

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**

複合コンポーネントチュートリアル



独立行政法人  
産業技術総合研究所

# 複合コンポーネントチュートリアル

このチュートリアルでは、複合コンポーネントの使用法と作成方法を習得します。複合コンポーネントの作成を通じて、『サブルーチン』、『ファンクション』、『イベント生成コンポーネント』の使用法も同時に習得します。

## ◆目次

複合コンポーネントを作成してみよう .....	1
Step.1 処理をまとめるコンポーネント .....	1
Step.2 複合コンポーネントの作成（サブルーチンとイベント生成） .....	3
Step.3 複合コンポーネントの拡張（ファンクション） .....	16

## ◆図表目次

### ◇図目次

図 1 拡張したアプリケーションの画面イメージ（その 1） .....	2
図 2 複合コンポーネントのコンポーネント接続関係（その 1） .....	13
図 3 拡張したアプリケーションの画面イメージ（その 2） .....	13
図 4 拡張したアプリケーションのコンポーネント接続関係（その 1） .....	15
図 5 複合コンポーネントのコンポーネント接続関係（その 2） .....	18
図 6 拡張したアプリケーションの画面イメージ（その 3） .....	19
図 7 拡張したアプリケーションのコンポーネント接続関係（その 2） .....	21

### ◇表目次

表 1 処理をまとめるコンポーネント一覧 .....	1
表 2 コンポーネント接続関係（平均年齢以上の住所録の作成） .....	5
表 3 複合コンポーネントの公開メソッド（その 1） .....	8
表 4 コンポーネント接続関係（生成したデータの複合コンポーネント外部への伝播） .....	12
表 5 コンポーネント接続関係（テーブルデータ生成と再設定）（その 1） .....	14
表 6 コンポーネント接続関係（テーブルデータの初期化） .....	15
表 7 コンポーネント接続関係（指定年齢以上の住所録の作成） .....	16
表 8 複合コンポーネントの公開メソッド（その 2） .....	18
表 9 コンポーネント接続関係（テーブルデータ生成と再設定）（その 2） .....	20

### ◇操作手順目次

操作手順 1 複合コンポーネントを作成する .....	3
操作手順 2 複合コンポーネントに名前を設定する .....	4
操作手順 3 複合コンポーネントの公開メソッドを設定する .....	8

## 複合コンポーネントを作成してみよう

ここでは複合コンポーネントの作成方法を以前のレッスンで作成したアプリケーションを拡張しながら習得します。同時に、処理をまとめるコンポーネントである『サブルーチン』『ファンクション』『イベント生成』について、複合コンポーネントを作成する立場から使用方法を習得します。

### Step.1 処理をまとめるコンポーネント

これまで、ビルダーを用いてコンポーネント間を接続してアプリケーションを作成する方法を学んできました。アプリケーションの構築においては、作成したアプリケーションがきちんと動くことが最も重要であることは確かですが、全体の処理の流れを見やすく、理解しやすく作成することもとても重要です。複雑な構成のままでは、他の人だけではなく作成した本人にとっても、処理の流れを理解することが難しくなり、アプリケーションを再利用したり、メンテナンスをすることが困難になるからです。このような問題の一つの解決方法として、複数の処理をある塊ごとにまとめる方法が考えられます。MZ Platform では、処理をまとめるためのコンポーネントとして、『サブルーチン』『ファンクション』『イベント生成』が用意されています。また複数のコンポーネントをまとめ、あたかも一つのコンポーネントとして取り扱うことを可能にする『複合コンポーネント』を作成することも可能です。『サブルーチン』『ファンクション』『イベント生成』『複合コンポーネント』の比較を次の表に示します。

表 1 処理をまとめるコンポーネント一覧

コンポーネント名	メソッド名	引数	戻り値	発生イベント
サブルーチン PFSubroutine	処理を呼び出す	—	—	アクションイベント
	イベントを渡して処理を呼び出す	イベント	—	引数として指定されたイベント
ファンクション PFFunction	ファンクションの呼出し	オブジェクト (0~9 個まで)	接続先コンポーネント（最後）の戻り値	処理要求イベント
	ファンクションの呼出し(引数リスト指定)	リスト		
イベント生成 PFEventGenerator	〇〇イベントの発生 (処理要求イベント以外)	オブジェクト、 整数、イベント 番号(メソッド により異なる)	—	〇〇イベント (メソッド名に対応)
	処理要求イベントの発生	オブジェクト、 イベント番号	接続先コンポーネント（最後）の戻り値	処理要求イベント
複合コンポーネント	任意	任意	任意	任意

