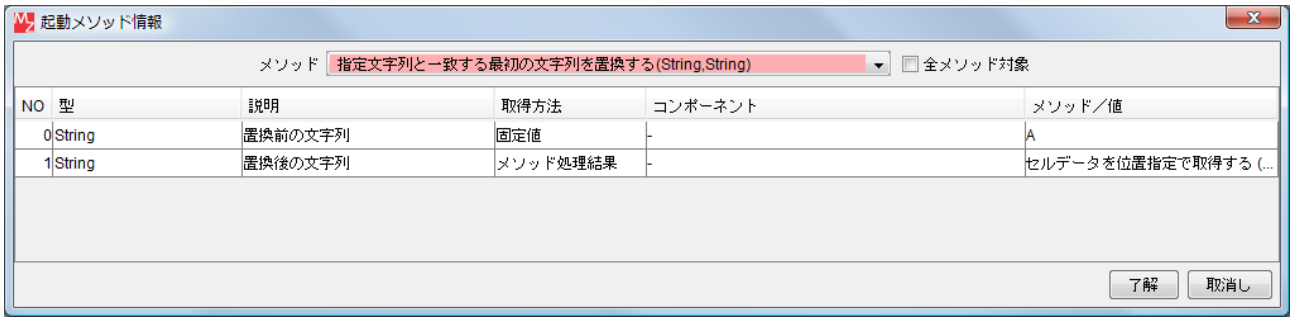
②の接続情報：計算式（変数 A,B を含む）を表現する文字列を設定する。



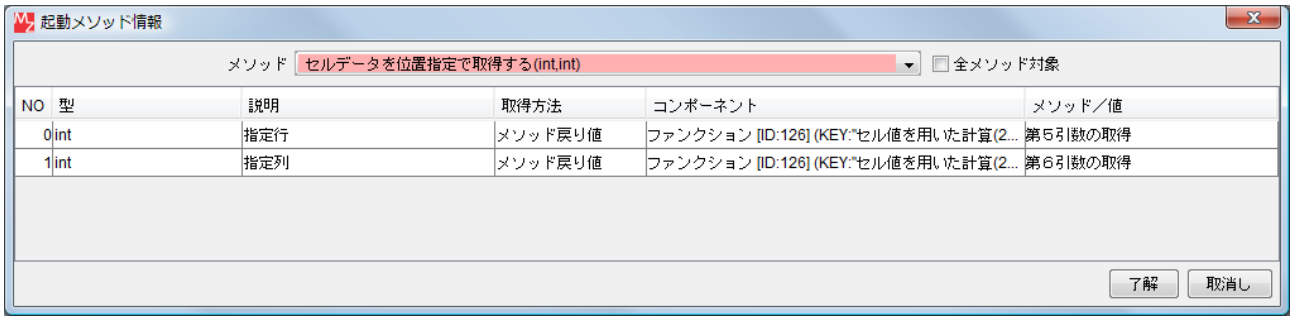
③の接続情報：テーブルからセルの値を取得する。



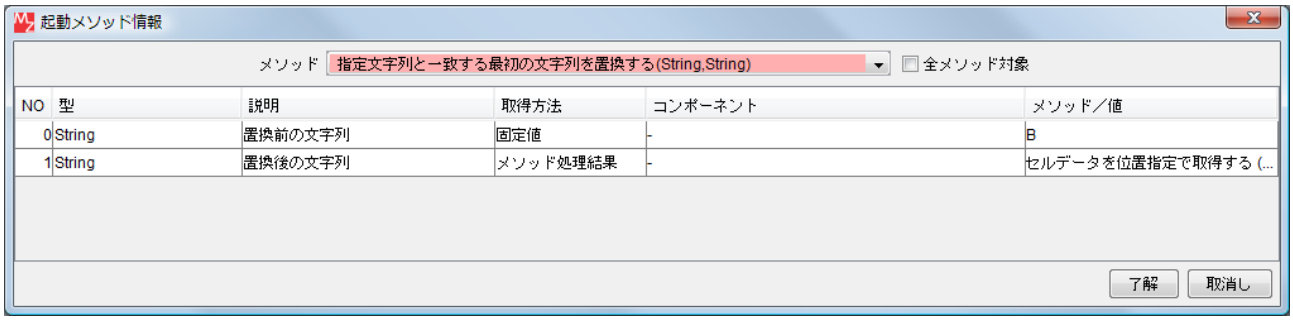
④の接続情報：文字列を置換し変数に値を代入した式を表現する文字列を生成する。



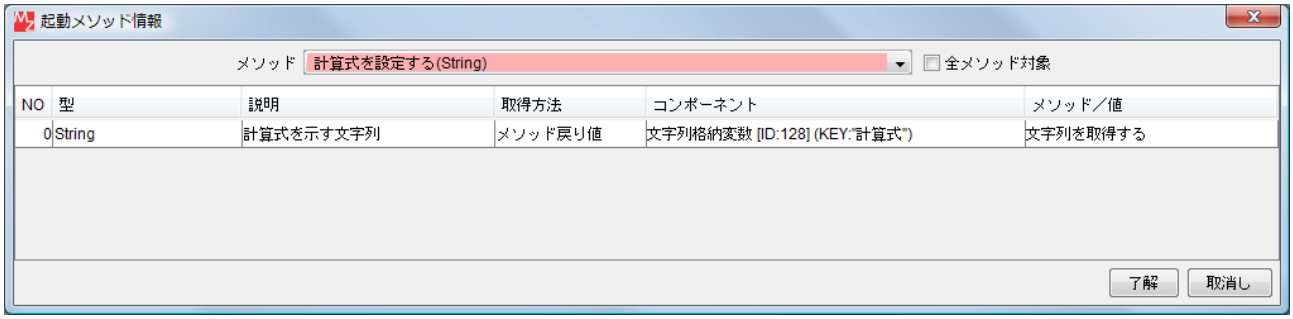
⑤の接続情報：テーブルからセルの値を取得する。



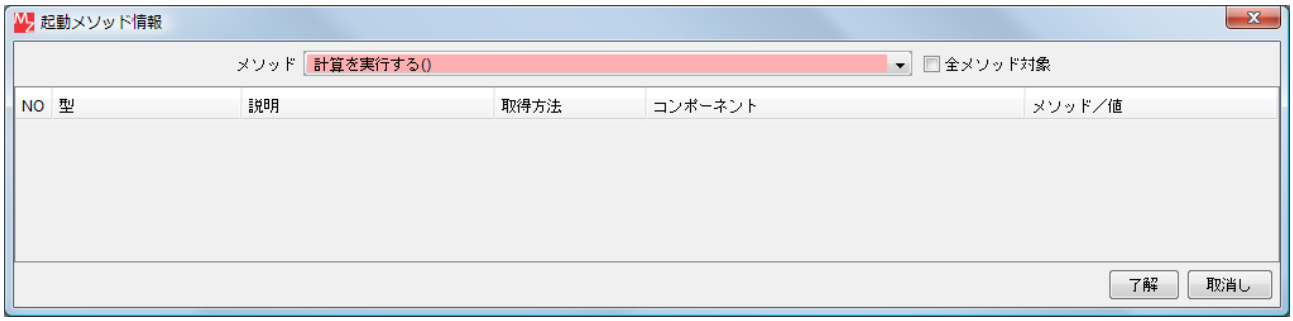
⑥の接続情報：文字列を置換し変数に値を代入した式を表現する文字列を生成する。



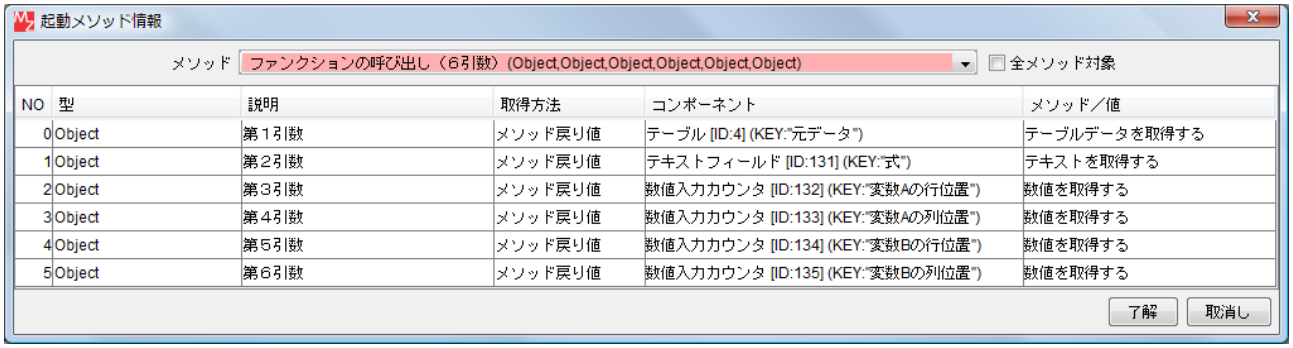
⑦の接続情報：計算に当たり計算式を設定する。



⑧の接続情報：計算を実行し、ファンクションの戻り値として返す。

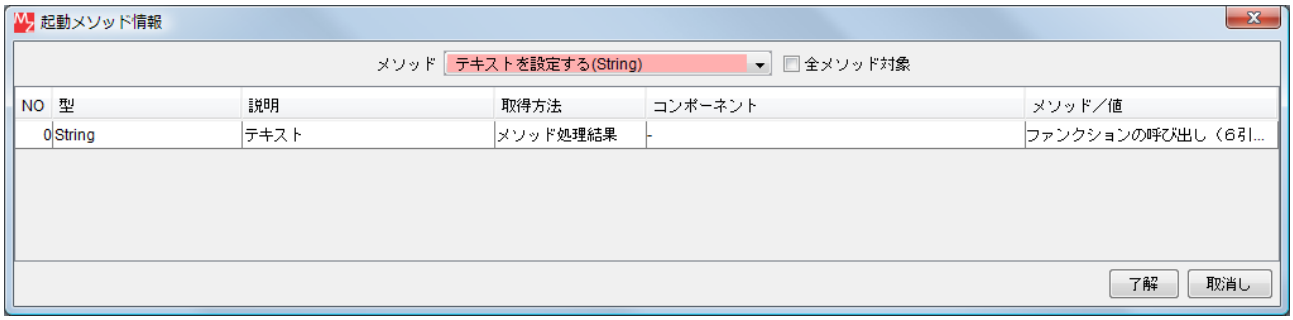


⑨の接続情報：定義したセル値計算処理を実行する



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	メソッド戻り値	テキストフィールド [ID:131] (KEY:"式")	テキストを取得する
2	Object	第3引数	メソッド戻り値	数値入力カウンタ [ID:132] (KEY:"変数Aの行位置")	数値を取得する
3	Object	第4引数	メソッド戻り値	数値入力カウンタ [ID:133] (KEY:"変数Aの列位置")	数値を取得する
4	Object	第5引数	メソッド戻り値	数値入力カウンタ [ID:134] (KEY:"変数Bの行位置")	数値を取得する
5	Object	第6引数	メソッド戻り値	数値入力カウンタ [ID:135] (KEY:"変数Bの列位置")	数値を取得する

⑩の接続情報：結果をテキストフィールドに設定する。

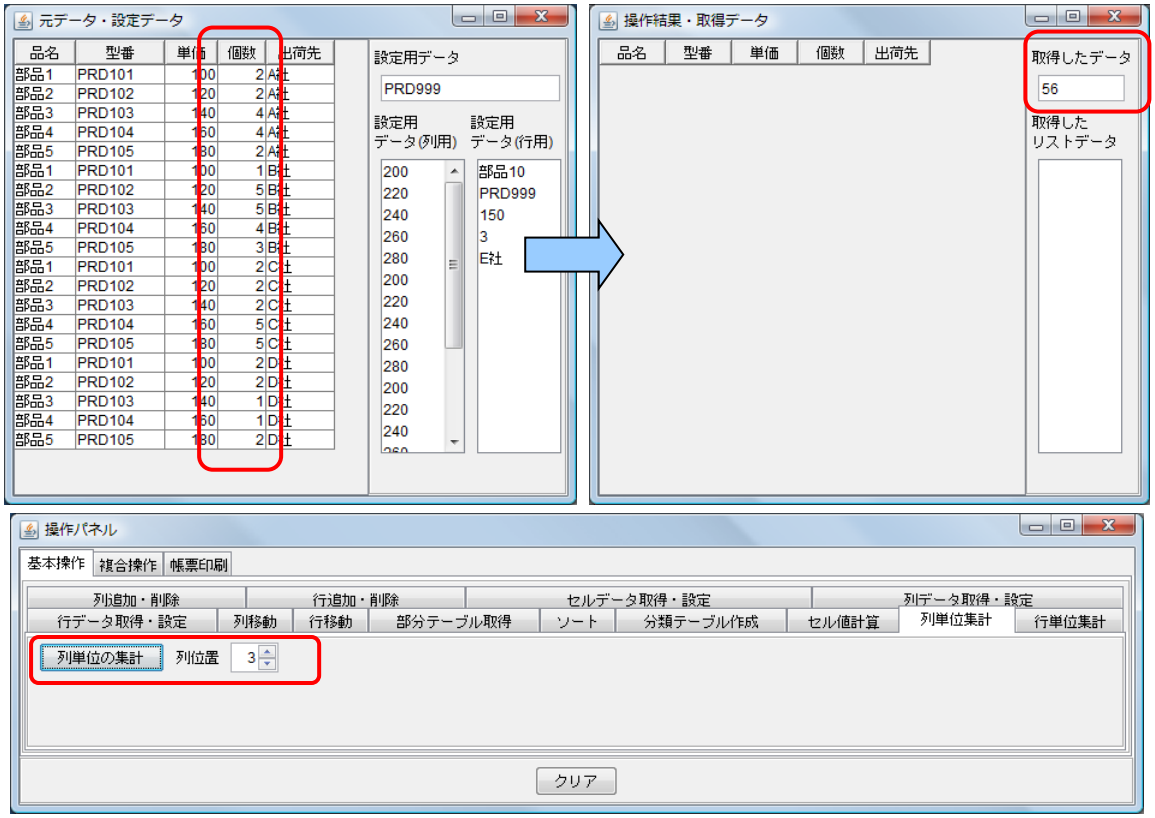


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	String	テキスト	メソッド処理結果	-	ファンクションの呼び出し（6引...

2.18 列単位の集計

動作イメージ

以下の図のように、テーブルデータのデータを指定列について集計する処理手順について説明します。



使用メソッド

テーブルデータを指定列について集計するため、次のコンポーネントのメソッドを使用します。

①集計対象の列のデータを取得する

コンポーネント	■テーブル格納変数
メソッド	列データリストを位置指定で取得する(int)
引数	第１引数：列位置
戻り値	PFObjectList
列位置を指定して列データ（リスト）を取得します。非数値データは無視されます。	

②リストデータを集計する

・集計対象データを設定する

コンポーネント	■基本統計処理(数値)
メソッド	集計対象データを設定する(PFObjectList)
引数	第１引数：リストデータ
戻り値	なし
集計対象データをリストとして設定します。	

・合計値（和）を取得する

メソッド	和を取得する()
引数	なし
戻り値	BigDecimal（数値）
和を計算します。	

類似する処理行うメソッドとして次のメソッドがあります。

メソッド	中央値を取得する()
引数	なし
戻り値	BigDecimal（数値）
中央値を計算します。	

メソッド	分散値を取得する()
引数	なし
戻り値	BigDecimal（数値）
分散値を計算します。	

メソッド	平均値を取得する()
引数	なし
戻り値	BigDecimal（数値）
平均値を計算します。	

メソッド	平方和（二乗和）を取得する()
引数	なし
戻り値	BigDecimal（数値）
平方和（二乗和）を計算します。	

メソッド	最大値を取得する()
引数	なし
戻り値	BigDecimal（数値）
最大値を計算します。	

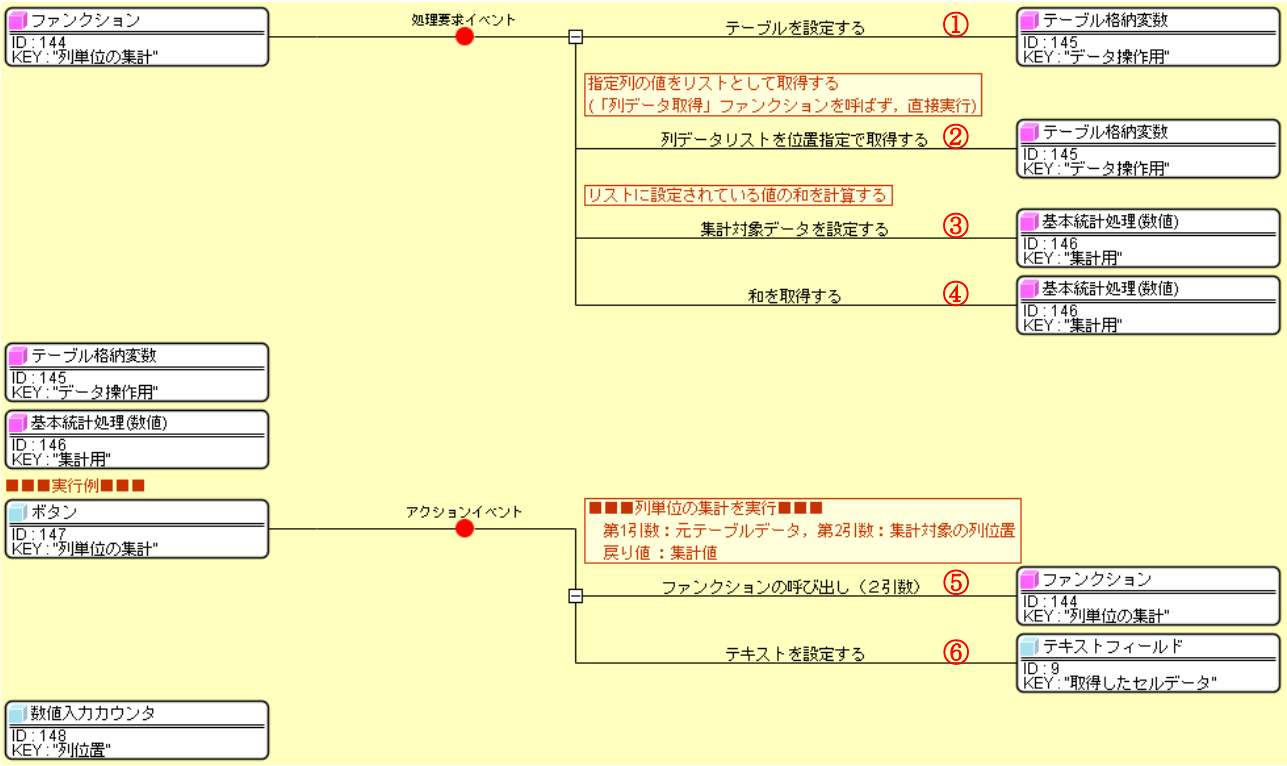
メソッド	最小値を取得する()
引数	なし
戻り値	BigDecimal（数値）
最小値を計算します。	

メソッド	最頻値を取得する()
引数	なし
戻り値	BigDecimal（数値）
最頻値を計算します。	

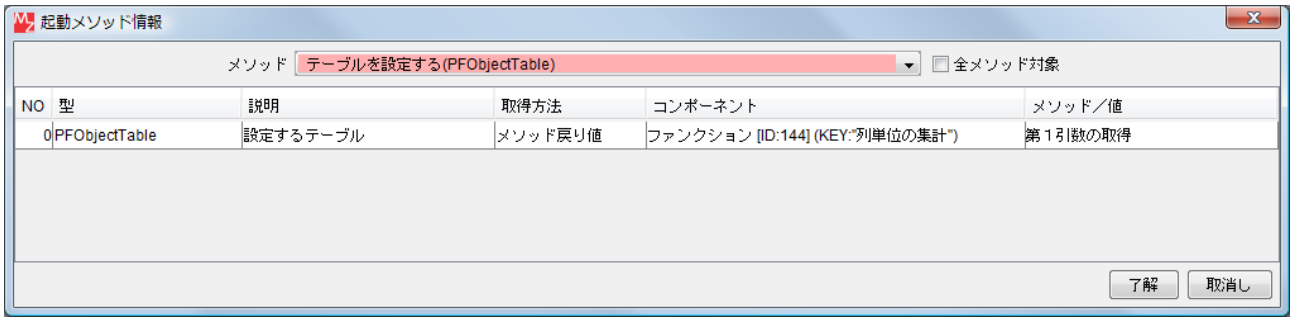
メソッド	標準偏差を取得する()
引数	なし
戻り値	BigDecimal（数値）
標準偏差を計算します。	

接続例

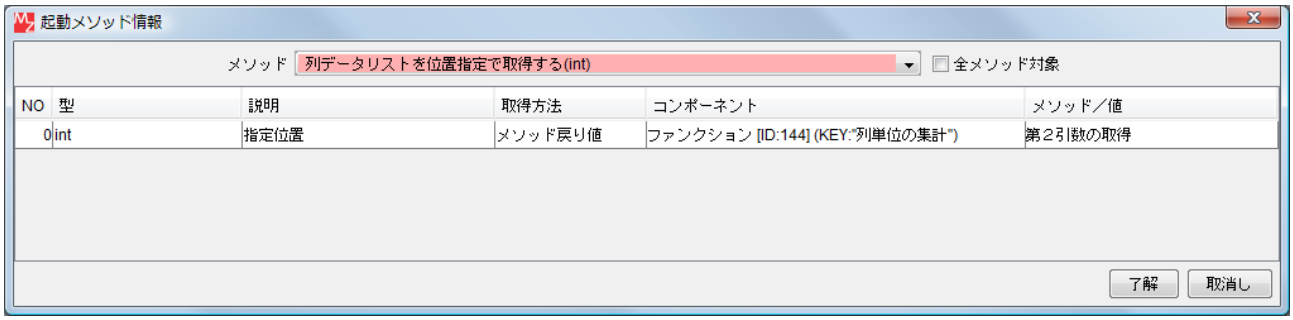
動作イメージのように、「数値入力カウンタ」で指定された列位置のデータについて集計を行い、その結果を右側の「テキストフィールド」に設定する処理手順を以下のように定義します。ここで列単位集計処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に集計対象の列位置を指定してメソッド「ファンクションの呼び出し（2 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。



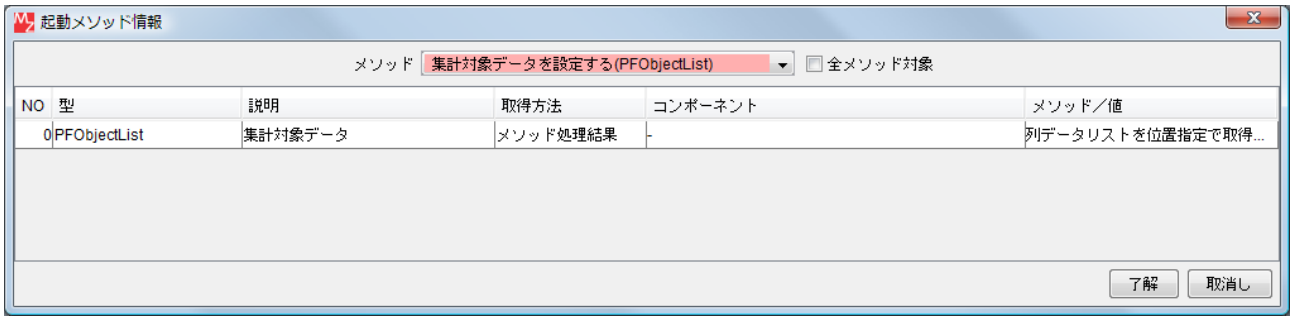
①の接続情報：処理対象のテーブルデータを設定する。



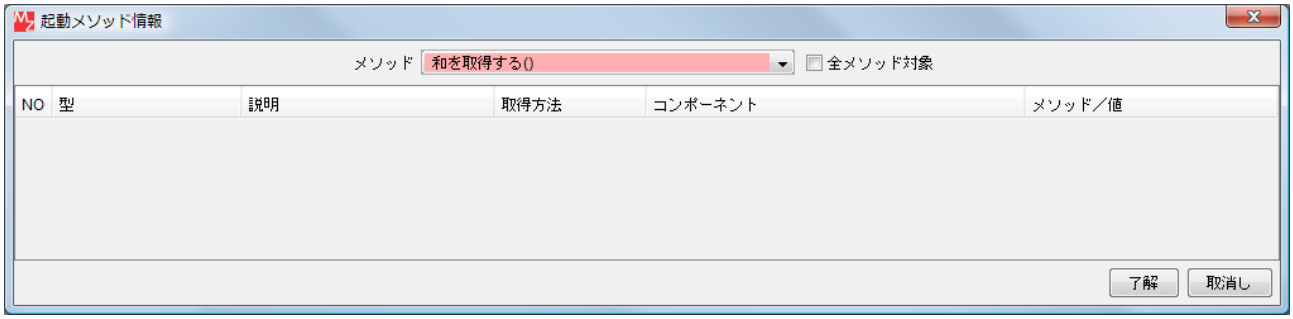
②の接続情報：集計対象の列のデータを取得する。



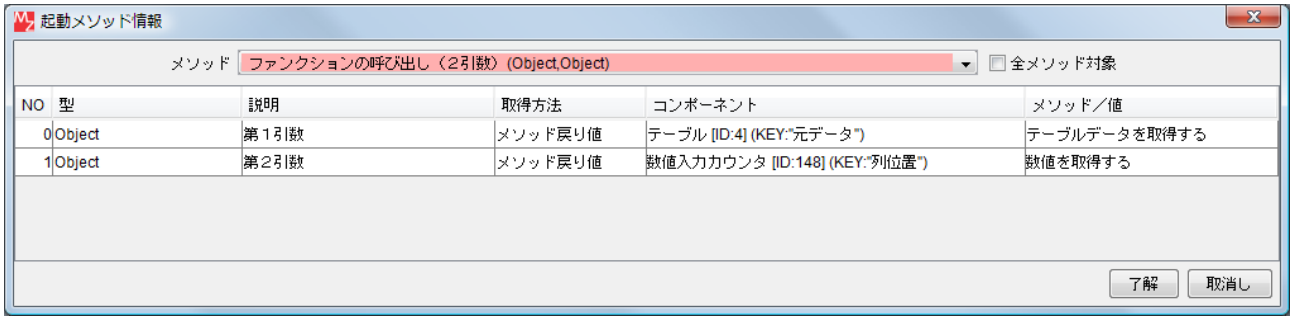
③の接続情報：集計対象データを設定する。



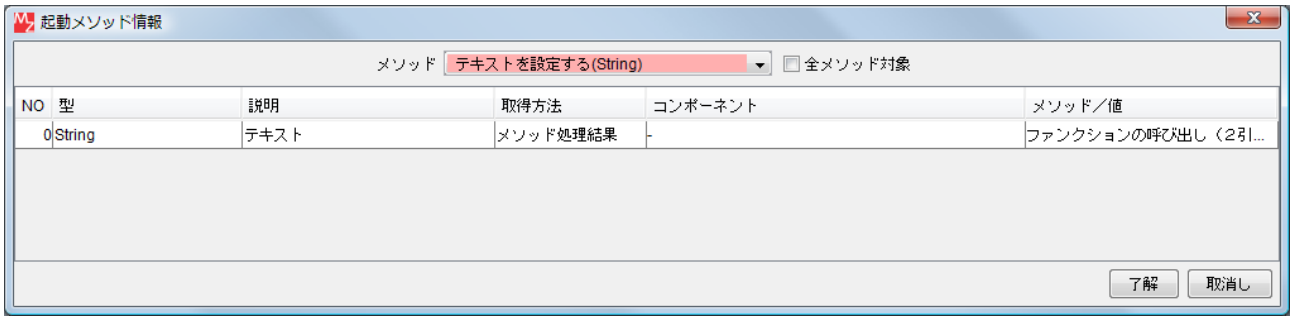
④の接続情報：和を計算し、ファンクションの戻り値として返す。



⑤の接続情報：定義した列単位の集計処理を実行する。



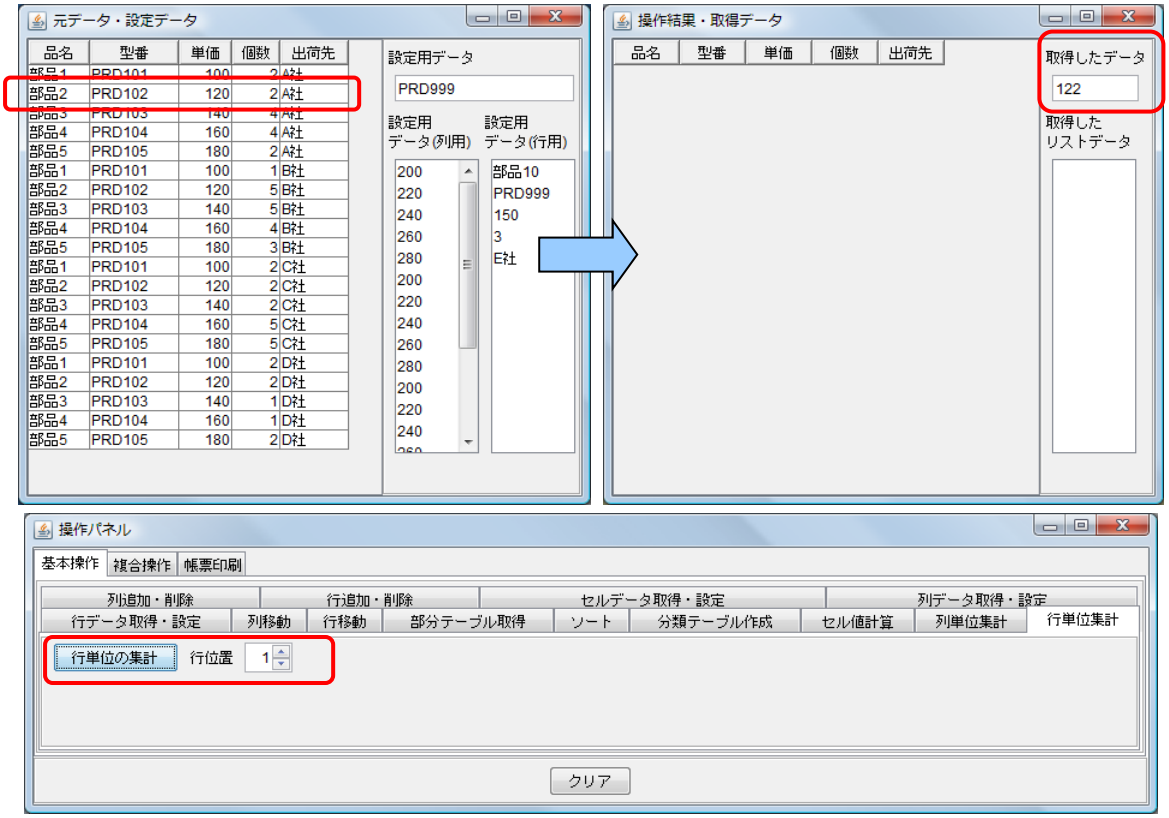
⑥の接続情報：結果をテキストフィールドに設定する。



2.19 行単位の集計

動作イメージ

以下の図のように、テーブルデータのデータを指定行について集計する処理手順について説明します。



使用メソッド

テーブルデータを指定列について集計するため、次のコンポーネントのメソッドを使用します。

①集計対象の列のデータを取得する

コンポーネント	■テーブル格納変数
メソッド	行データリストを位置指定で取得する(int)
引数	第１引数：行位置
戻り値	PFObjectList
行位置を指定して行データ（リスト）を取得します。	

②リストデータを集計する

・集計対象データを設定する

コンポーネント	■基本統計処理(数値)
メソッド	集計対象データを設定する(PFObjectList)
引数	第１引数：リストデータ
戻り値	なし
集計対象データをリストとして設定します。非数値データは無視されます。	

・合計値（和）を取得する

メソッド	和を取得する()
引数	なし
戻り値	BigDecimal（数値）
和を計算します。	

類似する処理行うメソッドとして次のメソッドがあります。

メソッド	中央値を取得する()
引数	なし
戻り値	BigDecimal（数値）
中央値を計算します。	

メソッド	分散値を取得する()
引数	なし
戻り値	BigDecimal（数値）
分散値を計算します。	

メソッド	平均値を取得する()
引数	なし
戻り値	BigDecimal（数値）
平均値を計算します。	

メソッド	平方和（二乗和）を取得する()
引数	なし
戻り値	BigDecimal（数値）
平方和（二乗和）を計算します。	

メソッド	最大値を取得する()
引数	なし
戻り値	BigDecimal（数値）
最大値を計算します。	

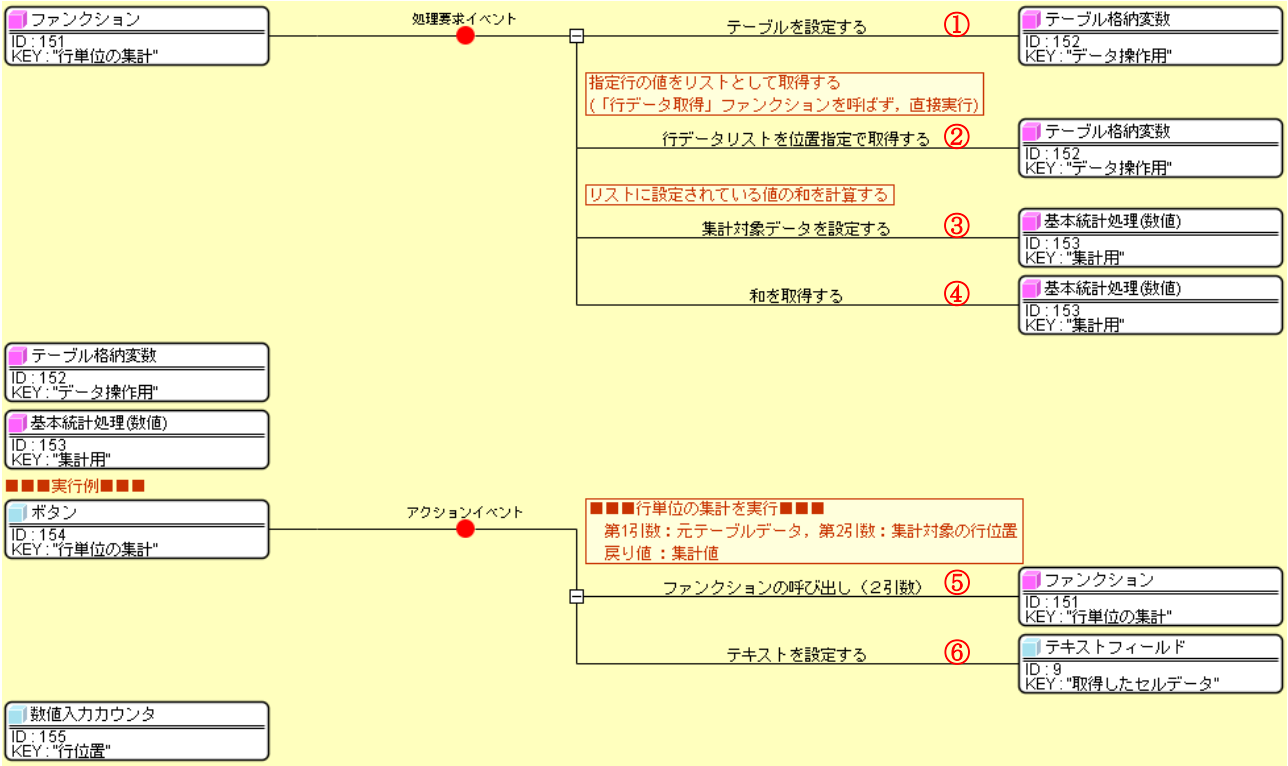
メソッド	最小値を取得する()
引数	なし
戻り値	BigDecimal（数値）
最小値を計算します。	

メソッド	最頻値を取得する()
引数	なし
戻り値	BigDecimal（数値）
最頻値を計算します。	

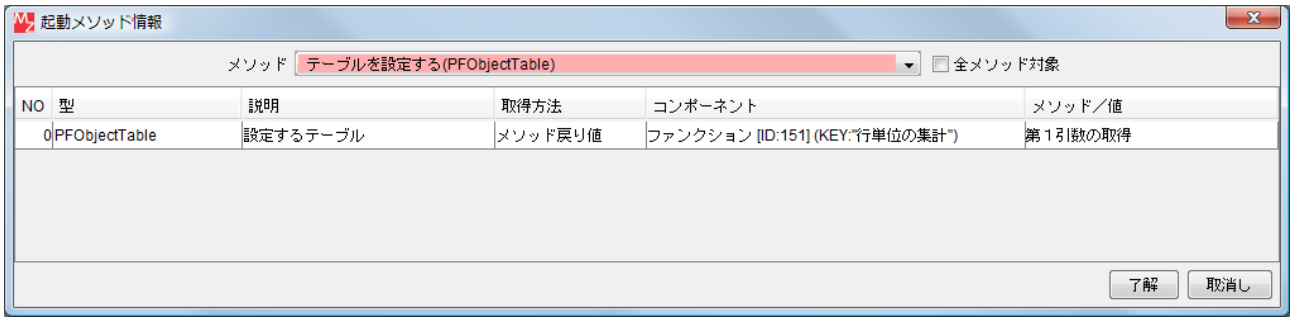
メソッド	標準偏差を取得する()
引数	なし
戻り値	BigDecimal（数値）
標準偏差を計算します。	

接続例

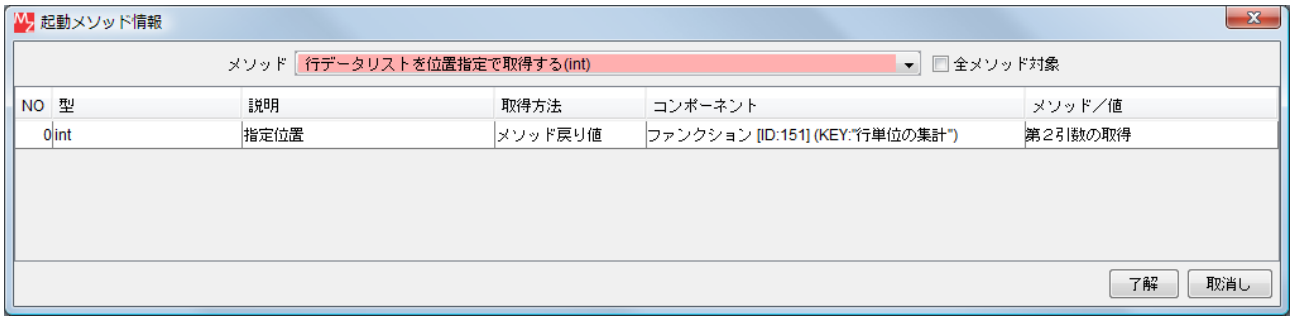
動作イメージのように、「数値入力カウンタ」で指定された行位置のデータについて集計を行い、その結果を右側の「テキストフィールド」に設定する処理手順を以下のように定義します。ここで行単位集計処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に集計対象の行位置を指定してメソッド「ファンクションの呼び出し（2 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。



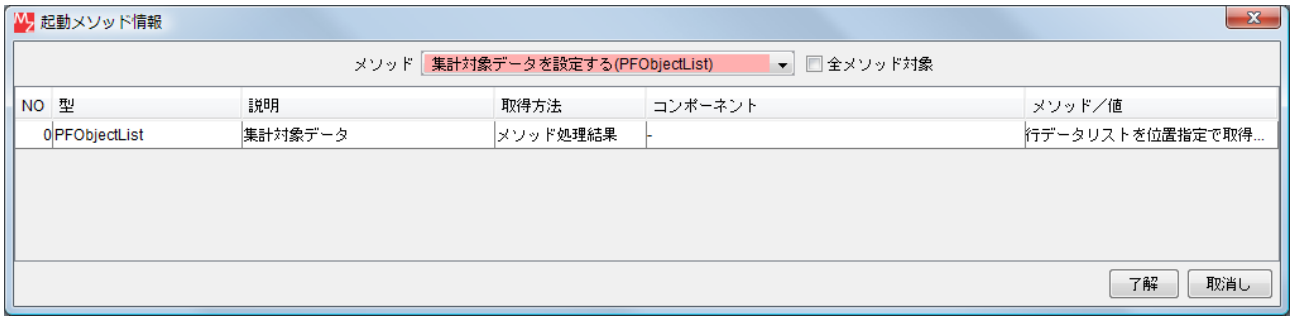
①の接続情報：処理対象のテーブルデータを設定する。



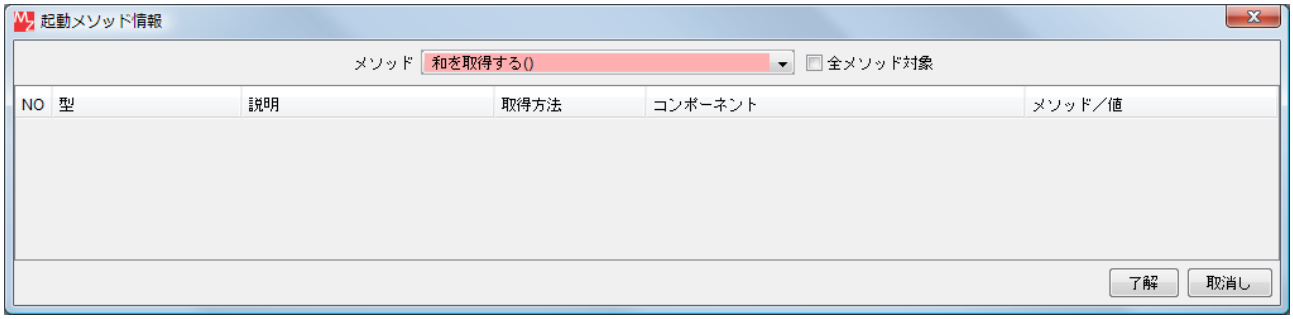
②の接続情報：集計対象の行のデータを取得する。



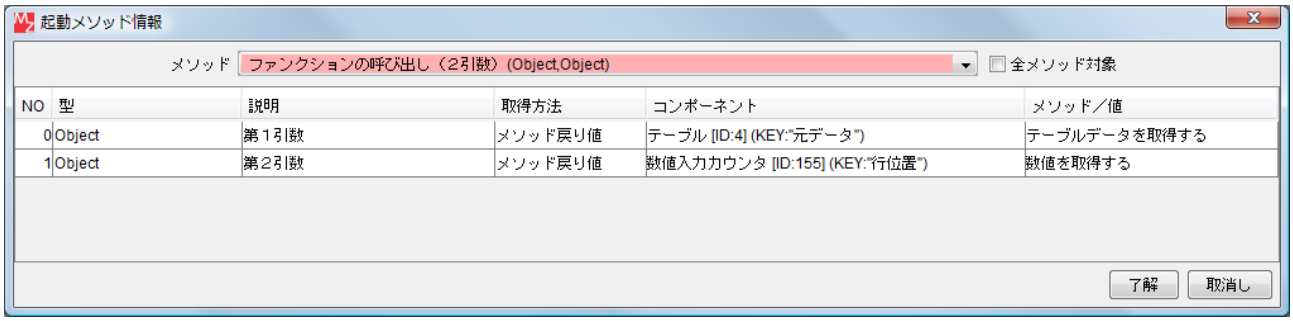
③の接続情報：集計対象データを設定する。



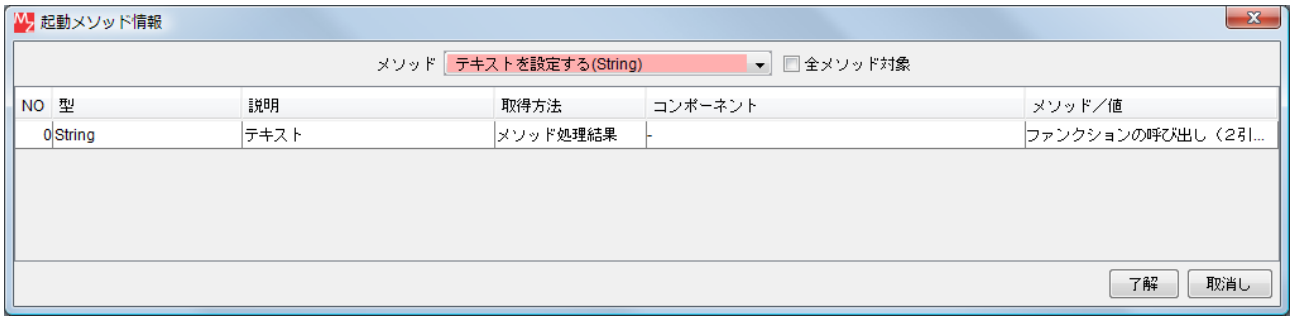
④の接続情報：和を計算し、ファンクションの戻り値として返す。



⑤の接続情報：定義した行単位の集計処理を実行する。



⑥の接続情報：結果をテキストフィールドに設定する。



第3章 テーブルデータ複合操作

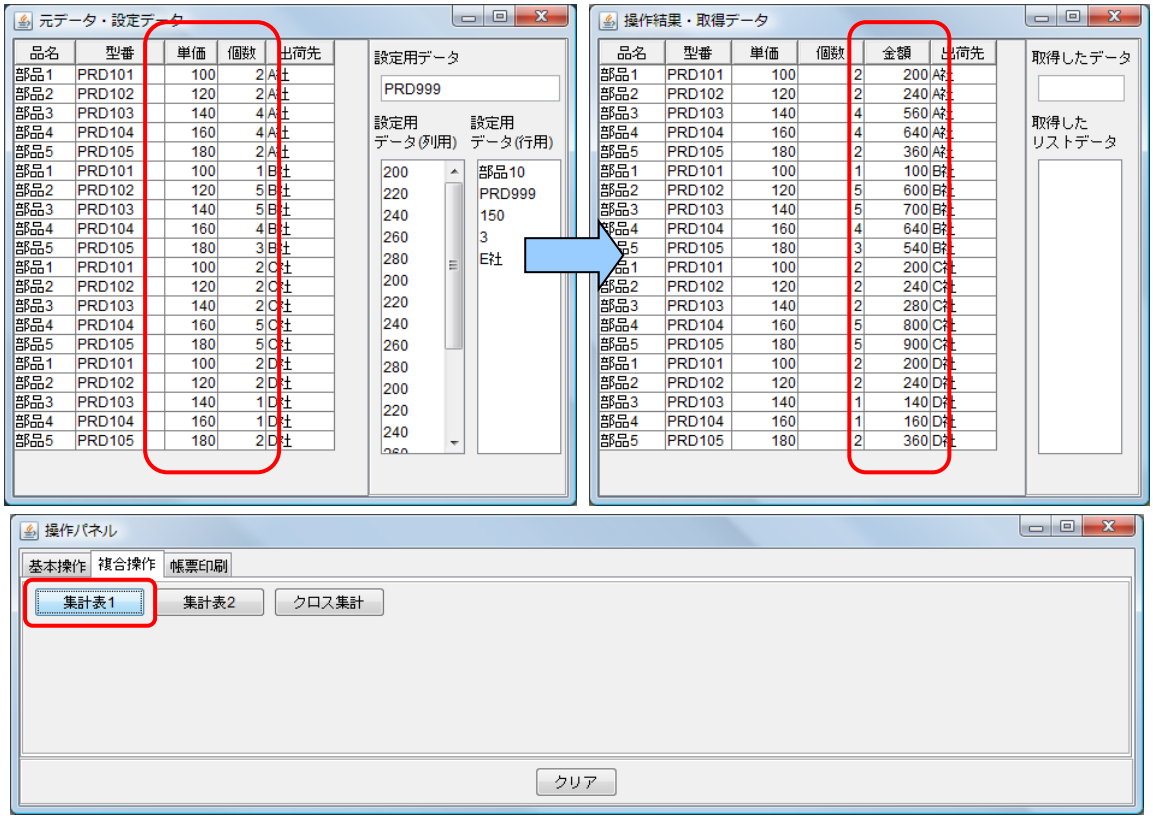
本章では前章の基本操作を組み合わせたテーブルデータ操作の処理手順について説明します。
本章でテーブルデータの処理のために使用する主なコンポーネントは次のとおりです。