サブルーチンとは、頻繁に利用する小さな処理を個別に作成し、必要に応じて呼び出すという手法です。『サブルーチン』コンポーネントでは、メソッドが呼び出された際に、接続先のコンポーネントの処理が実行されます。表に示した 2 つのメソッドの違いは、メソッドを呼び出す際に、引数として呼び出し側のコンポーネントから発生するイベントを渡すか否かにあります。ファンクションとは「関数」であり、入力値（引

数) に対し、それに対応した出力 (戻り値) の関係を示すものです。『ファンクション』コンポーネントでは、メソッドが呼び出された際に、接続先のコンポーネントのメソッドを順に実行し、最後に実行したメソッドの処理結果を戻り値として返します。『ファンクション』コンポーネントのメソッドには 0 個以上の引数が設定でき、引数を設定した際にはその引数を接続先コンポーネントのメソッド実行時に使用することができます。『イベント生成』コンポーネントは任意のイベントを発生させることができるコンポーネントです。この際メソッドに引数としてイベントの内包データを指定することができます。発生したイベントを用いて接続先のコンポーネントのメソッドを実行する必要があるときに使用します。『複合コンポーネント』は、上述したように、複数のコンポーネントをまとめ、あたかも 1 つのコンポーネントとして取り扱うことを可能にする概念で、新たに階層を作成すると言う意味で、他の 3 つのコンポーネントとは次元が異なります。

このレッスンでは、「アプリケーション開発チュートリアル」の Lesson.6 の住所録を複合コンポーネントを用いながら拡張する過程の中で、各コンポーネントの使用方法を習得します。もちろん、『サブルーチン』、『ファンクション』、『イベント生成』は単独でも使えますが、このレッスンでは『複合コンポーネント』の作成を通して、主要な使用方法に限定して説明していきます。下図にこのレッスンで拡張したアプリケーションの画面イメージを示します。このアプリケーションでは住所録から、平均年齢以上の人の住所録と、指定した年齢以上の人の住所録を表示できるように拡張します。