コンポーネント名	場所	説明
■テーブル格納変数	*[コンポーネント追加]*－［処理部品］－［変数］－［テーブル格納変数］	テーブルデータを格納し、様々な操作を行うコンポーネント。
■リスト格納変数	*[コンポーネント追加]*－［処理部品］－［変数］－［リスト格納変数］	リストデータを格納し、様々な操作を行うコンポーネント。本章では、各行、各列の集計値を格納するために使用しています。
■繰り返し制御（FOR）	*[コンポーネント追加]*－［処理部品］－［条件制御］－［繰り返し制御(FOR)］	繰り返し処理を行うコンポーネント。本章では、行または列について繰り返し処理を行う際に使用します。
■ファンクション	*[コンポーネント追加]*－［処理部品］－［サブルーチン］－［ファンクション］	与えられた引数に対し、処理結果を返すコンポーネント。処理をまとめるために使用します。本章では、元になるテーブルデータと処理に必要な情報とをファンクションのメソッドに引数として与え、処理結果を戻り値として取得することを基本とします。

3.1 集計表 1

動作イメージ

以下の図のように、テーブルデータの同一行・複数列の値を用いて計算を行い、その結果を追加列に設定する処理手順について説明します。



使用メソッド

この集計表の生成処理は、基本操作の章の「列の追加」および「セル値を用いた計算」を組み合わせた処理で実現します。ただし、列の追加については「列の追加」で定義した「ファンクション」コンポーネントを用いず、テーブル格納変数を用いて直接的に追加を行っています。集計表を作成するために使用する主なコンポーネントおよびメソッドは次のとおりです。

①列を追加する。

コンポーネント	■テーブル格納変数
メソッド	指定位置に列を追加する(int, String, Class)
引数	第 1 引数：追加位置 第 2 引数：列名 第 3 引数：型
戻り値	なし
列の位置、列の名前、データ型を指定して、列を追加します。データ型は、文字列の場合「java.lang.String」、整数の場合「java.math.BigInteger」、実数の場合「java.math.BigDecimal」、日付の場合「java.util.Date」、論理値の場合「java.lang.Boolean」を設定します。	

②行について繰り返し処理を行う。

コンポーネント	■繰り返し処理(FOR)
メソッド	繰り返し処理を実行する(int, boolean, int, boolean, int)
引数	第 1 引数：開始値 第 2 引数：開始値を含めるかどうか 第 3 引数：終了値 第 4 引数：終了値を含めるかどうか 第 5 引数：増減値
戻り値	なし
テーブルデータの行について繰り返し処理を実行します。	

③「セル値を用いた計算」を実行する。(前章を参照)

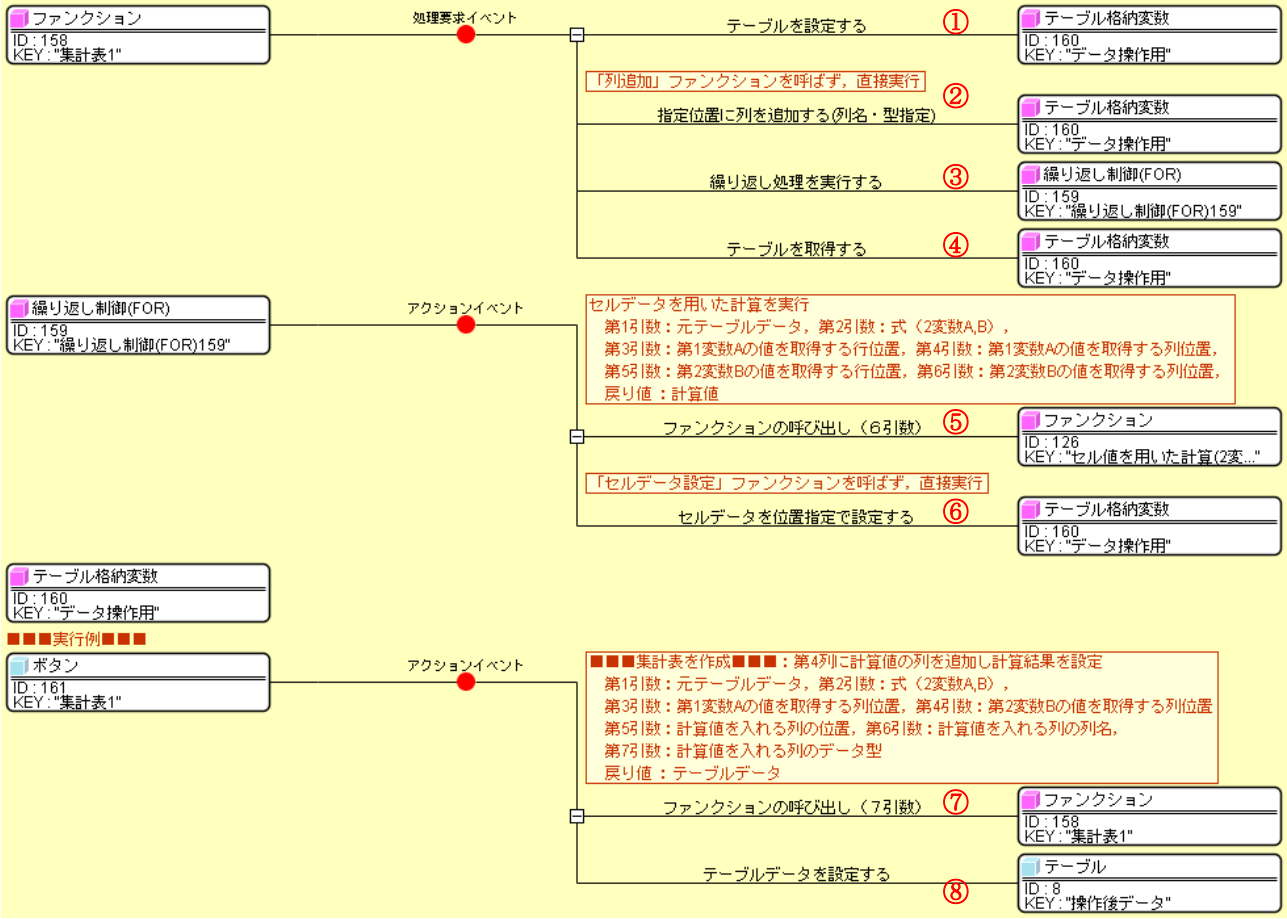
コンポーネント	■ファンクション（セル値を用いた計算）
メソッド	ファンクションの呼び出し(6 変数)(Object, Object, Object, Object, Object, Object)
引数	第 1 引数：元テーブルデータ 第 2 引数：式（2 変数 A,B） 第 3 引数：第 1 変数 A の値を取得する行位置 第 4 引数：第 1 変数 A の値を取得する列位置 第 5 引数：第 2 変数 B の値を取得する行位置 第 6 引数：第 2 変数 B の値を取得する列位置
戻り値	Object(計算結果、BigDecimal)
セル値の値を用いて計算を実行します。	

④計算値をセルに設定する。

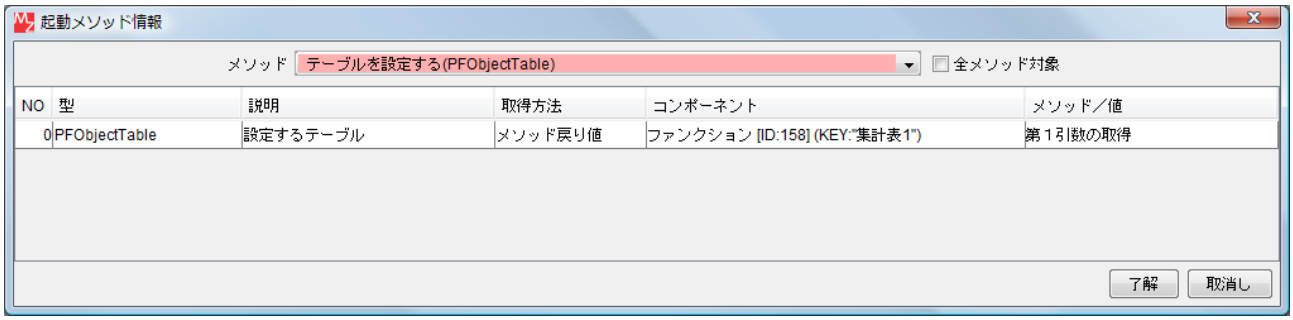
コンポーネント	■テーブル格納変数
メソッド	セルデータを位置指定で設定する(int, int, Object)
引数	第 1 引数：行位置 第 2 引数：列位置 第 3 引数：データ
戻り値	なし
行位置および列位置を指定してセルデータを設定します。	

接続例

集計表作成処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に計算式（2 変数 A,B)、第 3 引数に第 1 変数 A の値を取得する列位置、第 4 引数に第 2 変数 B の値を取得する列位置、第 5 引数に計算値を入れる列（追加する列）の位置、第 6 引数に計算値を入れる列の列名、第 7 引数に計算値を入れる列のデータ型を指定してメソッド「ファンクションの呼び出し（7 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。

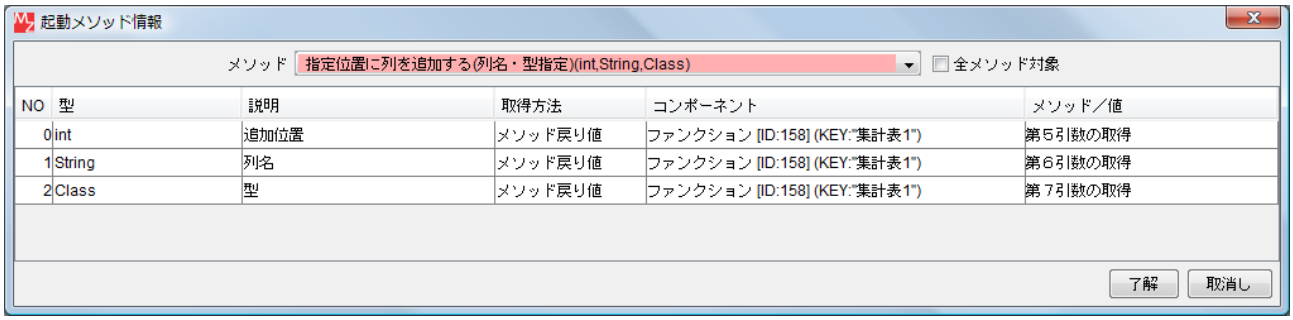


①の接続情報：処理対象のテーブルデータを設定する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	設定するテーブル	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第１引数の取得

②の接続情報：計算結果を入れる列を追加する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	追加位置	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第５引数の取得
1	String	列名	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第６引数の取得
2	Class	型	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第７引数の取得

③の接続情報：行について繰り返し処理を実行する。

起動メソッド情報

メソッド 繰り返し処理を実行する(int,boolean,int,boolean,int) 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	int	開始値	固定値	-	0
1	boolean	開始値を含めるかどうか	固定値	-	true
2	int	終了値	メソッド戻り値	テーブル格納変数 [ID:160] (KEY:"データ操作用")	行数を取得する
3	boolean	終了値を含めるかどうか	固定値	-	false
4	int	増減値	固定値	-	1

了解 取消し

④の接続情報：ファンクションの戻り値としてテーブルデータを返す。

起動メソッド情報

メソッド テーブルを取得する() 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

了解 取消し

⑤の接続情報：各行についてセル値を用いた計算を実行する。

起動メソッド情報

メソッド ファンクションの呼び出し（6引数）(Object,Object,Object,Object,Object,Object) 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	Object	第1引数	メソッド戻り値	テーブル格納変数 [ID:160] (KEY:"データ操作用")	テーブルを取得する
1	Object	第2引数	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第2引数の取得
2	Object	第3引数	メソッド戻り値	繰り返し制御(FOR) [ID:159] (KEY:"繰り返し制御(FOR...	現在値を取得する
3	Object	第4引数	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第3引数の取得
4	Object	第5引数	メソッド戻り値	繰り返し制御(FOR) [ID:159] (KEY:"繰り返し制御(FOR...	現在値を取得する
5	Object	第6引数	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第4引数の取得

了解 取消し

⑥の接続情報：計算結果を指定列に設定する。

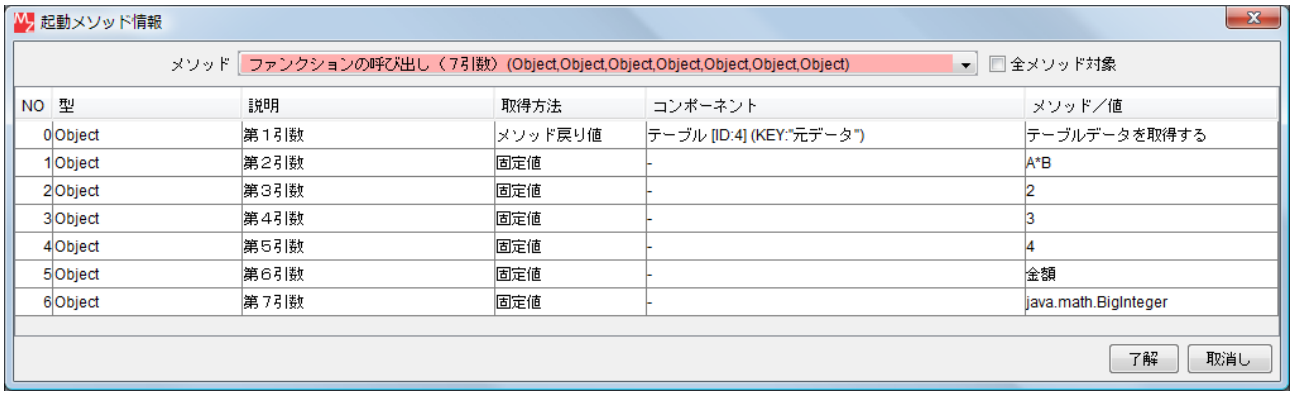
起動メソッド情報

メソッド セルデータを位置指定で設定する(int,int,Object) 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	int	指定行	メソッド戻り値	繰り返し制御(FOR) [ID:159] (KEY:"繰り返し制御(FOR...	現在値を取得する
1	int	指定列	メソッド戻り値	ファンクション [ID:158] (KEY:"集計表1")	第5引数の取得
2	Object	設定するデータ	メソッド処理結果	-	ファンクションの呼び出し（6引...

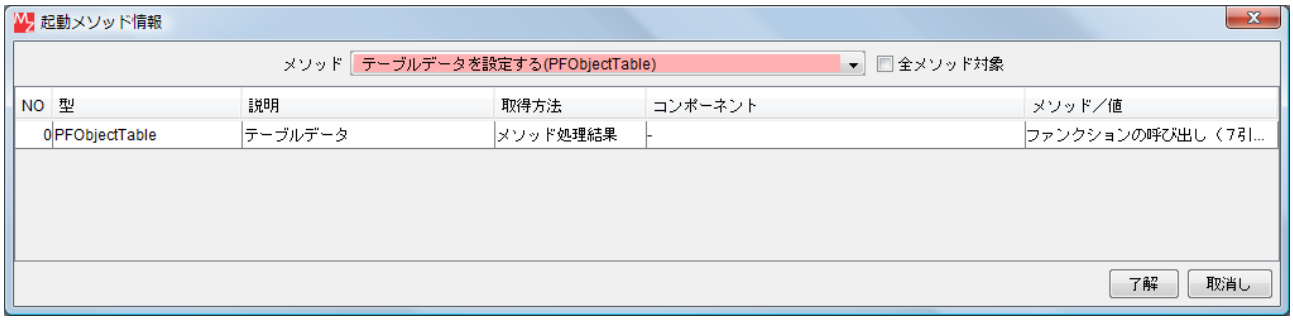
了解 取消し

⑦の接続情報：定義した集計表作成処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	固定値	-	A*B
2	Object	第3引数	固定値	-	2
3	Object	第4引数	固定値	-	3
4	Object	第5引数	固定値	-	4
5	Object	第6引数	固定値	-	金額
6	Object	第7引数	固定値	-	java.math.BigInteger

⑧の接続情報：結果をテーブルに設定する。

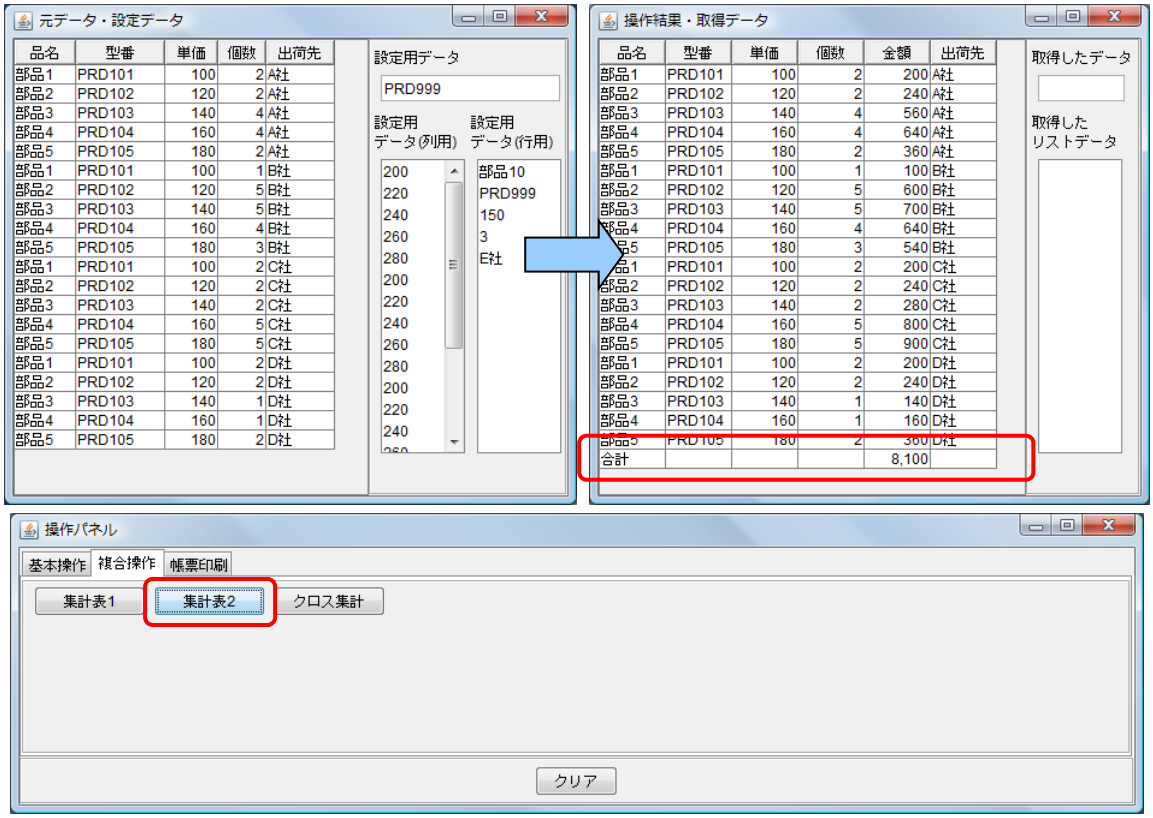


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（7引...

3.2 集計表２

動作イメージ

以下の図のように、前節の集計表１に行を追加し合計値を設定する処理手順について説明します。



使用メソッド

この集計表の生成処理は、前節の「集計表 1」、基本操作の章の「列単位の集計」、「行の追加」等を組み合わせた処理で実現します。ただし、行の追加については「行の追加」で定義した「ファンクション」コンポーネントを用いず、テーブル格納変数を用いて直接的に追加を行っています。集計表を作成するために使用する主なコンポーネントおよびメソッドは次のとおりです。

①集計表 1 を生成する。

コンポーネント	■ファンクション（集計表 1）
メソッド	ファンクションの呼び出し（7 変数）(Object, Object, Object, Object, Object, Object, Object)
引数	第 1 引数：元テーブルデータ 第 2 引数：式（2 変数 A,B） 第 3 引数：第 1 変数 A の値を取得する列位置 第 4 引数：第 2 変数 B の値を取得する列位置 第 5 引数：計算値を入れる列の位置 第 6 引数：計算値を入れる列の列名 第 7 引数：計算値を入れる列のデータ型
戻り値	Object(計算結果、PFObjectTable)
集計表 1 を生成します。	

②列単位の集計を実行する。

コンポーネント	■ファンクション（列単位の集計）
メソッド	ファンクションの呼び出し（2 変数）(Object, Object)
引数	第 1 引数：元テーブルデータ 第 2 引数：集計対象の列位置
戻り値	Object(集計結果、BigDecimal)
指定列のデータの集計値を計算します。	

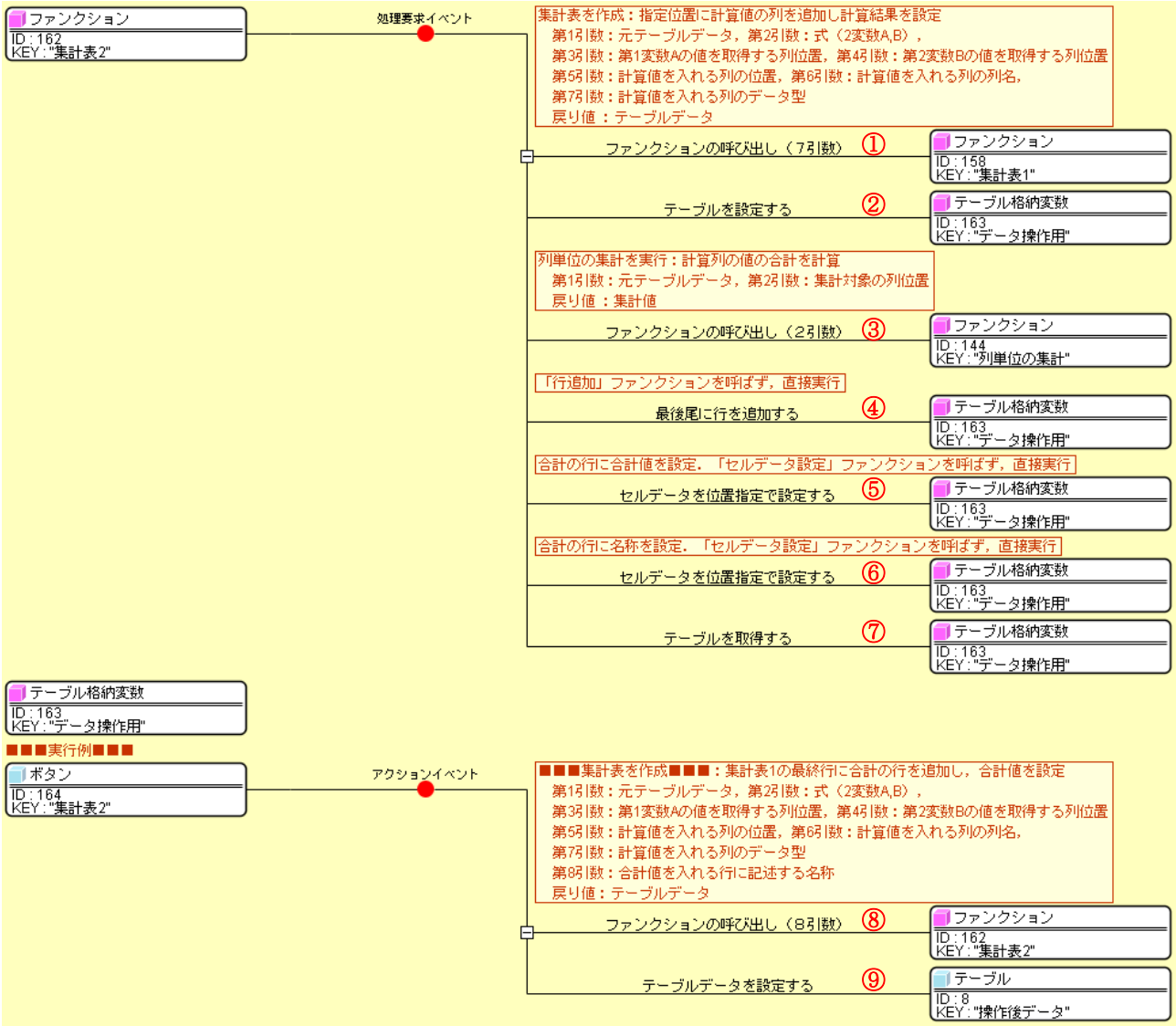
③行を追加し、合計値を設定する。

コンポーネント	■テーブル格納変数
メソッド	最終尾に行を追加する()
引数	なし
戻り値	なし
テーブルデータの最終尾に行（合計値の行）を追加します。	

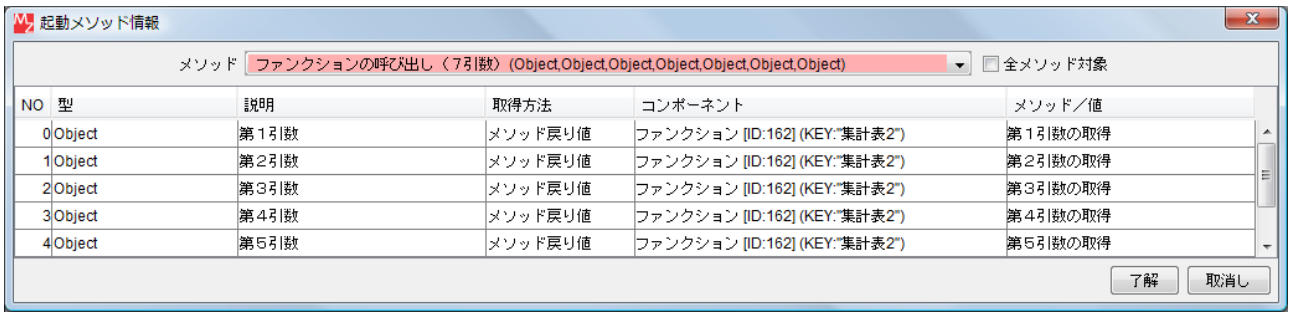
メソッド	セルデータを位置指定で設定する(int, int, Object)
引数	第 1 引数：行位置 第 2 引数：列位置 第 3 引数：データ
戻り値	なし
行位置および列位置を指定してセルデータを設定します。（合計値、「合計」の文字を設定）	

接続例

集計表作成処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に計算式（2 変数 A,B）、第 3 引数に第 1 変数 A の値を取得する列位置、第 4 引数に第 2 変数 B の値を取得する列位置、第 5 引数に計算値を入れる列の位置、第 6 引数に計算値を入れる列の列名、第 7 引数に計算値を入れる列のデータ型、第 8 引数に合計値を入れる行に記述する名称を指定してメソッド「ファンクションの呼び出し（8 引数）」を実行すると、結果の値が戻り値として得られるように定義しています。

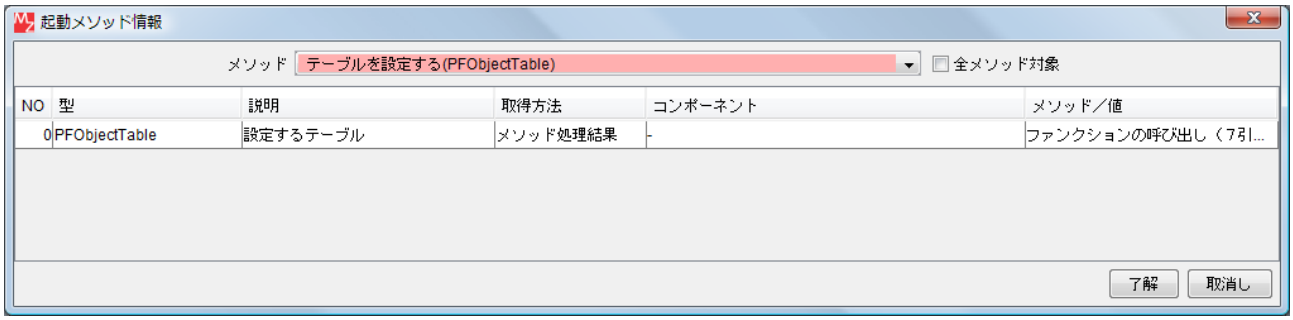


①の接続情報：集計表１を生成する。

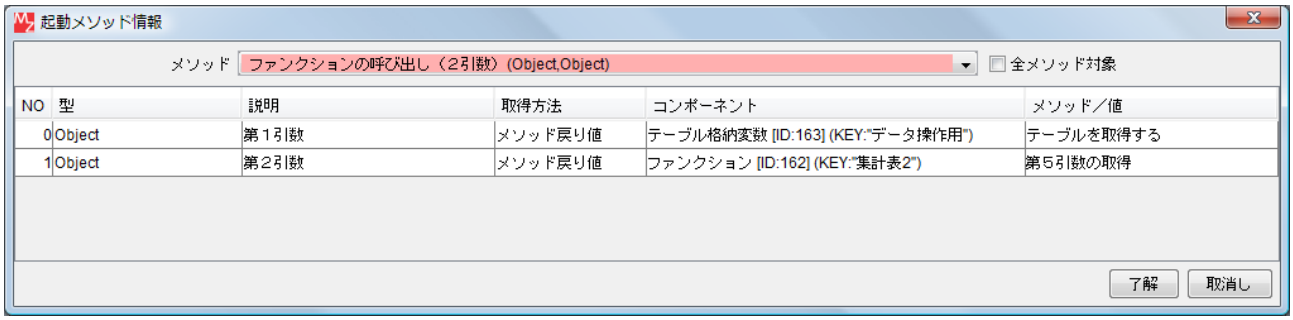


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第１引数	メソッド戻り値	ファンクション [ID:162] (KEY:"集計表2")	第１引数の取得
1	Object	第２引数	メソッド戻り値	ファンクション [ID:162] (KEY:"集計表2")	第２引数の取得
2	Object	第３引数	メソッド戻り値	ファンクション [ID:162] (KEY:"集計表2")	第３引数の取得
3	Object	第４引数	メソッド戻り値	ファンクション [ID:162] (KEY:"集計表2")	第４引数の取得
4	Object	第５引数	メソッド戻り値	ファンクション [ID:162] (KEY:"集計表2")	第５引数の取得

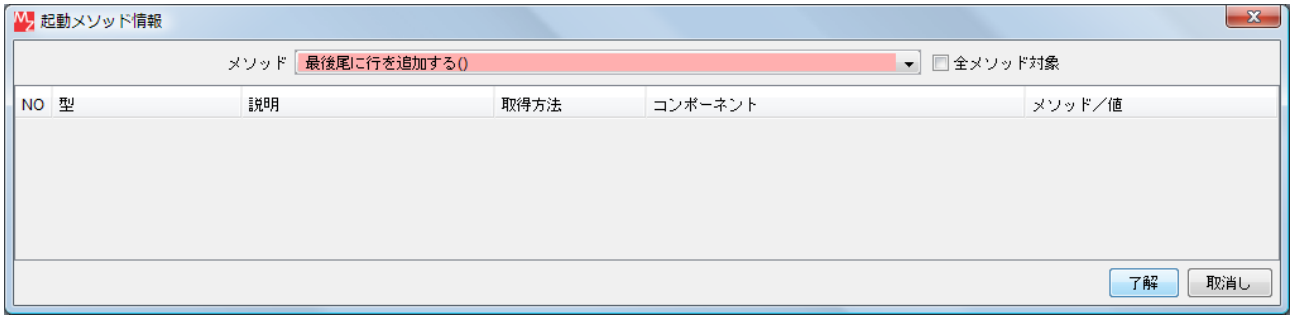
②の接続情報：集計表 1 を処理対象として設定する。



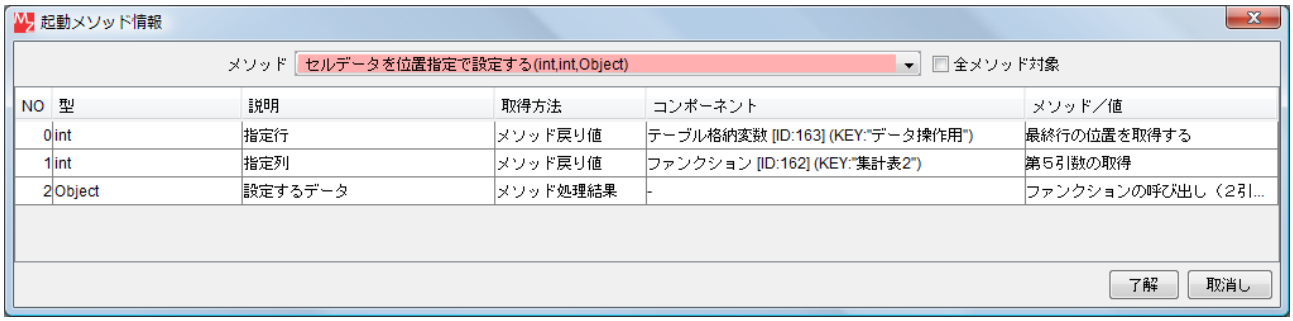
③の接続情報：集計対象列について集計を実行する。



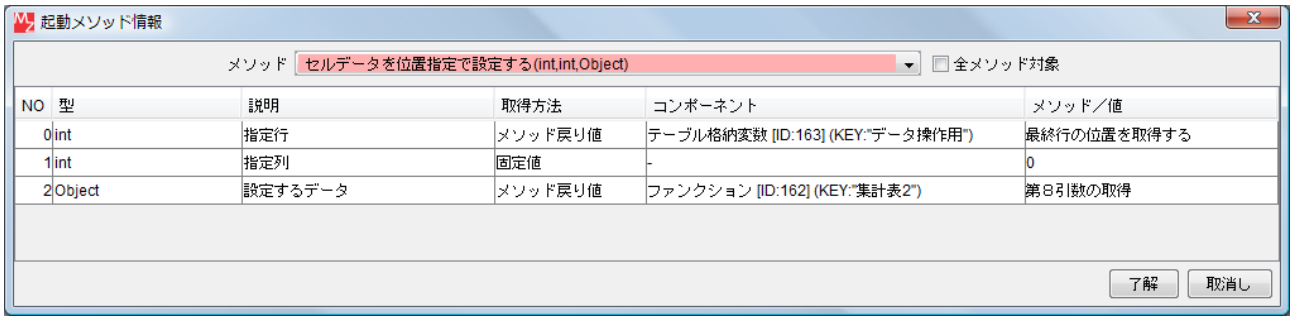
④の接続情報：合計値を設定する行を追加する。



⑤の接続情報：合計値を設定する。

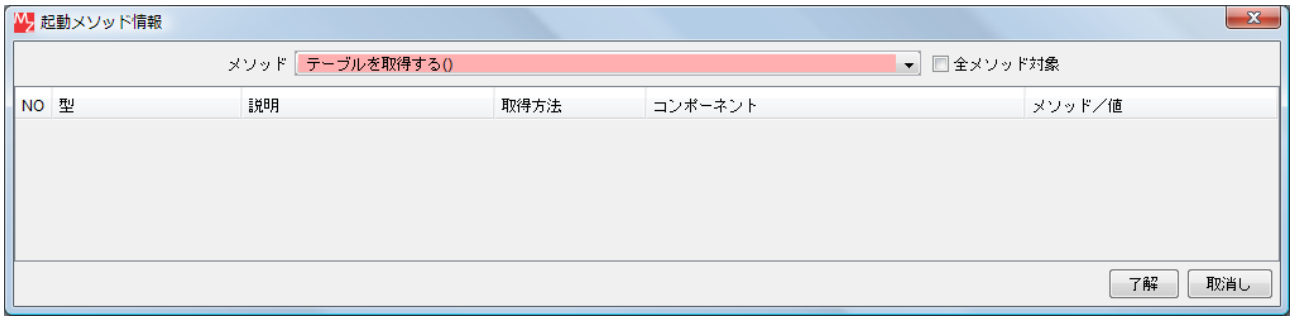


⑥の接続情報：合計値を意味する文字列を設定する。



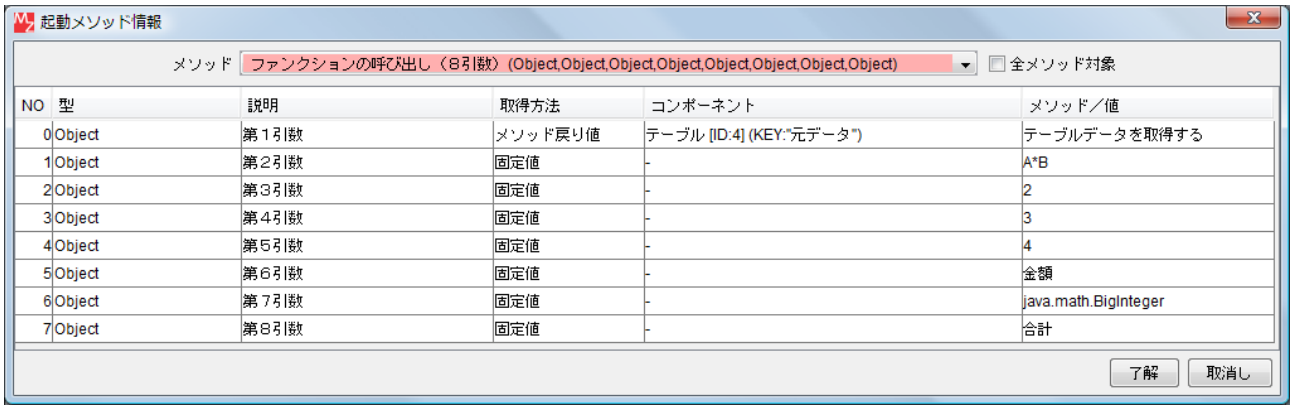
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	指定行	メソッド戻り値	テーブル格納変数 [ID:163] (KEY:"データ操作用")	最終行の位置を取得する
1	int	指定列	固定値	-	0
2	Object	設定するデータ	メソッド戻り値	ファンクション [ID:162] (KEY:"集計表2")	第8引数の取得

⑦の接続情報：ファンクションの戻り値としてテーブルデータを返す。



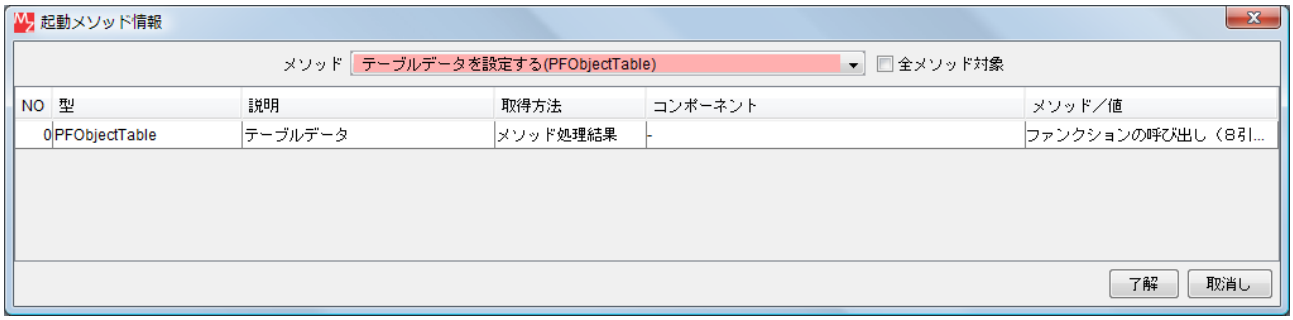
NO	型	説明	取得方法	コンポーネント	メソッド／値

⑧の接続情報：定義した集計表作成処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	固定値	-	A*B
2	Object	第3引数	固定値	-	2
3	Object	第4引数	固定値	-	3
4	Object	第5引数	固定値	-	4
5	Object	第6引数	固定値	-	金額
6	Object	第7引数	固定値	-	java.math.BigInteger
7	Object	第8引数	固定値	-	合計

⑨の接続情報：結果をテーブルに設定する。

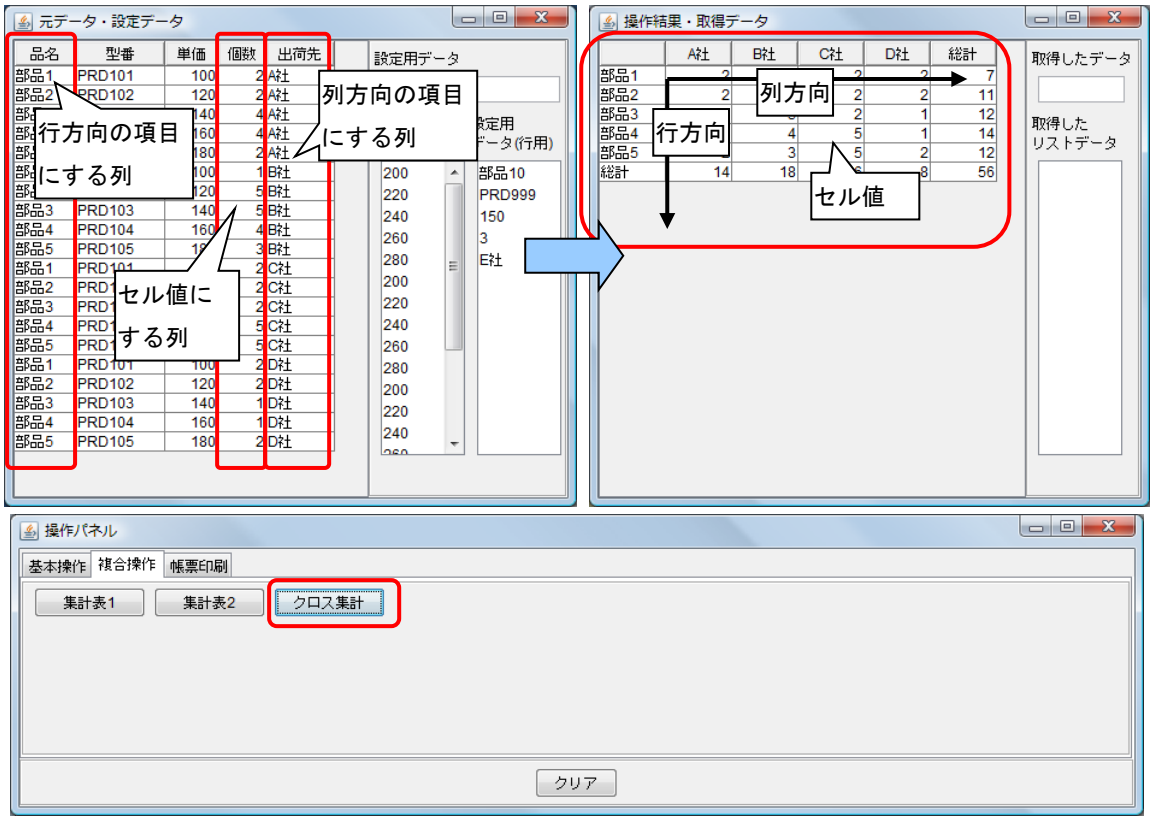


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	テーブルデータ	メソッド処理結果	-	ファンクションの呼び出し（8引...