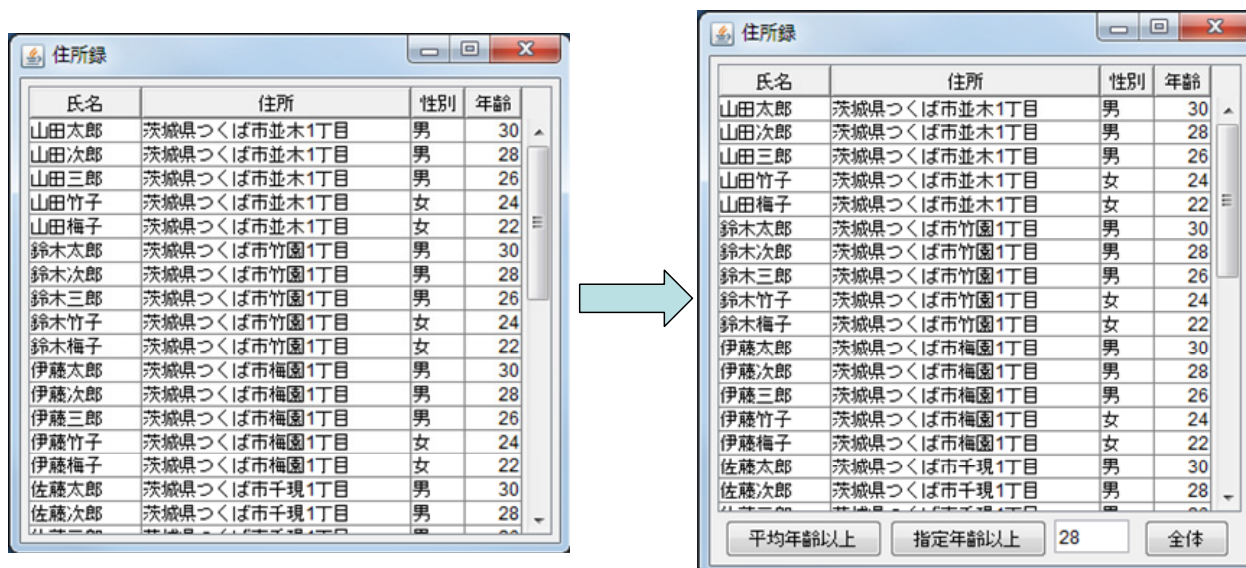


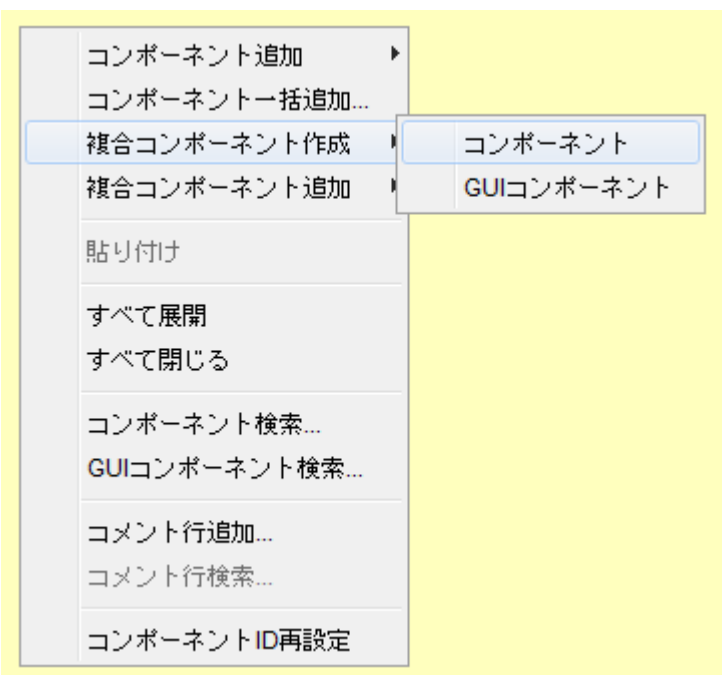
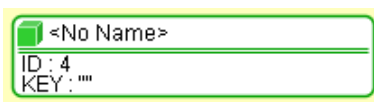
図 1 拡張したアプリケーションの画面イメージ (その 1)

## Step.2 複合コンポーネントの作成（サブルーチンとイベント生成）

このレッスンでは住所録のデータがある条件でフィルタリングし、住所録のサブセットを得るための複合コンポーネントを作成します。このステップでは、『サブルーチン』、『イベント生成』コンポーネント等を用いて、表示されている住所録から平均年齢以上のメンバーの住所録を得る機能を作成します。

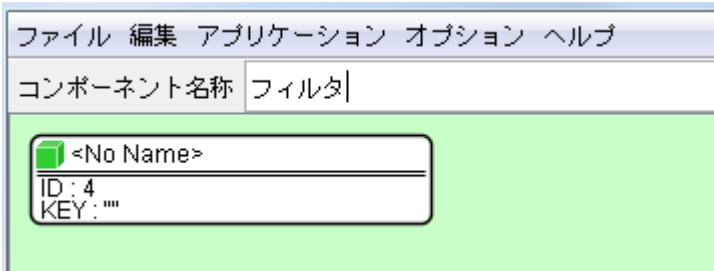
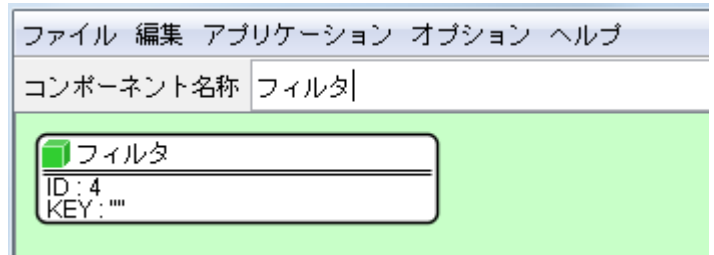
それでは、はじめに元になるアプリケーションをロードします。ビルダー画面の[ロード]ボタンを押して、“AP\_DATA¥Tutorial” に収録されている “Lesson06(住所録)-2.mzax” をロードしましょう。ロードができれば、早速、複合コンポーネントの作成に移ります。複合コンポーネントの作成は、次の手順で行います。

### 操作手順 1 複合コンポーネントを作成する

操作手順：複合コンポーネントを作成する	
1 操作	<p>ビルダー画面上でマウスを右クリックし、 [複合コンポーネント作成]－[コンポーネント]を選択</p> 
結果	<p>複合コンポーネントが画面上に表示される</p> 

追加された緑色の枠のコンポーネントが複合コンポーネントです。それでは実際に中身を作成していきます。この複合コンポーネントをダブルクリックしてください。画面の背景色が変化したと思います。この画面が複合コンポーネントの構築画面です。はじめに、複合コンポーネントの名前を設定しましょう。複合コンポーネントの名前は、次の手順で設定します。ここでは、複合コンポーネントの名前を“フィルタ”と設定することとします。

## 操作手順 2 複合コンポーネントに名前を設定する

操作手順：複合コンポーネントに名前を設定する		
1	操作	画面上の[コンポーネント名称]の空欄にコンポーネント名を入力し Enter キーを押す 
	結果	複合コンポーネントに名前が設定される 

次に複合コンポーネントで用いるコンポーネントを追加します。追加するコンポーネントは『サブルーチン』、『テーブル格納変数』、『基本統計処理(数値)』、『比較(≧)』、『繰り返し制御(FOR)』、『イベント生成』、コンポーネントです。コンポーネントのカテゴリーは次のとおりです。

コンポーネント名	必要数	
■ サブルーチン	1	[処理部品]－[サブルーチン]－[サブルーチン]
■ テーブル格納変数	1	[処理部品]－[変数]－[テーブル格納変数]
■ 基本統計処理(数値)	1	[処理部品]－[統計]－[基本統計処理(数値)]
■ 比較演算(≧)	1	[処理部品]－[条件制御]－[比較演算(≧)]
■ 繰り返し制御(FOR)	1	[処理部品]－[条件制御]－[繰り返し制御(FOR)]
■ イベント生成	1	[処理部品]－[イベント]－[イベント生成]

いま、テーブル格納変数に住所録のテーブルデータが設定されていると仮定します。このテーブルデータから平均年齢以上のメンバー以外を削除する処理をサブルーチン (ID:4-1) にまとめてみましょう。平均年齢以上のメンバーをテーブルデータから削除する手順は、『テーブルから年齢の列 (インデックス 3) のデータをリストとして取得する』、『年齢を集計し平均年齢を算出する』、『テーブルの第 3 列の値を順に取得し平均値と比較する』、『年齢が平均値以上でなければ、テーブルから対象行を削除する』と表現できます。『サブルーチン』コンポーネントのメソッド『処理を呼び出す()』が実行された場合、アクションイベントが発生します。アクションイベントが発生した際にこれら処理を実現するように、次のようにコンポーネント間を接続します。

表 2 コンポーネント接続関係（平均年齢以上の住所録の作成）

項目	内容
イベント発生元コンポーネント	■ サブルーチン (ID:4-1)
発生イベント	アクションイベント
イベント番号	定常起動
接続先	<p>① テーブルから年齢の列をリストとして取得する</p> <p>◇ 接続先コンポーネント</p> <p>■ テーブル格納変数 (ID:4-2)</p> <p>◇ 起動メソッド</p> <p>列データリストを位置指定で取得する (int)</p> <p>&lt;引数: インデックス&gt;</p> <p>取得方法: 『固定値』</p> <p>値: “3”</p> <p>② 平均年齢を算出するためデータを設定する</p> <p>◇ 接続先コンポーネント</p> <p>■ 基本統計処理 (数値) (ID:4-3)</p> <p>◇ 起動メソッド</p> <p>集計対象データを設定する (PFObjectList)</p> <p>&lt;引数: 集計対象データ&gt;</p> <p>取得方法: 『メソッド処理結果』</p> <p>取得メソッド: “列データリストを位置指定で取得する”</p> <p>③ 平均年齢を計算し比較演算子に設定</p> <p>◇ 接続先コンポーネント</p> <p>■ 比較演算 (≥) (ID:4-4)</p> <p>◇ 起動メソッド</p> <p>数値に変換して右オペランドに設定する (String)</p> <p>&lt;引数: 右オペランド&gt;</p> <p>取得方法: 『メソッド戻り値』</p> <p>取得先コンポーネント: 基本統計処理 (数値) (ID:4-3)</p> <p>取得メソッド: “平均値を取得する”</p>

④繰り返し処理(テーブルデータの最終行から開始行の繰り返し)

◇接続先コンポーネント

- 繰り返し制御(FOR) (ID:4-5)