3.3 クロス集計（ピボットテーブル生成）

動作イメージ

以下の図のように、テーブルデータから列指定（行方向の項目にする列、列方向の項目にする列、セル値にする列）の情報に基づいて分類テーブルを生成し、各行・各列について集計する（クロス集計、ピボットテーブル生成）処理手順について説明します。



使用メソッド

このクロス集計処理は、前章の「分類テーブルの作成」、「列単位の集計」、「行単位の集計」等を組み合わせた処理で実現します。ただし、集計値を入れる行および列の追加については「行の追加」「列の追加」で定義した「ファンクション」コンポーネントを用いず、テーブル格納変数を用いて直接的に行っています。集計表を作成するために使用する主なコンポーネントおよびメソッドは次のとおりです。

①分類テーブルを生成する。

コンポーネント	■ファンクション（分類テーブル作成）
メソッド	ファンクションの呼び出し（4 変数）(Object, Object, Object, Object)
引数	第 1 引数：元テーブルデータ 第 2 引数：行方向の列名または列位置, 第 3 引数：列方向の列名または列位置 第 4 引数：セル値(集計対象)の列名または列位置
戻り値	Object(PFObjectTable)
分類テーブルを生成します。	

②列単位の集計を実行する。

コンポーネント	■ファンクション（列単位の集計）
メソッド	ファンクションの呼び出し（2 変数）(Object, Object)
引数	第 1 引数：元テーブルデータ 第 2 引数：集計対象の列位置
戻り値	Object(集計結果、BigDecimal)
指定列のデータの集計値を計算します。	

③行単位の集計を実行する。

コンポーネント	■ファンクション（行単位の集計）
メソッド	ファンクションの呼び出し（2 変数）(Object, Object)
引数	第 1 引数：元テーブルデータ 第 2 引数：集計対象の行位置
戻り値	Object(集計結果、BigDecimal)
指定行のデータの集計値を計算します。	

④行を追加し、集計値を設定する。

コンポーネント	■テーブル格納変数
メソッド	最終尾に行を追加する(データリスト指定)(PFObjectList)
引数	第 1 引数：データリスト
戻り値	なし
テーブルデータの最終尾に行（集計値の行）を追加します。	

⑤列を追加し、集計値を設定する。

メソッド	最終尾に列を追加する(列名・型・データリスト指定)(String, Class, PFObjectList)
引数	第 1 引数：列名 第 2 引数：型 第 3 引数：データリスト
戻り値	なし
列の名前、データ型、列データを指定して、最終列に列を追加します。	

⑥リストに要素を追加し、各行・各列の集計値を設定する。

コンポーネント	■リスト格納変数
メソッド	最終尾に要素を追加する(Object)
引数	第 1 引数：オブジェクト（集計値）
戻り値	なし
リストの最終尾に要素（集計値）を追加します。	

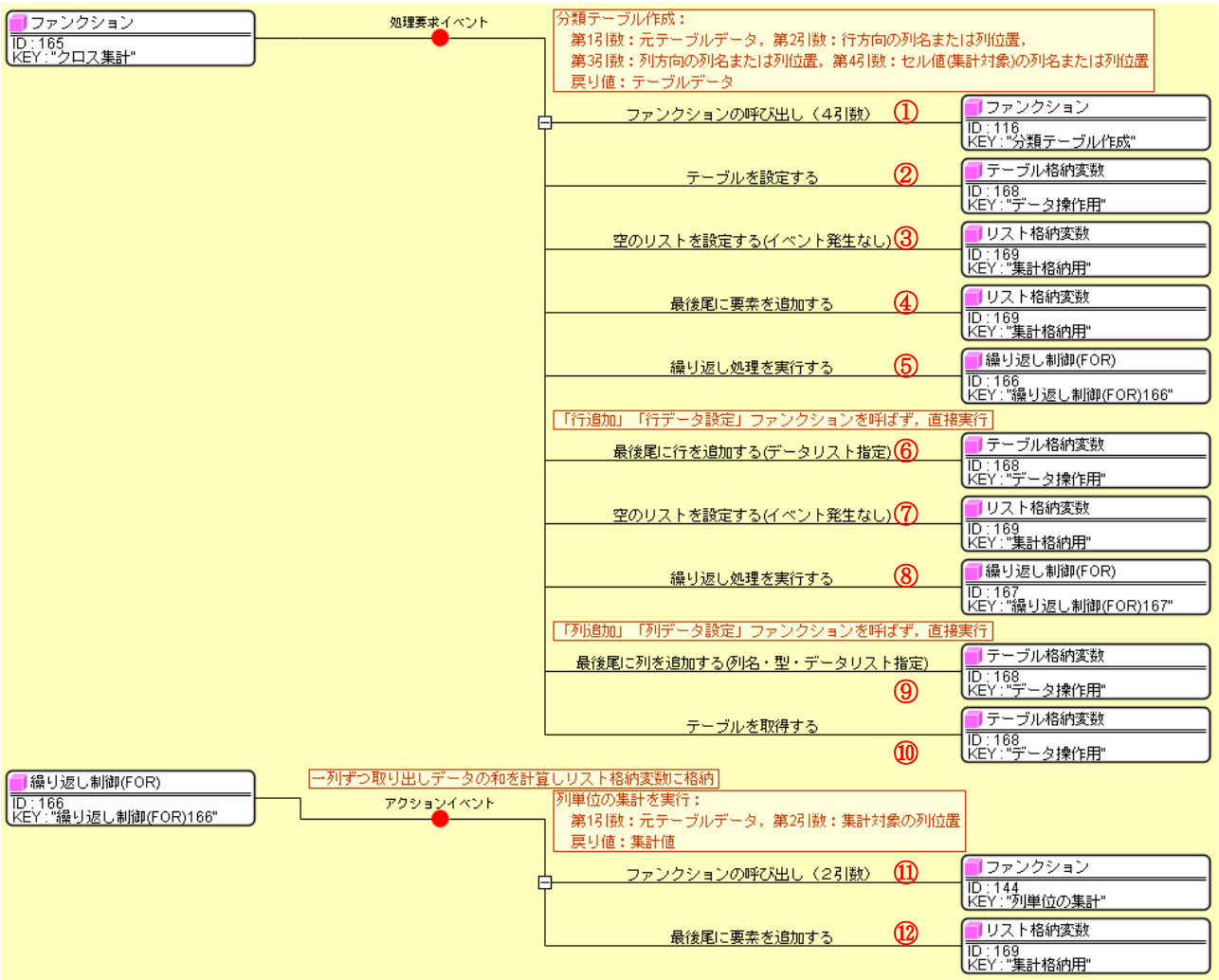
メソッド	空のリストを設定する(イベント発生なし)()
引数	なし
戻り値	なし
リストを初期化します。	

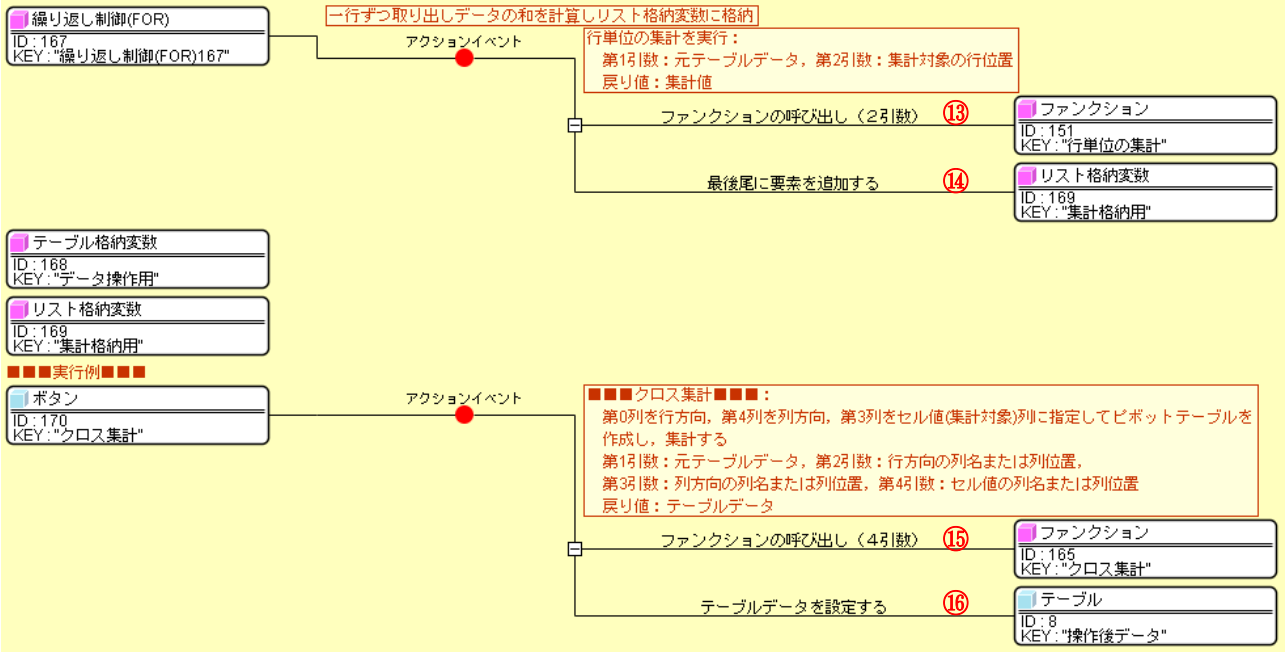
⑦行および列について繰り返し処理を行う。

コンポーネント	■繰り返し処理(FOR)
メソッド	繰り返し処理を実行する(int, boolean, int, boolean, int)
引数	第 1 引数：開始値 第 2 引数：開始値を含めるかどうか 第 3 引数：終了値 第 4 引数：終了値を含めるかどうか 第 5 引数：増減値
戻り値	なし
テーブルデータの行および列について繰り返し処理を実行します。	

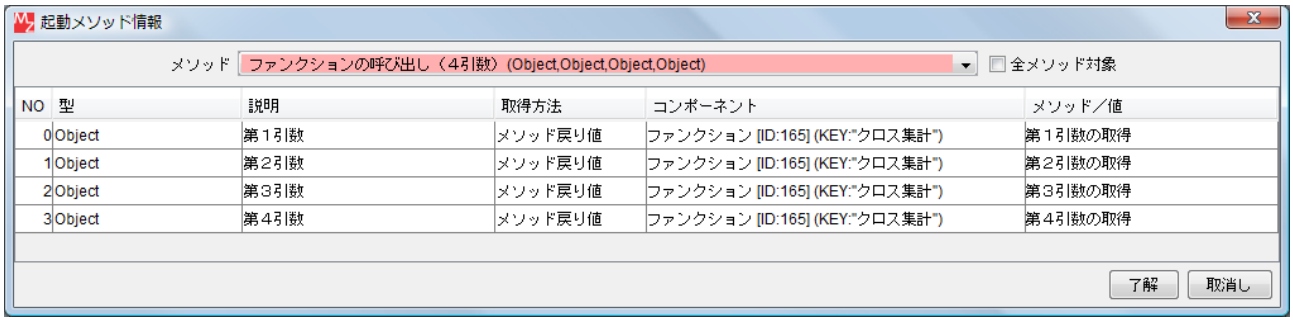
接続例

クロス集計処理は「ファンクション」コンポーネントの処理要求イベントに定義しています。第 1 引数に元テーブルデータ、第 2 引数に行方向の列名または列位置、第 3 引数に列方向の列名または列位置、第 4 引数にセル値の列名または列位置を指定してメソッド「ファンクションの呼び出し（4 引数)」を実行すると、結果の値が戻り値として得られるように定義しています。



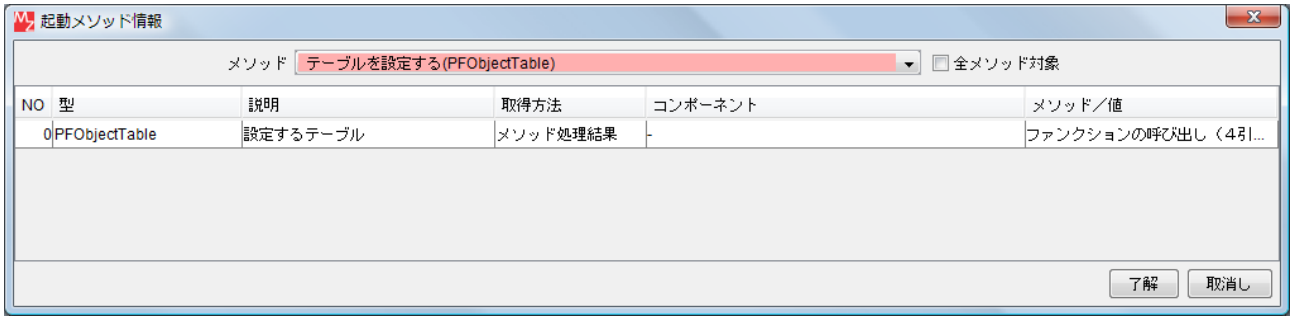


①の接続情報：分類テーブルを生成する。



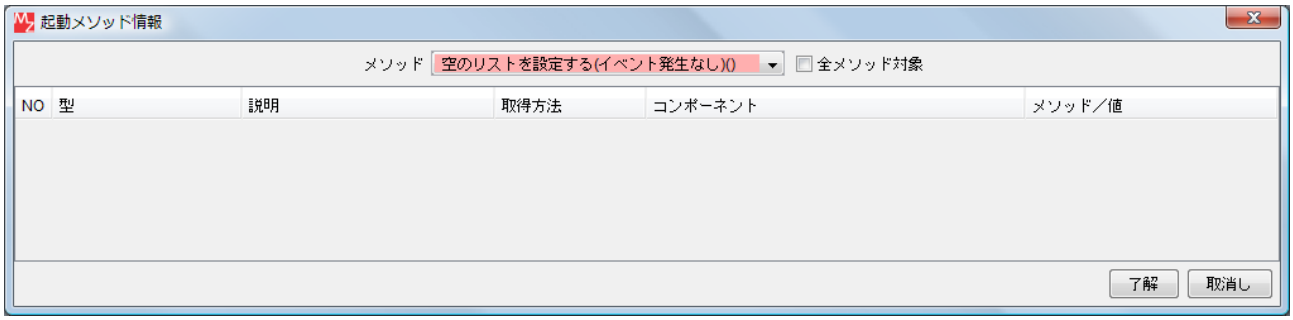
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第１引数	メソッド戻り値	ファンクション [ID:165] (KEY:"クロス集計")	第１引数の取得
1	Object	第２引数	メソッド戻り値	ファンクション [ID:165] (KEY:"クロス集計")	第２引数の取得
2	Object	第３引数	メソッド戻り値	ファンクション [ID:165] (KEY:"クロス集計")	第３引数の取得
3	Object	第４引数	メソッド戻り値	ファンクション [ID:165] (KEY:"クロス集計")	第４引数の取得

②の接続情報：分類テーブルを処理対象として設定する。

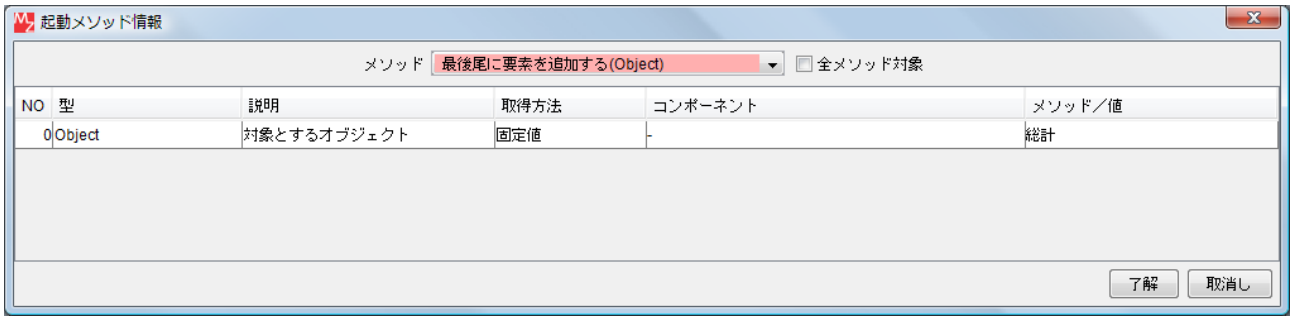


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	PFObjectTable	設定するテーブル	メソッド処理結果	-	ファンクションの呼び出し（4引...

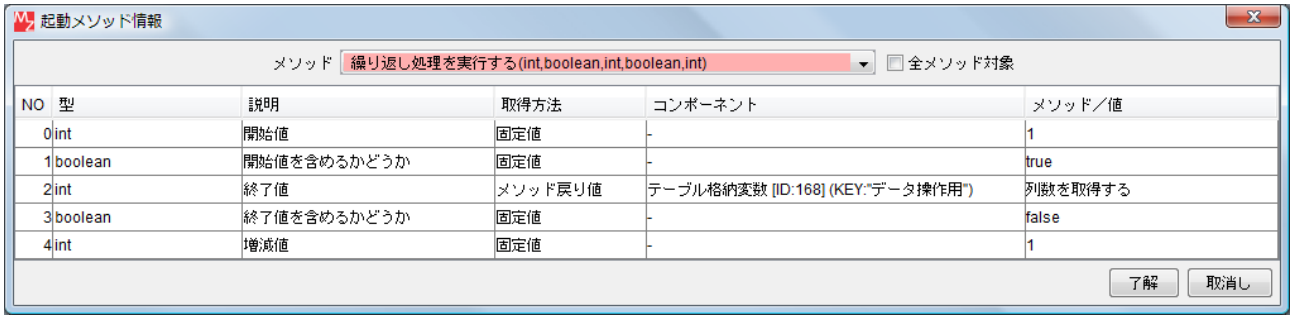
③の接続情報：集計値を格納するリストを初期化する。



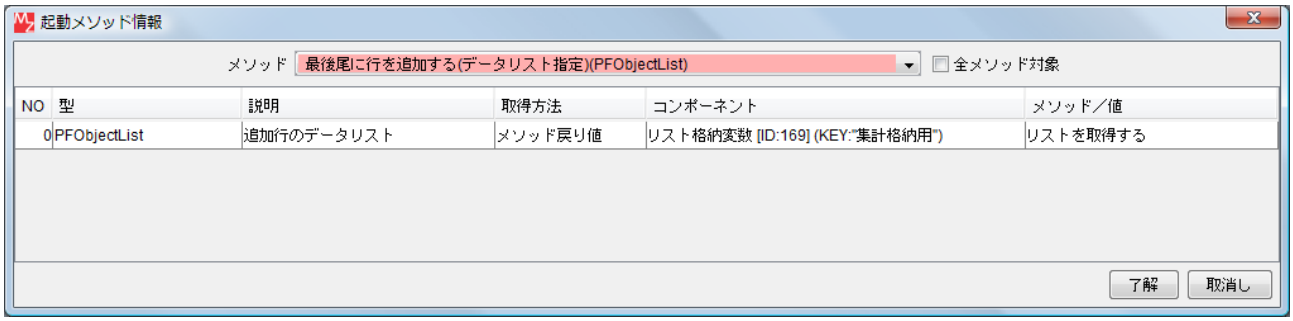
④の接続情報：集計行の名称を設定する。



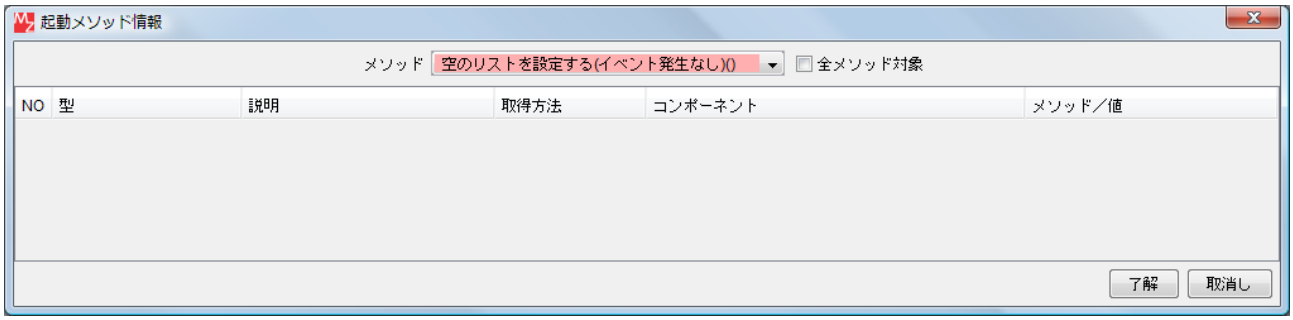
⑤の接続情報：各列について繰り返し処理を実行する。



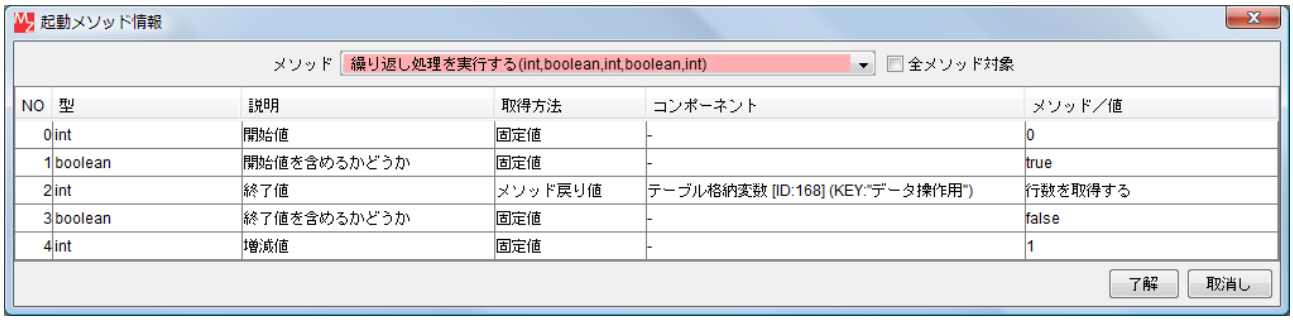
⑥の接続情報：各列の集計値を最終行に設定する。



⑦の接続情報：集計値を格納するリストを初期化する。

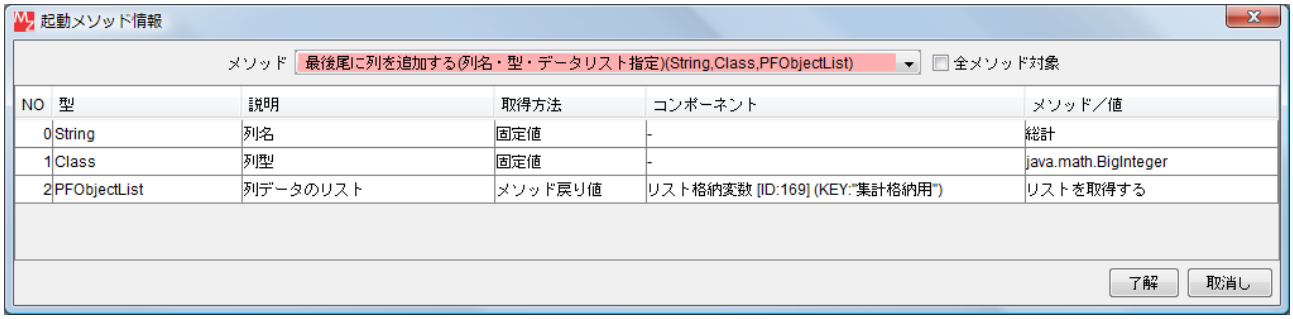


⑧の接続情報：各行について繰り返し処理を実行する。



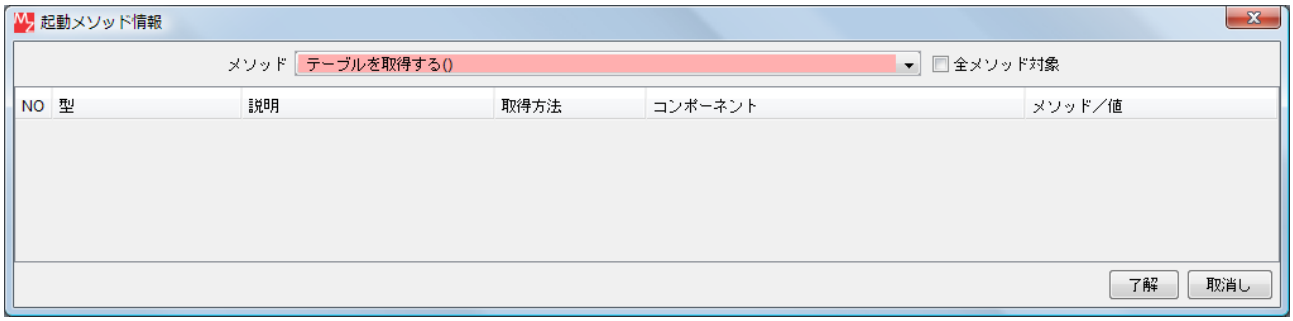
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	int	開始値	固定値	-	0
1	boolean	開始値を含めるかどうか	固定値	-	true
2	int	終了値	メソッド戻り値	テーブル格納変数 [ID:168] (KEY:"データ操作用")	行数を取得する
3	boolean	終了値を含めるかどうか	固定値	-	false
4	int	増減値	固定値	-	1

⑨の接続情報：各行の集計値を最終列に設定する。

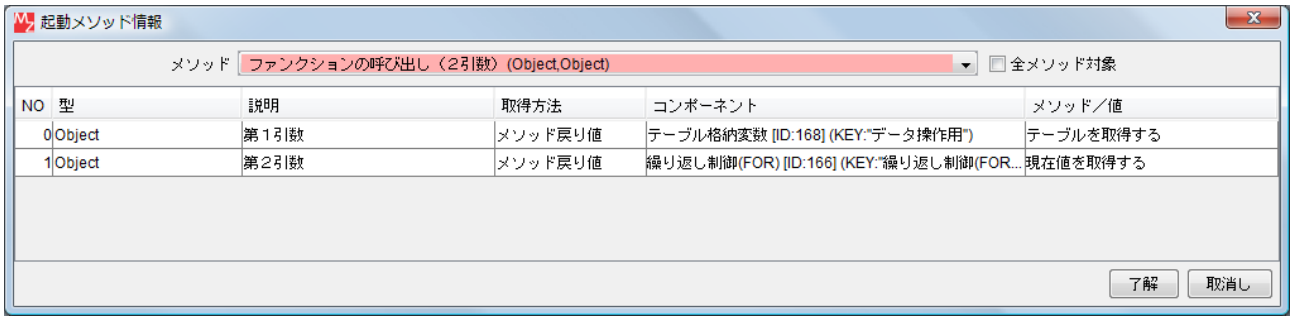


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	String	列名	固定値	-	総計
1	Class	列型	固定値	-	java.math.BigInteger
2	PFObjectList	列データのリスト	メソッド戻り値	リスト格納変数 [ID:169] (KEY:"集計格納用")	リストを取得する

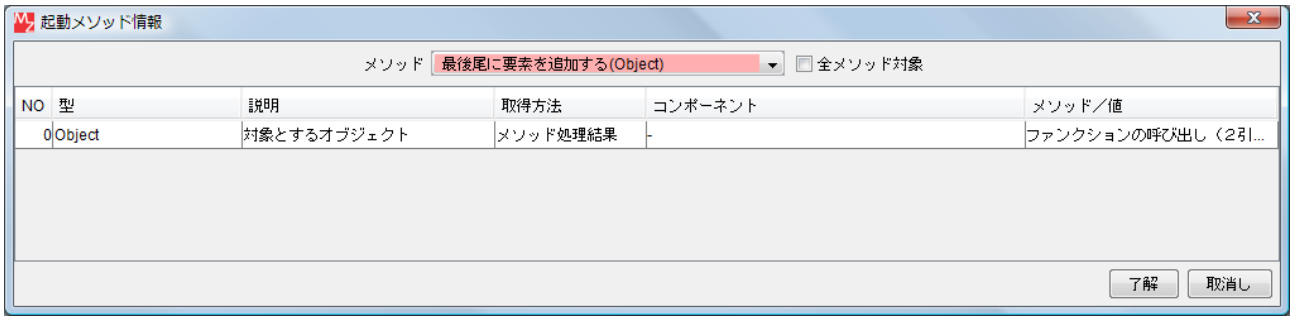
⑩の接続情報：ファンクションの戻り値としてテーブルデータを返す。



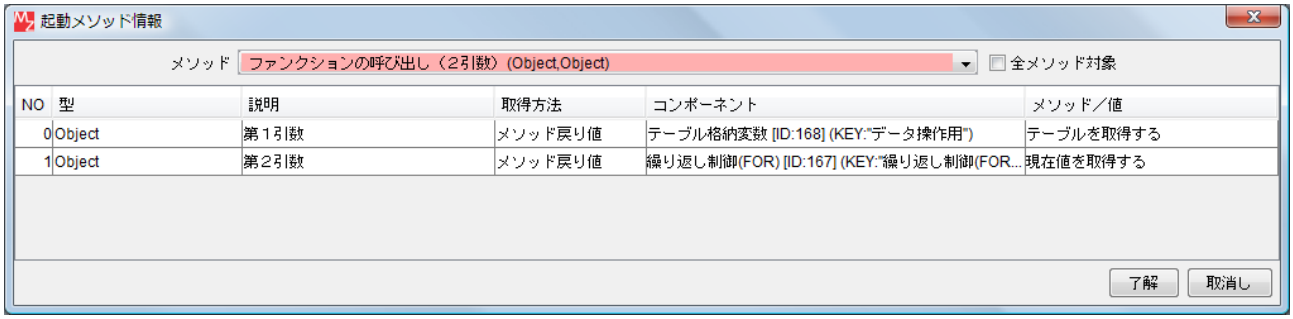
⑪の接続情報：列単位の集計を実行する。



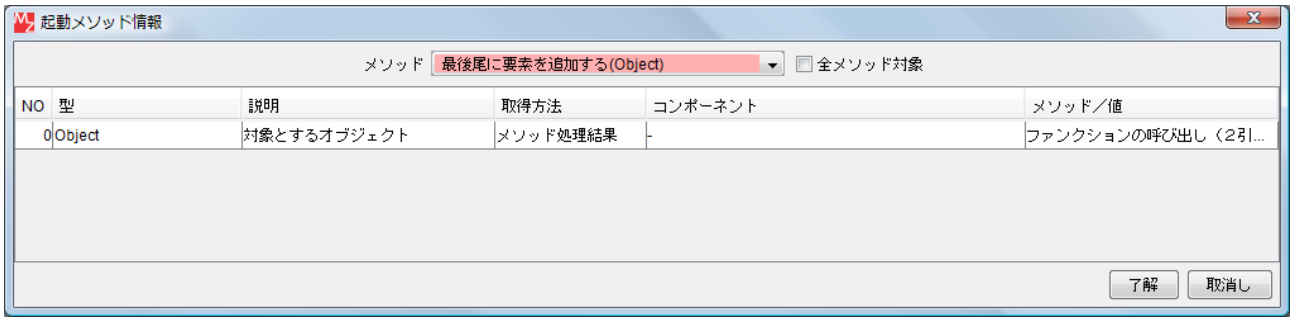
⑫の接続情報：各列の集計値をリストに追加する。



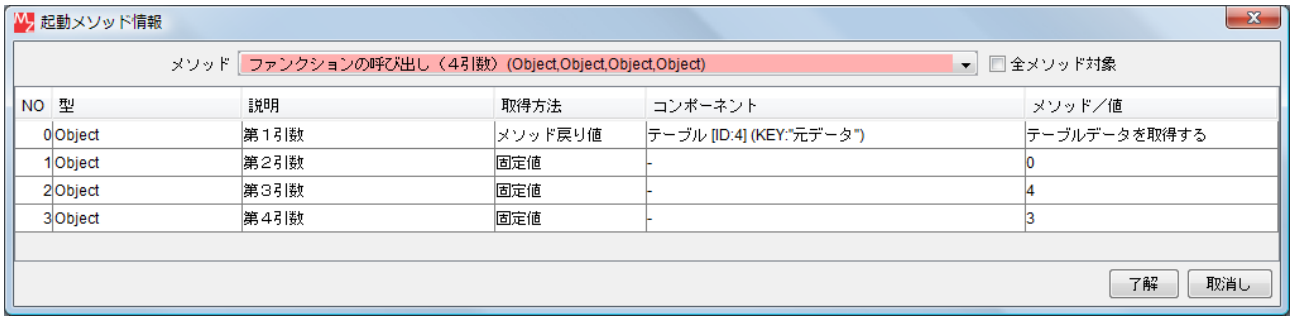
⑬の接続情報：行単位の集計を実行する。



⑭の接続情報：各行の集計値をリストに追加する。

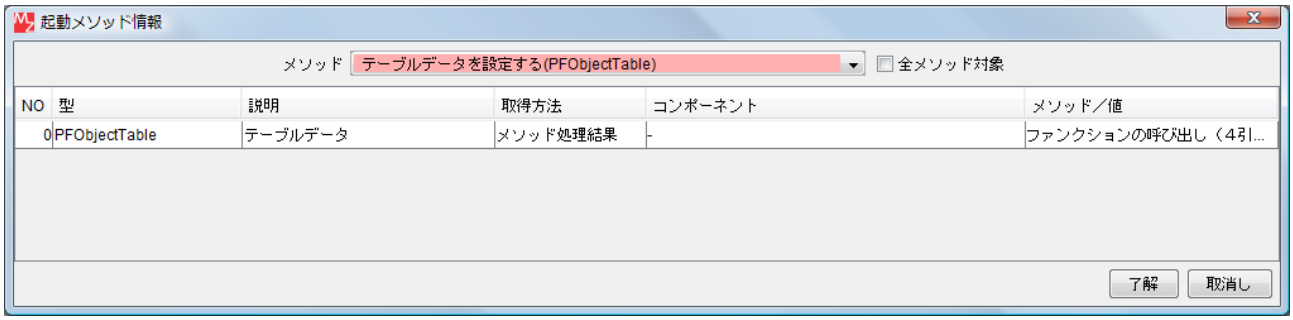


⑮の接続情報：定義したクロス集計処理を実行する。



NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	固定値	-	0
2	Object	第3引数	固定値	-	4
3	Object	第4引数	固定値	-	3

⑯の接続情報：結果をテーブルに設定する。



NO	型	説明	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	メソッド処理結果	-	ファンクションの呼び出し（4引...

第4章 帳票印刷

帳票機能とはデータの印刷機能です。帳票機能を使用するためには帳票コンポーネントを用います。本章では帳票コンポーネントを用いたテーブルデータの印刷について説明します。特に後半ではテーブルデータの繰り返し印刷機能（流し込み印刷機能）について設定方法を説明します。帳票についての基本的事項は「サンプル（帳票)」「注文票アプリケーション作成マニュアル」等をご覧ください。

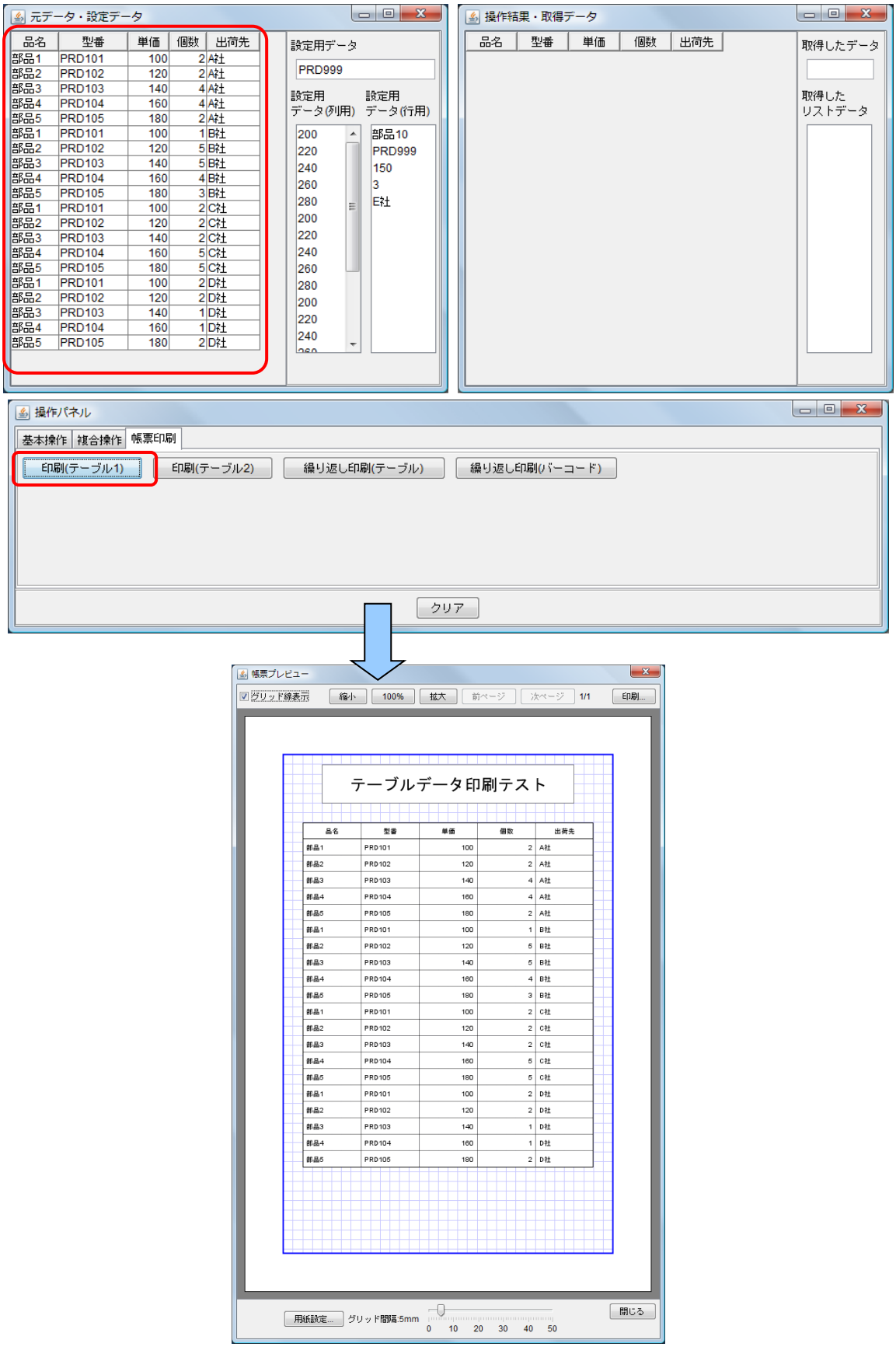
本章で使用する主なコンポーネントは次のとおりです。

コンポーネント名	場所	説明
■帳票	*[コンポーネント追加]*－［入出力］－［帳票］－［帳票］	データを印刷するためのコンポーネント。本章では特にテーブルデータの繰り返し印刷（流し込み印刷）について説明します。

4.1 テーブルデータの印刷

動作イメージ

以下の図のように、テーブルデータを印刷するための設定について説明します。

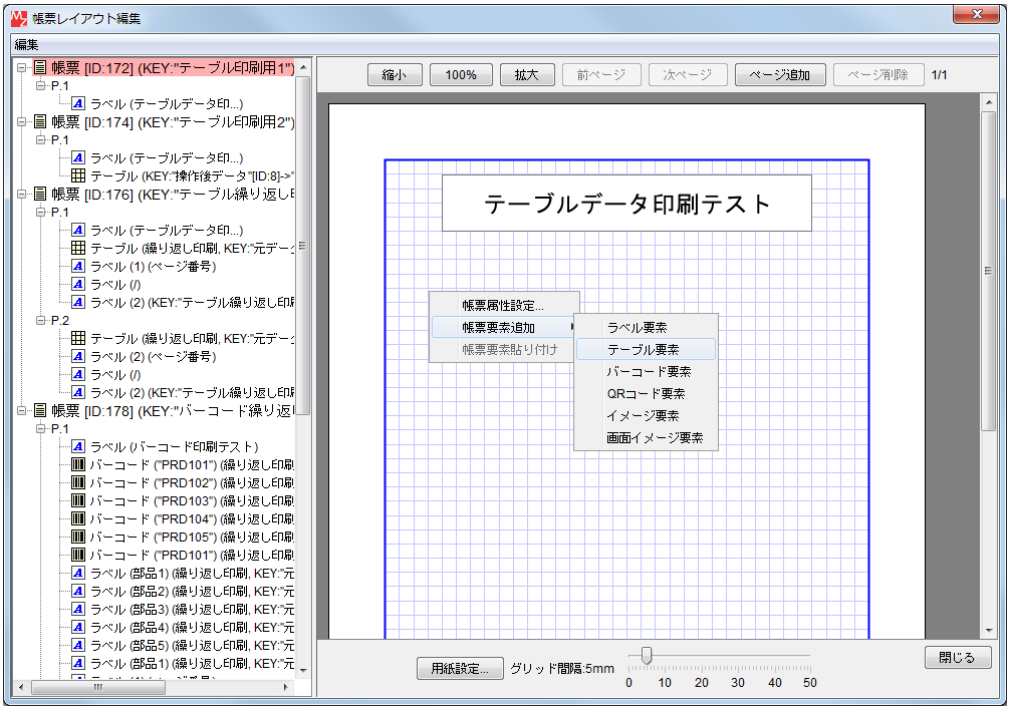


設定手順

テーブルデータを印刷するためには、帳票編集画面でテーブル要素を追加し、データ取得するコンポーネントおよびメソッドを選択します。

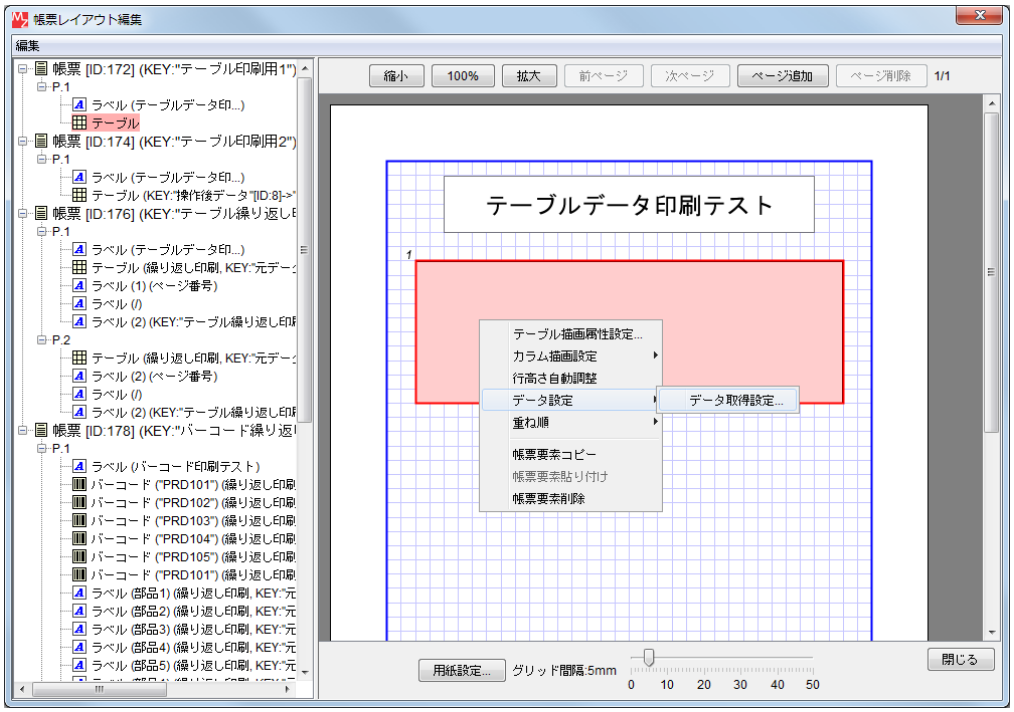
①テーブル要素の追加

帳票編集画面でマウス右クリックし、[帳票要素追加]-[テーブル要素]を選択します。

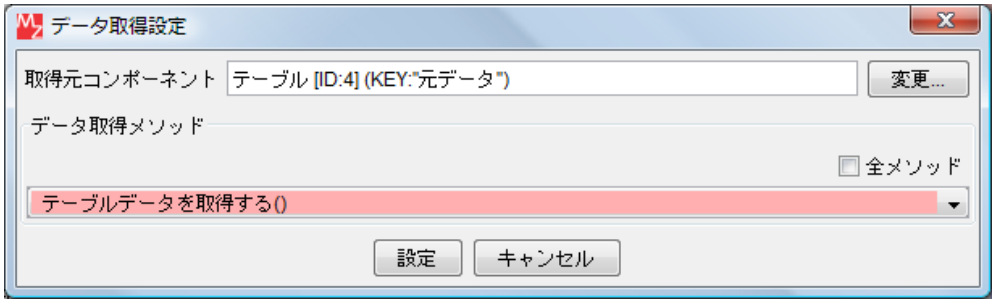


②データ取得設定

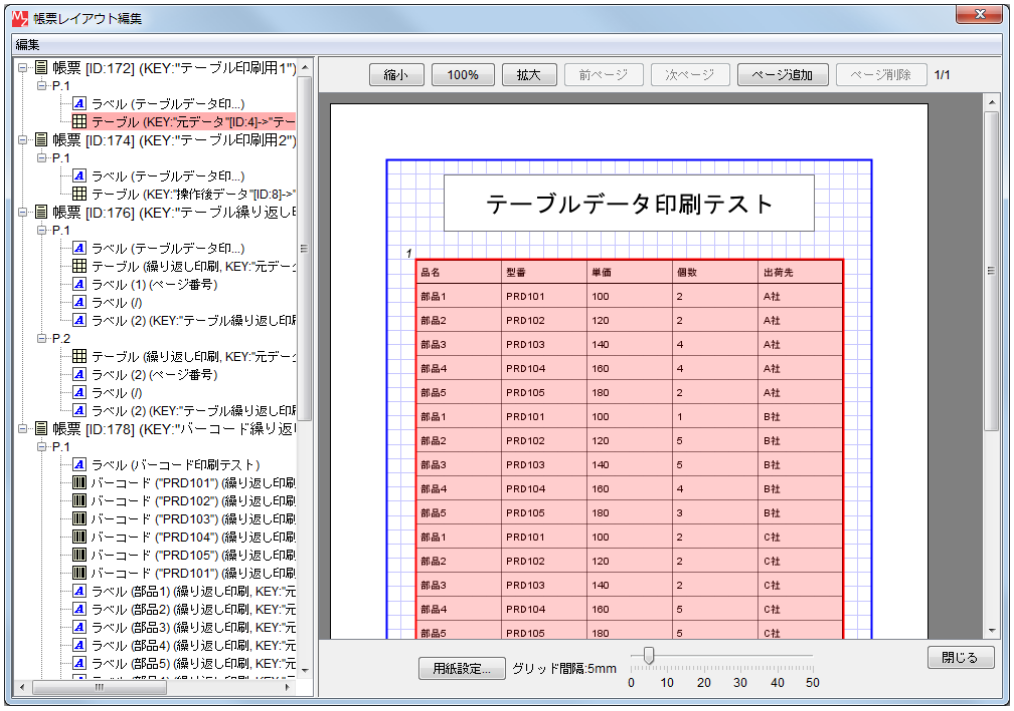
追加されたテーブル要素上でマウス右クリックし、[データ設定]-[データ取得設定...]を選択します。