◇起動メソッド

繰り返し処理を実行する (int, boolean, int, boolean, int)

<引数: 開始値>

取得方法: 『メソッド戻り値』

取得先コンポーネント: テーブル格納変数 (ID:4-2)

取得メソッド: “行数を取得する”

<引数: 開始値を含めるかどうか>

取得方法: 『固定値』

値: “false”

<引数: 終了値>

取得方法: 『固定値』

値: “0”

<引数: 終了値を含めるかどうか>

取得方法: 『固定値』

値: “true”

<引数: 増減値>

取得方法: 『固定値』

値: “-1”

項目	内容
イベント発生元コンポーネント	■繰り返し制御(FOR) (ID:4-5)
発生イベント	アクションイベント
イベント番号	定常起動
接続先	<p>①指定行の年齢データを取得する</p> <p>◇接続先コンポーネント</p> <ul style="list-style-type: none"> <li>■テーブル格納変数 (ID:4-2)</li> </ul> <p>◇起動メソッド</p> <p>セルデータを指定位置で取得する (int, int)</p> <p>&lt;引数: 指定行&gt;</p> <p>取得方法: 『メソッド戻り値』</p> <p>取得先コンポーネント: 繰り返し制御(FOR) (ID:4-5)</p> <p>取得メソッド: “現在値を取得する ()”</p> <p>&lt;引数: 指定列&gt;</p> <p>取得方法: 『固定値』</p> <p>値: “3”</p>



	<p>②平均年齢と比較するため比較演算子に設定</p> <p>◇接続先コンポーネント</p> <ul style="list-style-type: none"> <li>■比較演算(≥) (ID:4-4)</li> </ul> <p>◇起動メソッド</p> <p>数値に変換して左オペランドを設定する (String)</p> <p>&lt;引数: 左オペランド&gt;</p> <p>取得方法: 『メソッド処理結果』</p> <p>取得メソッド: “セルデータを指定位置で取得する”</p>
	<p>③比較演算を実行する</p> <p>◇接続先コンポーネント</p> <ul style="list-style-type: none"> <li>■比較演算 (≥) (ID:4-4)</li> </ul> <p>◇起動メソッド</p> <p>演算を行う()</p>

項目	内容
イベント発生元コンポーネント	■比較演算(≥) (ID:4-4)
発生イベント	処理完了イベント
イベント番号	0
接続先	<p>◇接続先コンポーネント</p> <ul style="list-style-type: none"> <li>■テーブル格納変数 (ID:4-2)</li> </ul> <p>◇起動メソッド</p> <p>行を指定位置で削除する (int)</p> <p>&lt;引数: 指定位置&gt;</p> <p>取得方法: 『メソッド戻り値』</p> <p>取得先コンポーネント: 繰り返し制御(FOR) (ID:4-5)</p> <p>取得メソッド: “現在値を取得する()”</p>

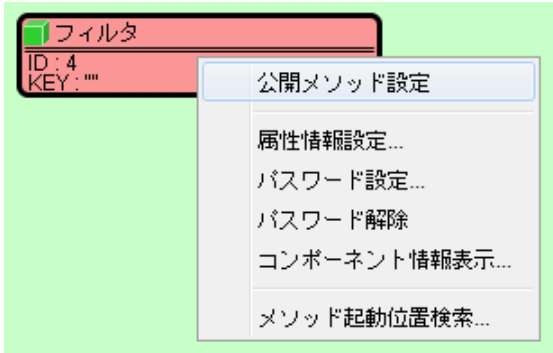
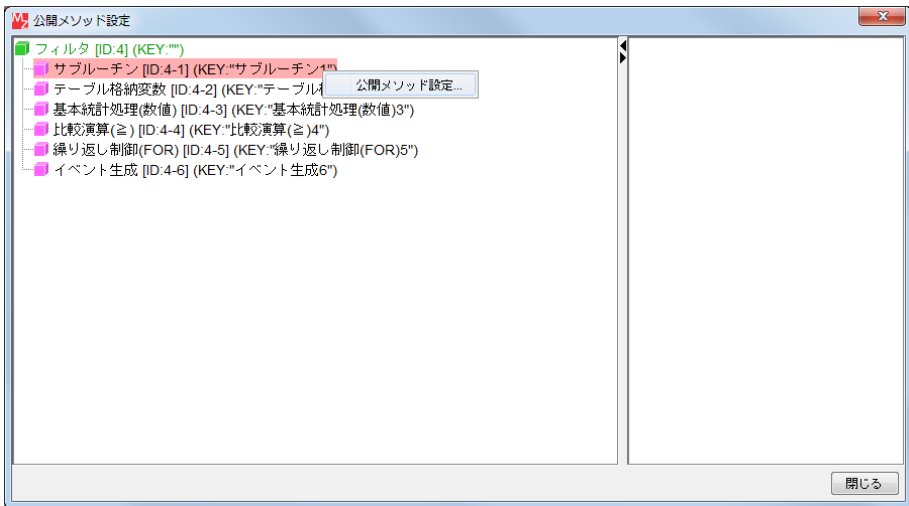
このままでは外部のコンポーネントから、複合コンポーネントに含まれるコンポーネントのメソッドを呼び出すことができません。外部からのメソッドの呼び出しを可能にするには、メソッドを公開する必要があります。このメソッドを「公開メソッド」と呼びます。住所録をこの複合コンポーネントに設定し、平均年齢以上の住所録を作成できるように以下のメソッドを公開メソッドとして設定する必要があります。