表示された「データ取得設定」ダイアログで、データを取得するコンポーネントとメソッドを設定します。



取得先のテーブルにデータが設定されている場合、以下のように表示されます。

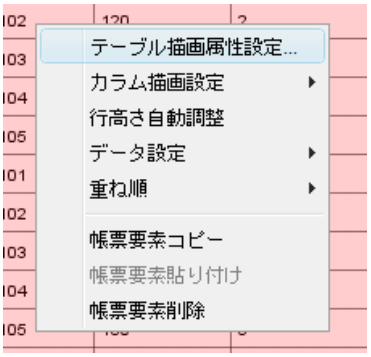


③描画設定

テーブルの描画設定は、テーブルデータ全体にかかわる設定と各列の設定が可能です。

・テーブル全体の描画設定

テーブル要素上でマウス右クリックし、[テーブル描画属性設定...]を選択します。

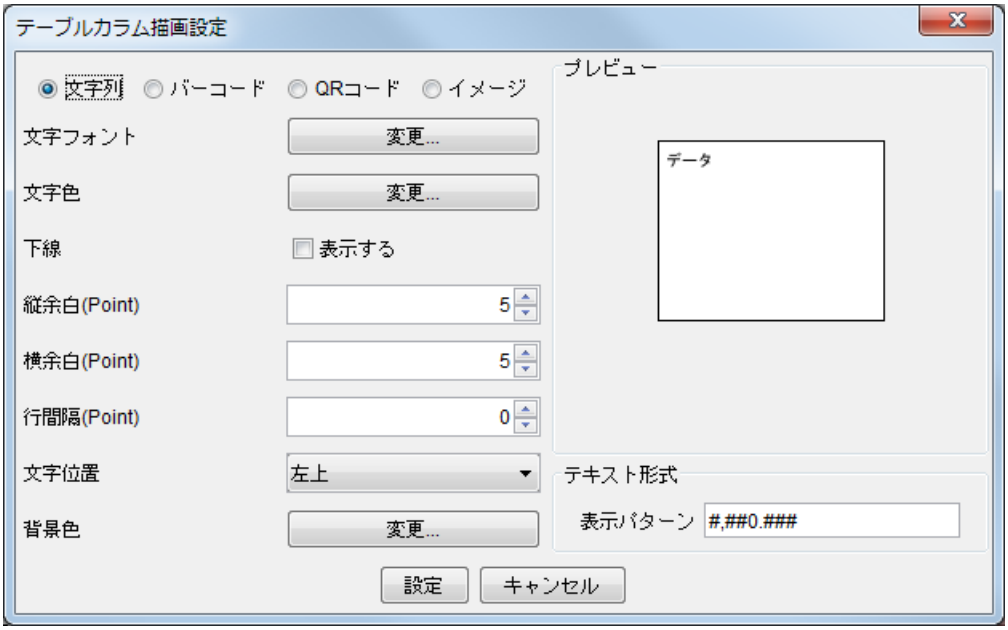


表示された「テーブル描画設定」ダイアログで、各種設定を行います。

・各列の描画設定

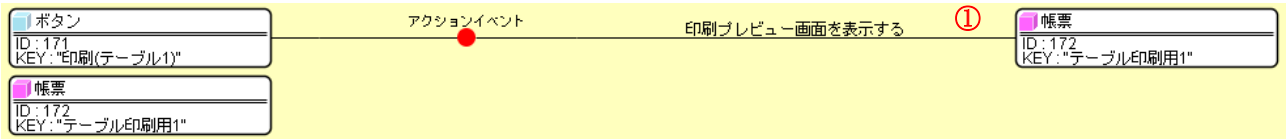
テーブル要素上でマウス右クリックし、[カラム描画設定]-[*列名*]-[描画属性設定...]を選択します。

表示された「テーブルカラム描画設定」ダイアログで、各種設定を行います。

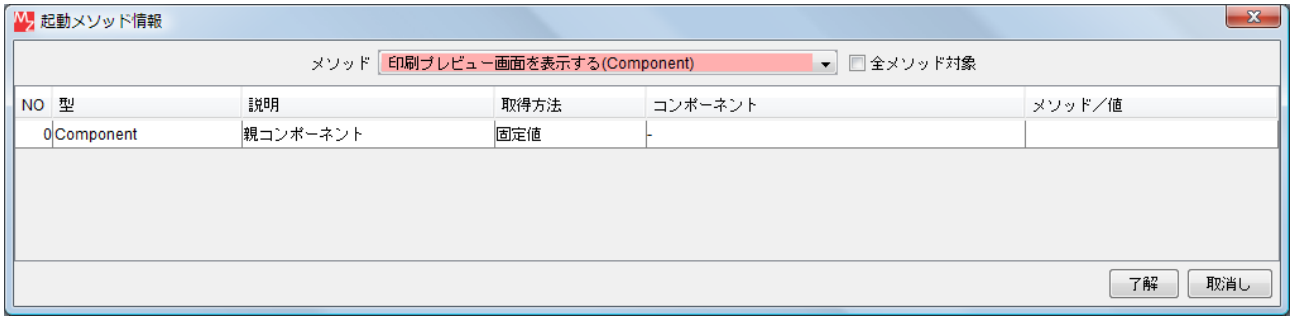


接続例

帳票の印刷は帳票コンポーネントのメソッドを実行することにより行います。



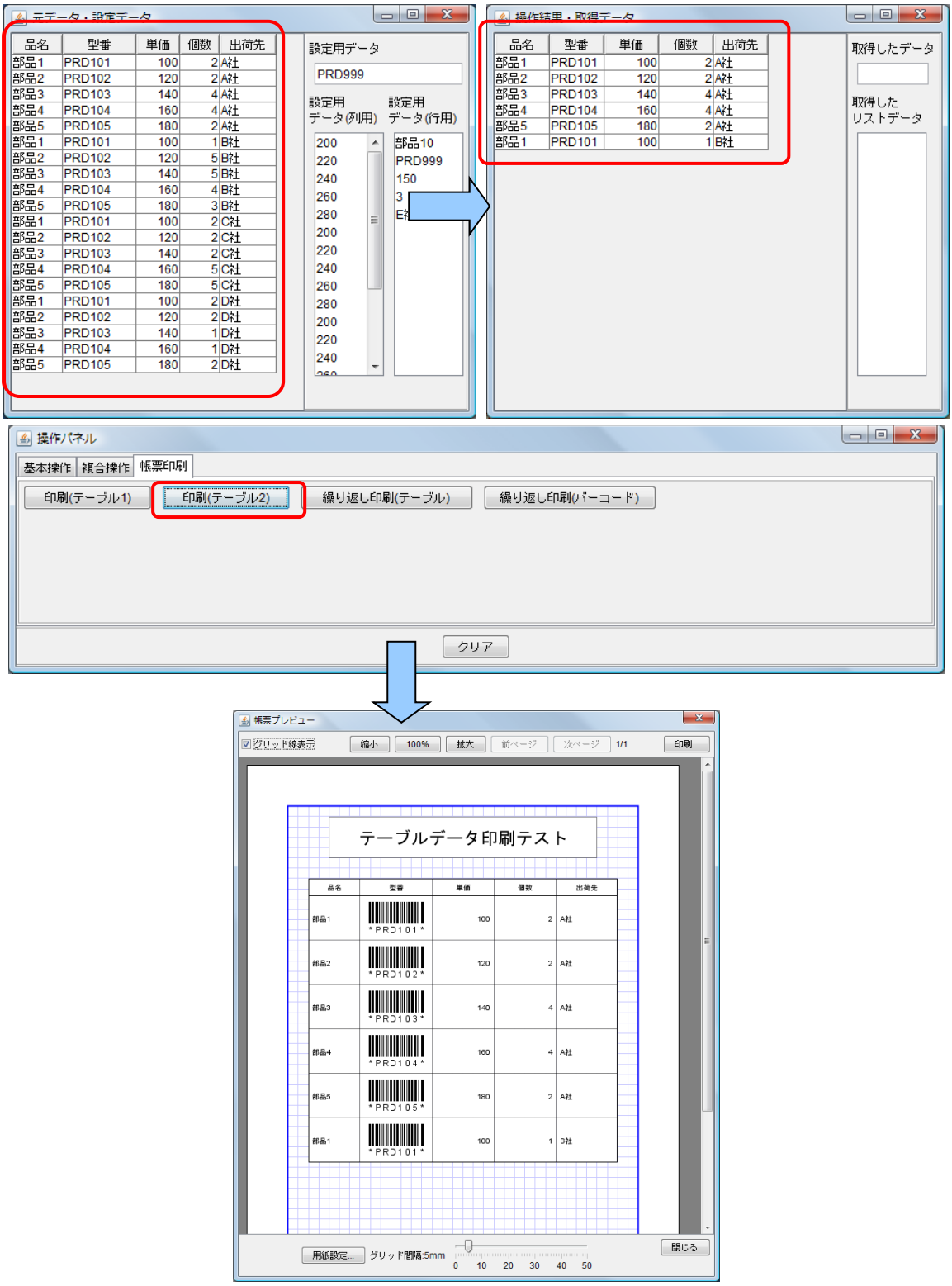
①の接続情報：印刷を実行する。



4.2 テーブル内のバーコード・ＱＲコード・イメージ印刷

動作イメージ

以下の図のように、テーブルデータ内にバーコード、QR コード[1]、または、イメージを描画し印刷するための設定について説明します。



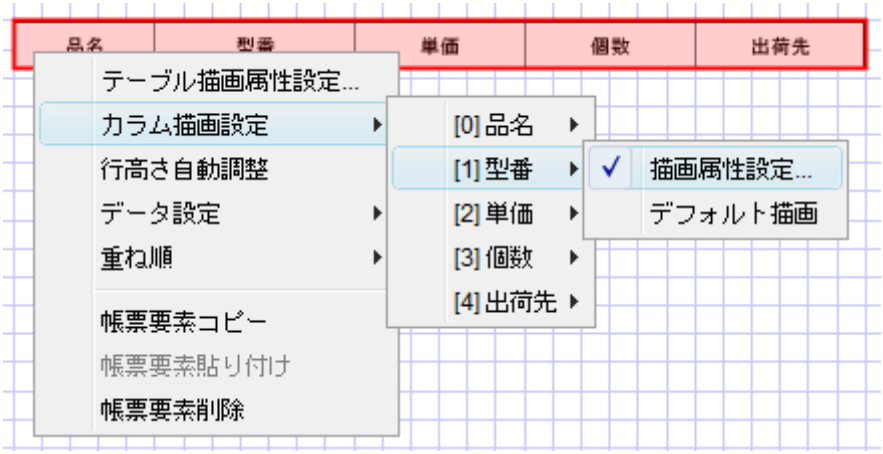
[1] QR コードは(株)デンソーウェーブの登録商標です。

設定手順

テーブルデータにバーコード、QR コード、または、イメージを描画し印刷するためには、「4.1 テーブルデータの印刷」の「設定手順」-「③描画設定」-「各列の描画設定」において、以下のように設定します。

・各列の描画設定

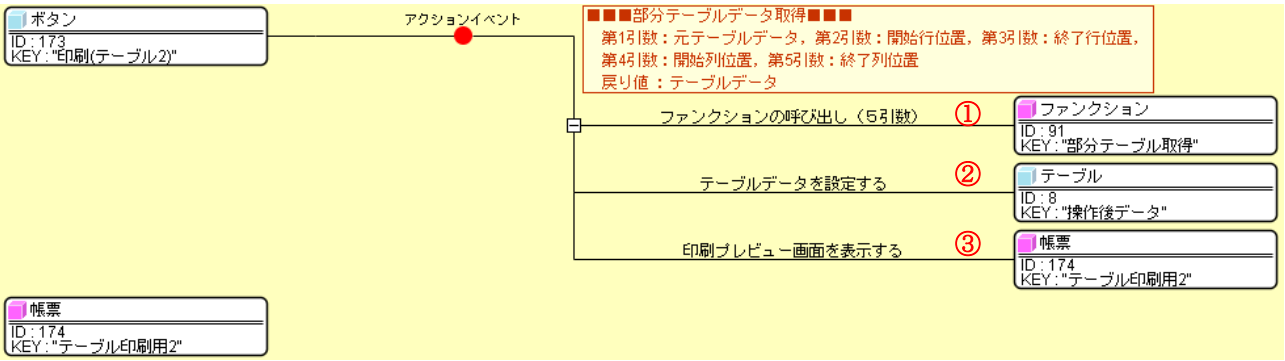
テーブル要素上でマウス右クリックし、[カラム描画設定]-[*列名*]-[描画属性設定...]と選択し、該当列の描画設定ダイアログを表示します。



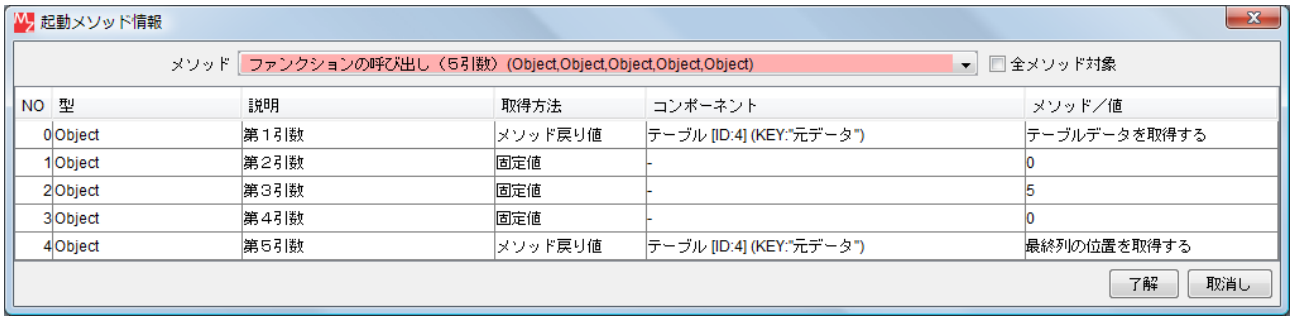
表示された「テーブルカラム描画設定」ダイアログで、「バーコード」、「QR コード」、または、「イメージ」を選択します。「イメージ」を選択した場合、テーブルのセルにはイメージファイルのパスが記述されている必要があります。

接続例

帳票の印刷は帳票コンポーネントのメソッドを実行することにより行います。ページ内にテーブルが収まるようにするため、行数を制限したテーブルを生成し、帳票テーブル要素からデータ取得先として設定しています。

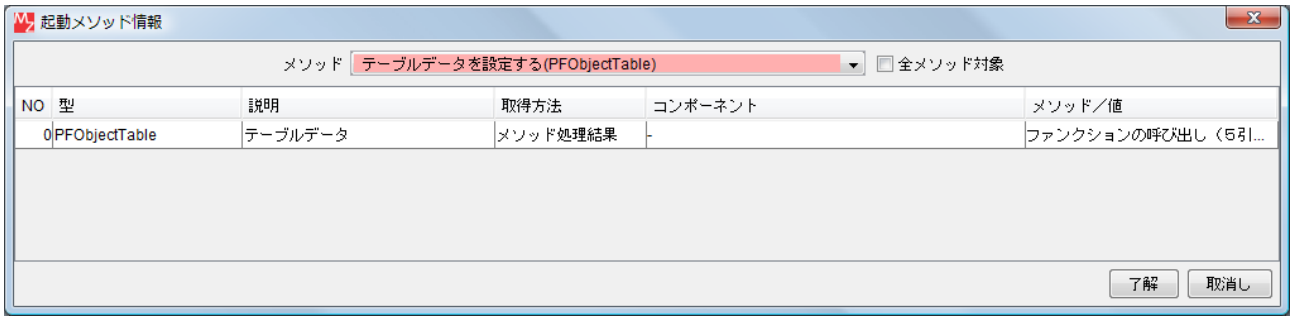


①の接続情報：部分テーブルを取得する。（5 行のテーブルを生成）



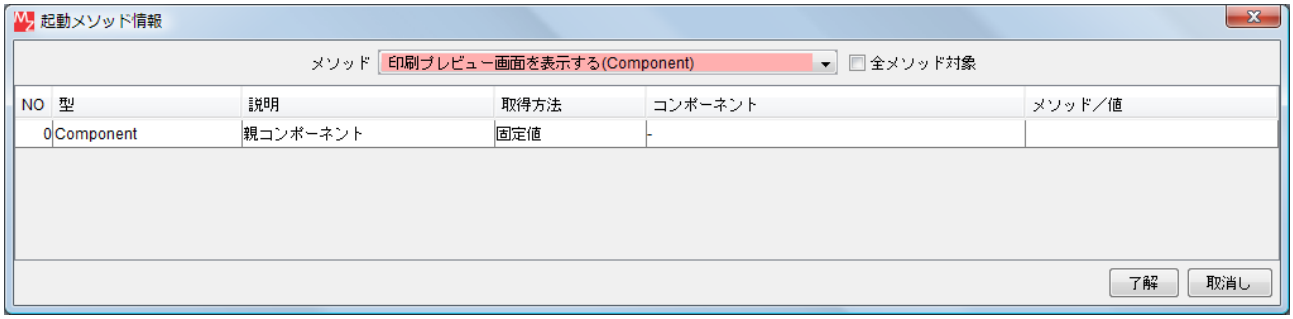
NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Object	第1引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	テーブルデータを取得する
1	Object	第2引数	固定値	-	0
2	Object	第3引数	固定値	-	5
3	Object	第4引数	固定値	-	0
4	Object	第5引数	メソッド戻り値	テーブル [ID:4] (KEY:"元データ")	最終列の位置を取得する

②の接続情報：部分テーブルを設定する。



NO	型	説明	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	メソッド処理結果	-	ファンクションの呼び出し（5引...

③の接続情報：印刷を実行する。

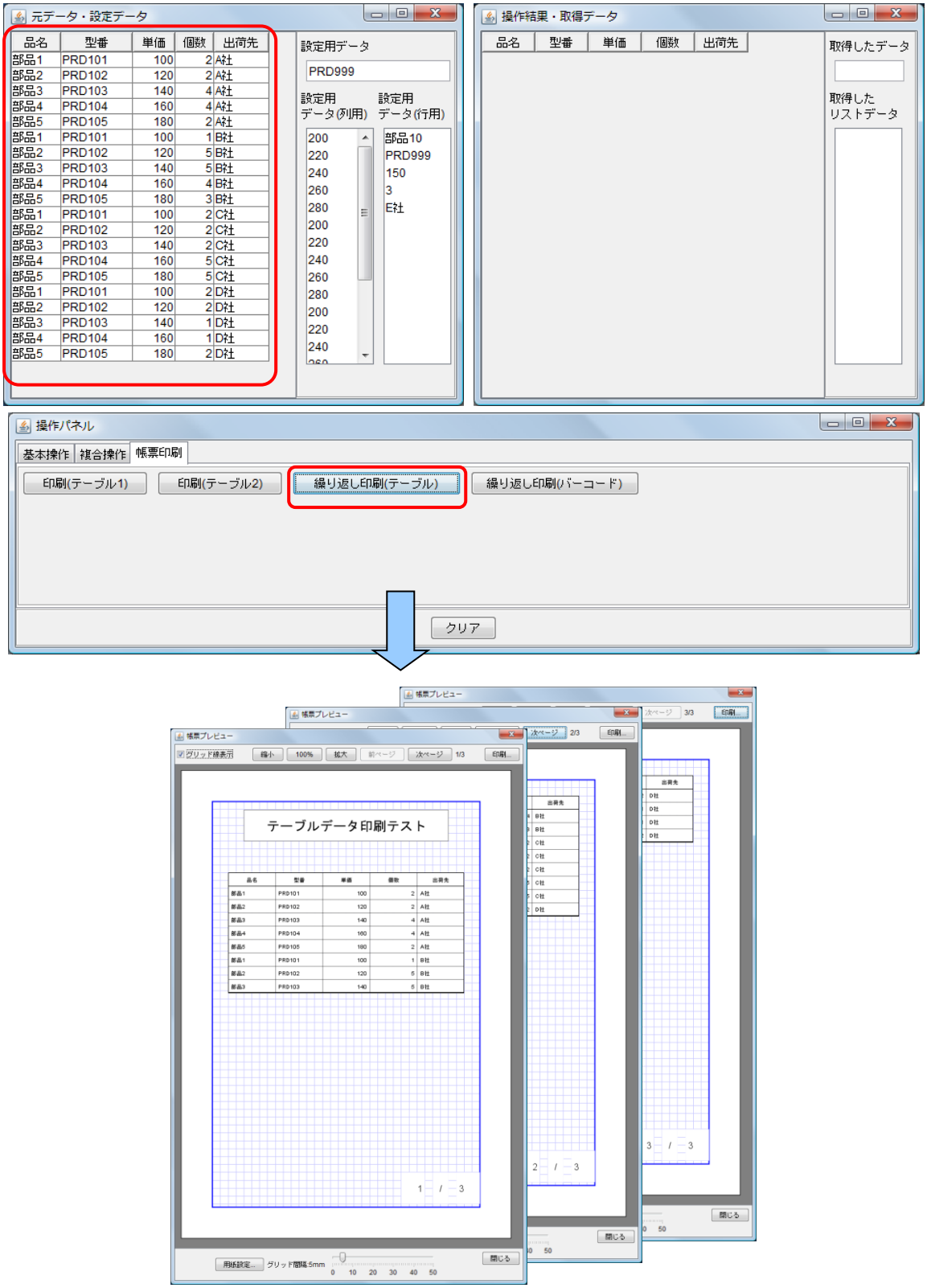


NO	型	説明	取得方法	コンポーネント	メソッド／値
0	Component	親コンポーネント	固定値	-	

4.3 繰り返し印刷 1（流し込み印刷）：テーブル編

動作イメージ

以下の図のように、テーブルデータを順に帳票のテーブル要素に流し込み、最後のテーブル要素に収まりきらない場合そのページを繰り返し印刷する機能を「繰り返し印刷」機能と呼びます。本節ではテーブルデータのテーブル要素としての繰り返し印刷のための設定について説明します。