表 3 複合コンポーネントの公開メソッド (その 1)

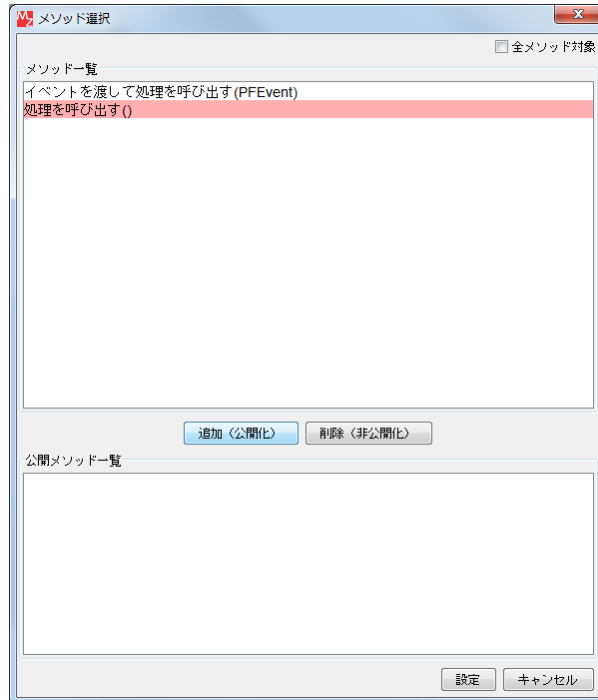
コンポーネント名	メソッド名
■ テーブル格納変数 (ID:4-2)	テーブルを設定する (PFObjectTable)
■ サブルーチン (ID:4-1)	処理を呼び出す ()