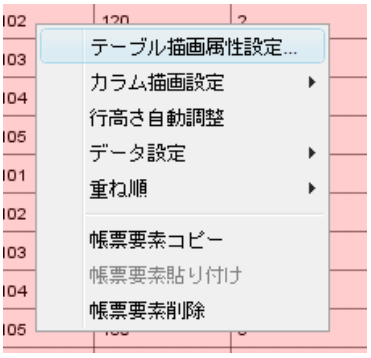


設定手順

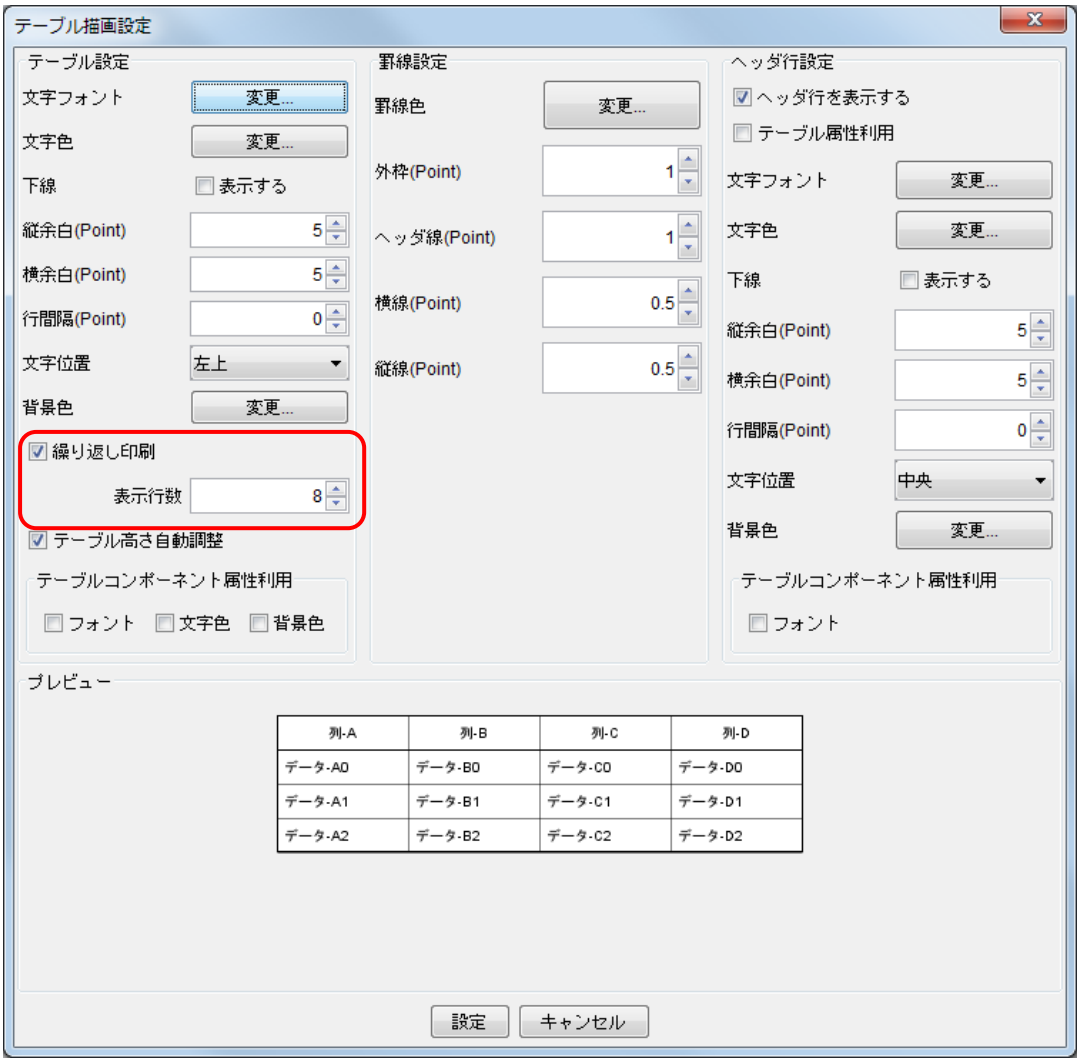
テーブル要素の繰り返し印刷機能を使用するためには、「4.1 テーブルデータの印刷」の「設定手順」-「③描画設定」-「テーブル全体の描画設定」において以下のように設定します。

・テーブル全体の描画設定

テーブル要素上でマウス右クリックし、[テーブル描画属性設定...]を選択します。

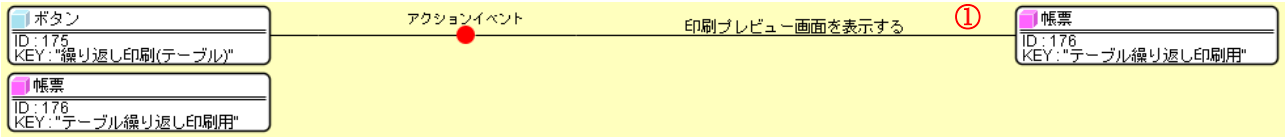


表示された「テーブル描画設定」ダイアログで、「繰り返し印刷」をチェックし、表示行数を入力してください。このように設定された 1 つの帳票内のすべてのテーブル要素は同一のテーブルデータを参照することになります。

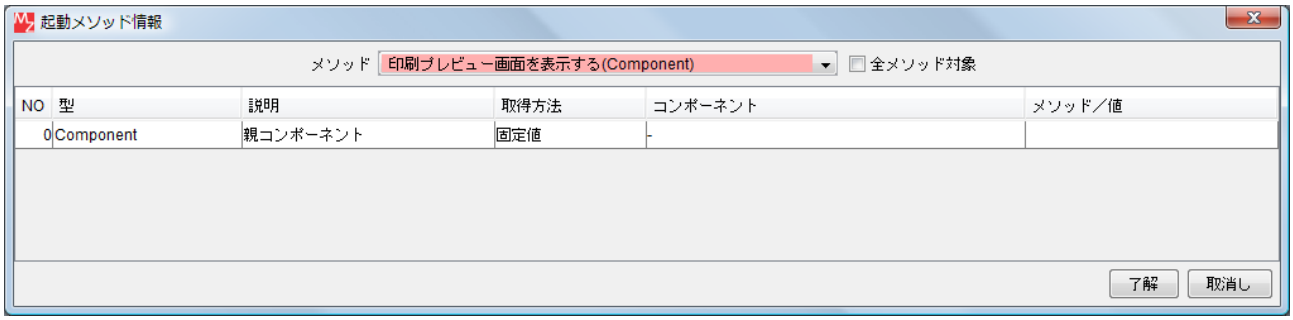


接続例

帳票の印刷は帳票コンポーネントのメソッドを実行することにより行います。



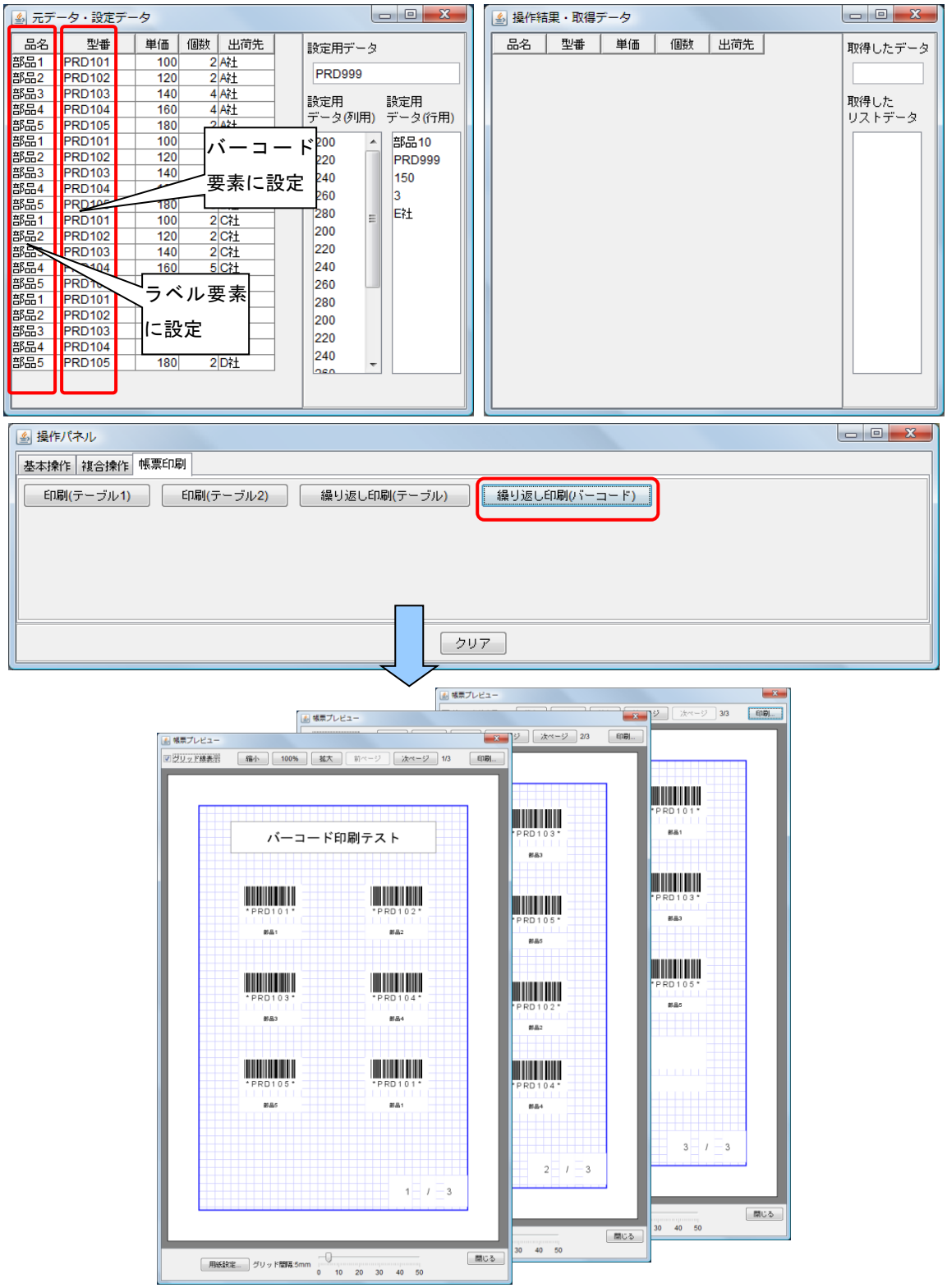
①の接続情報：印刷を実行する。



4.4 繰り返し印刷 2（流し込み印刷）：ラベル・バーコード・QR コード・イメージ編

動作イメージ

以下の図のように、テーブルデータの指定された列のデータを順に帳票のラベル要素、バーコード要素、QR コード要素、または、イメージ要素に流し込み、最後の要素に収まりきらない場合そのページを繰り返し印刷することが可能です。本節ではラベル要素、バーコード要素、QR コード要素、または、イメージ要素の繰り返し印刷のための設定について説明します。



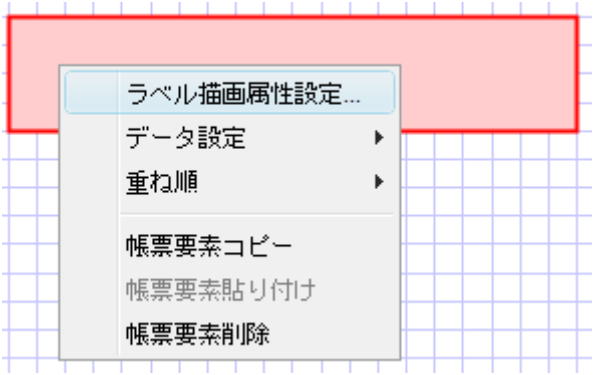
設定手順

ラベル要素、バーコード要素、QR コード要素、または、イメージ要素の繰り返し印刷機能を使用するためには、帳票編集画面で繰り返し印刷機能を使用することを指定した上で、データ取得先を設定する必要があります。

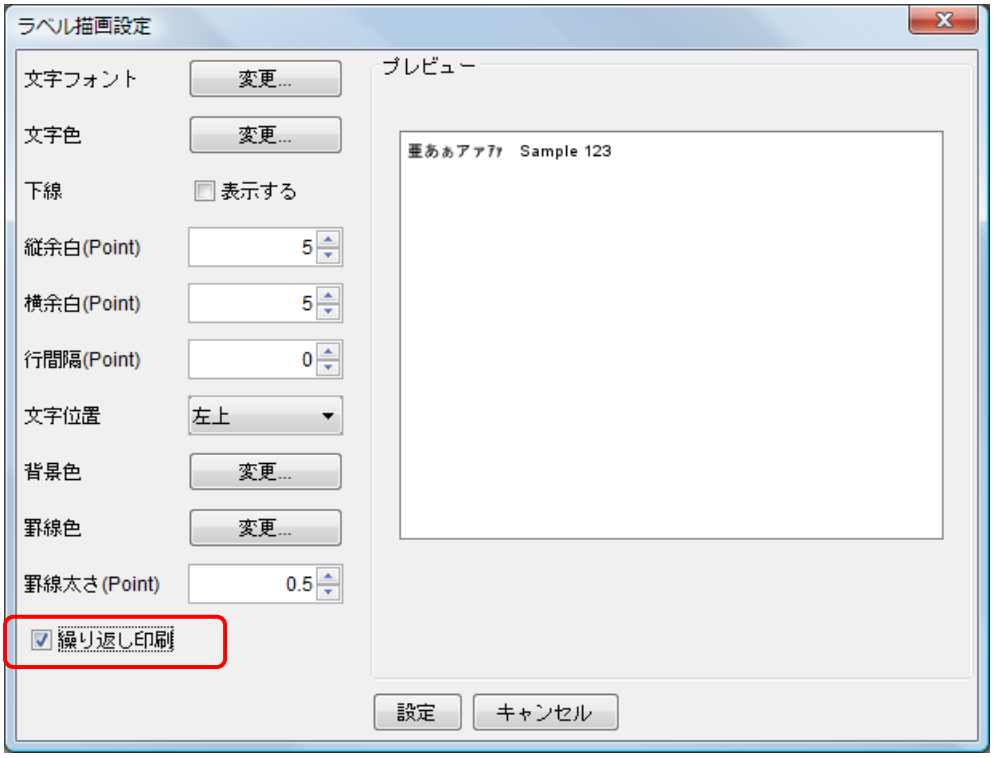
①描画属性設定

・ラベル要素の場合

帳票ラベル要素上でマウス右クリックし「ラベル描画属性設定...」を選択します。

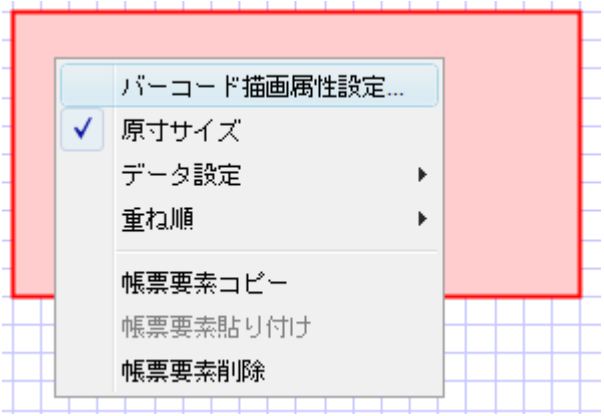


表示された「ラベル描画設定」ダイアログで、「繰り返し印刷」をチェックします。このように設定された 1 つの帳票内のすべてのラベル要素は同一のテーブルデータ指定列のデータを参照することになります。

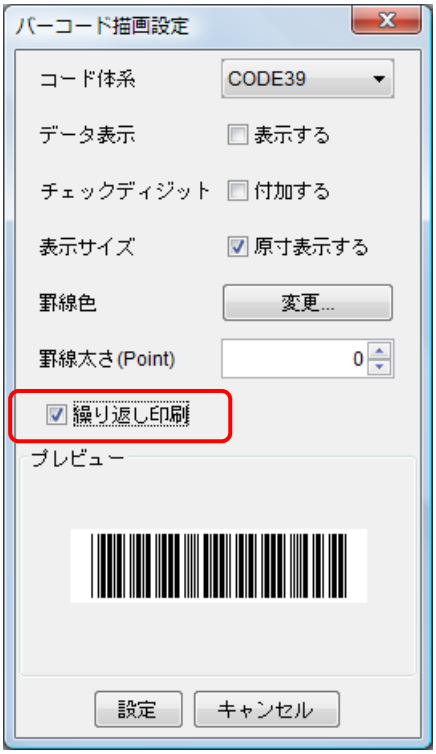


・バーコード要素の場合

帳票バーコード要素上でマウス右クリックし「バーコード描画属性設定...」を選択します。

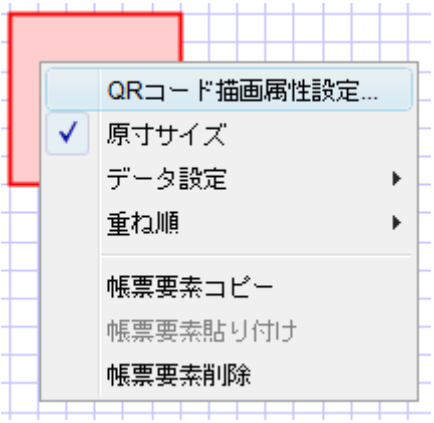


表示された「バーコード描画設定」ダイアログで、「繰り返し印刷」をチェックします。このように設定された1 つの帳票内のすべてのバーコード要素は同一のテーブルデータ指定列のデータを参照することになります。

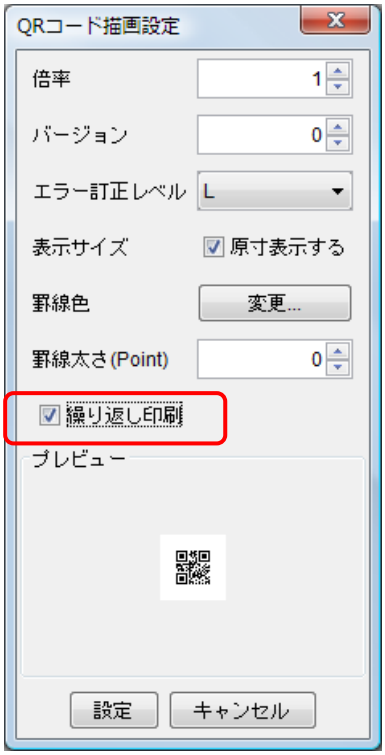


・QR コード要素の場合

帳票 QR コード要素上でマウス右クリックし「QR コード描画属性設定...」を選択します。

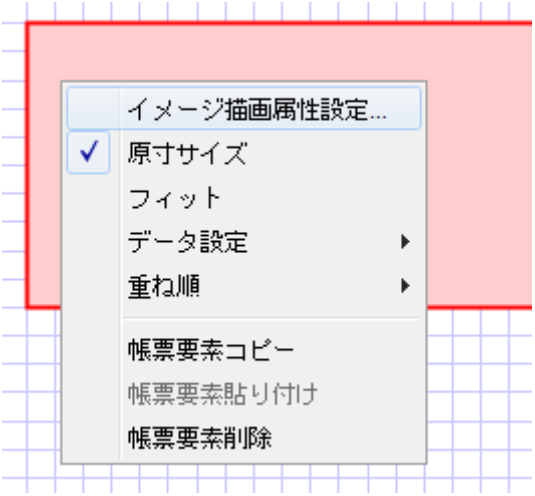


表示された「QR コード描画設定」ダイアログで、「繰り返し印刷」をチェックします。このように設定された 1 つの帳票内のすべての QR コード要素は同一のテーブルデータ指定列のデータを参照することになります。

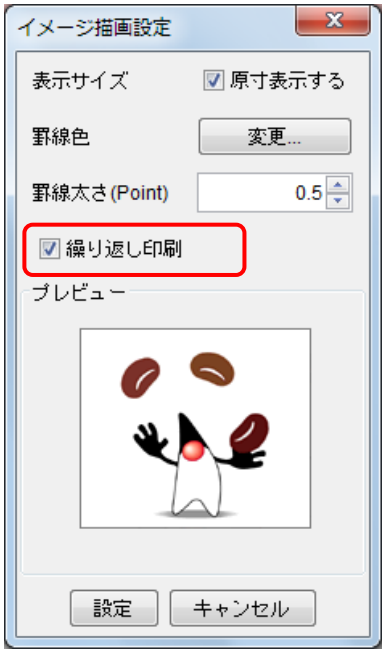


・イメージ要素の場合

帳票イメージ要素上でマウス右クリックし「イメージ描画属性設定...」を選択します。

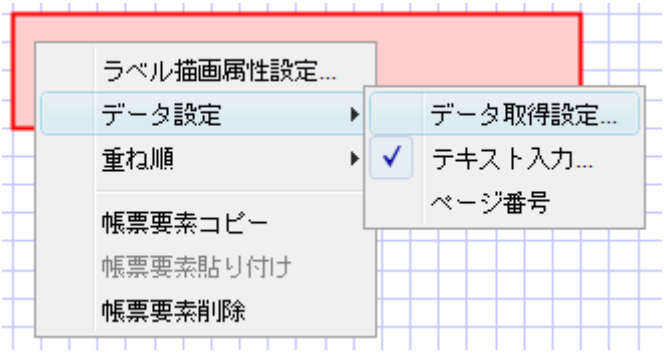


表示された「イメージ描画設定」ダイアログで、「繰り返し印刷」をチェックします。このように設定された 1 つの帳票内のすべてのイメージ要素は同一のテーブルデータ指定列のデータを参照することになります。

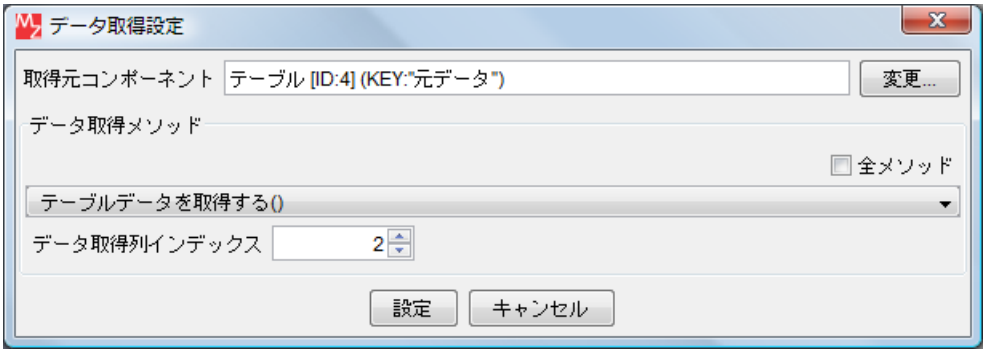


②データ取得設定

一連のラベル要素、バーコード要素、QR コード要素、イメージ要素に設定するテーブル列データの取得情報を設定します。各要素上でマウス右クリックし[データ設定]-[データ取得設定...]を選択します。

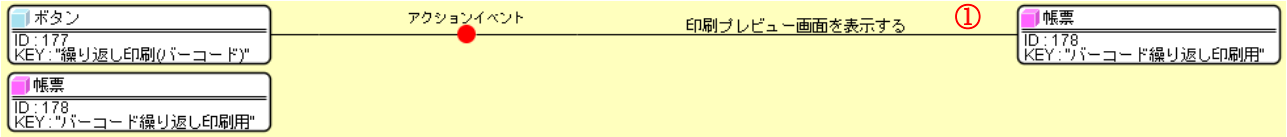
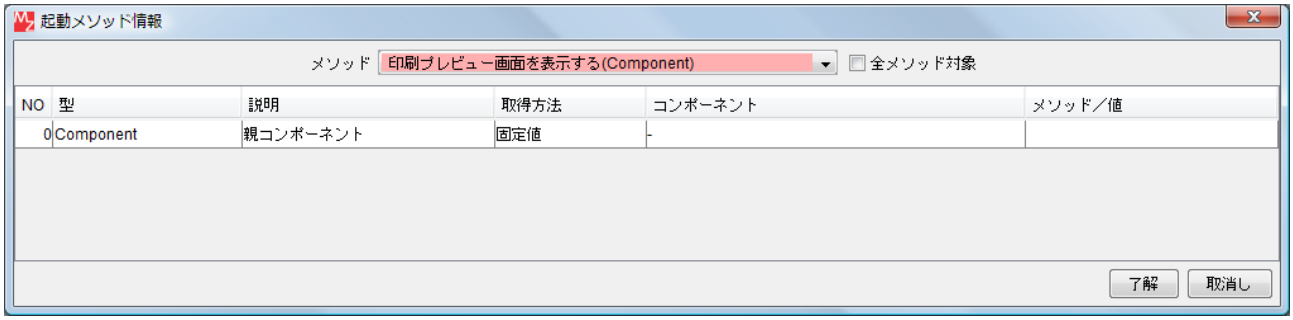


表示された「データ取得設定」ダイアログで、データ取得先コンポーネント、テーブルデータ取得メソッド、データ取得列番号（列インデックス）を指定します。



接続例

帳票の印刷は帳票コンポーネントのメソッドを実行することにより行います。



①の接続情報：印刷を実行する。