それではこれら 2 つの公開メソッドを設定しましょう。公開メソッドの設定手順は次のとおりです。

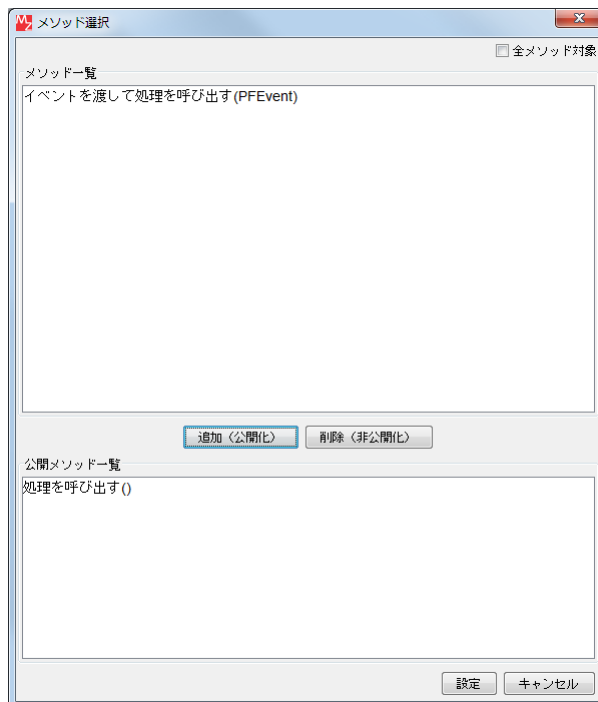
### 操作手順 3 複合コンポーネントの公開メソッドを設定する

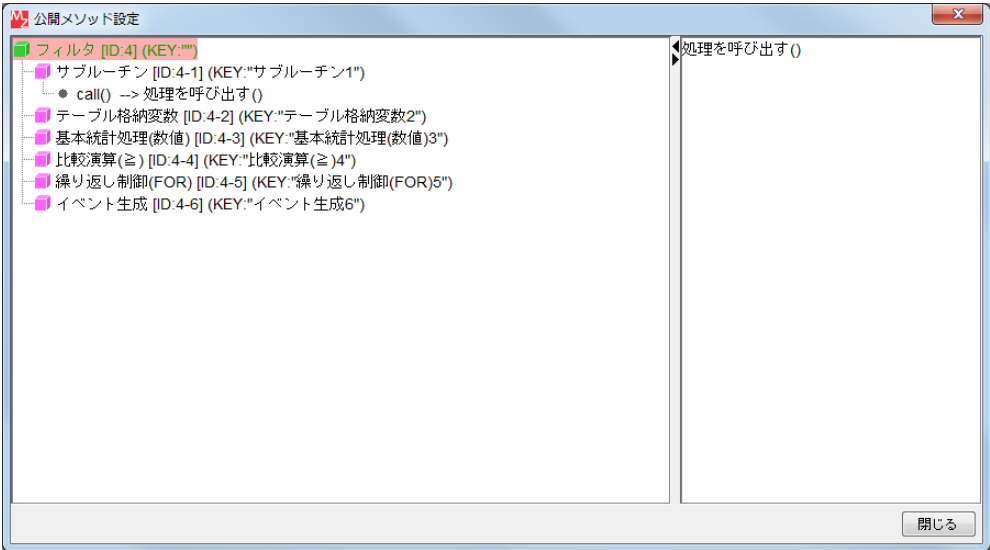
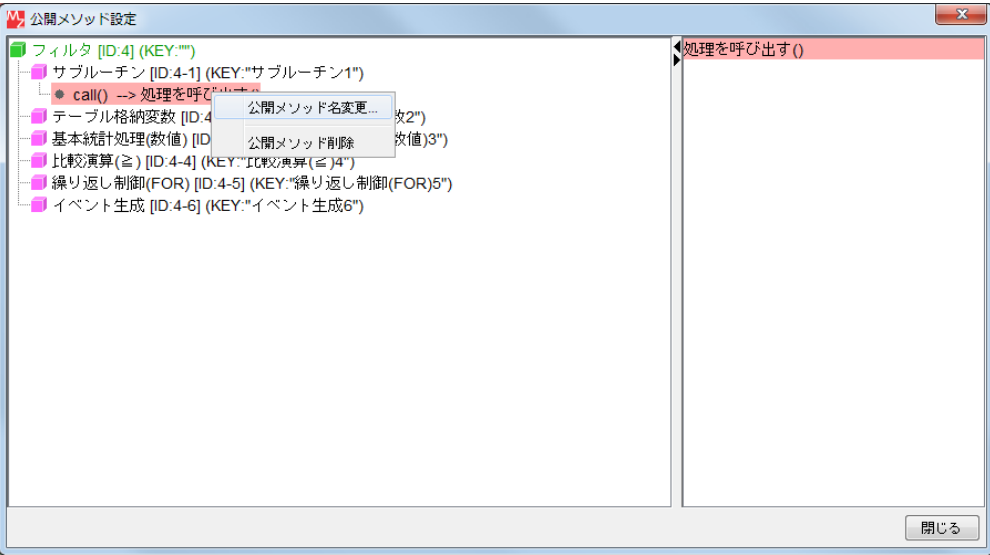
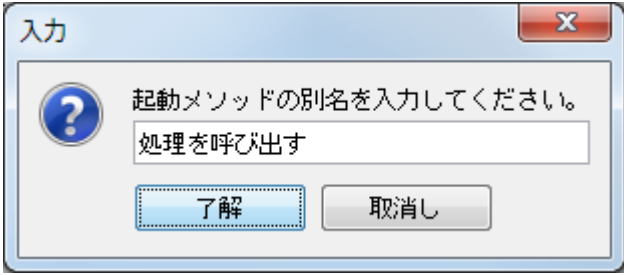
操作手順：複合コンポーネントの公開メソッドを設定する	
1	<p>操作 複合コンポーネント上でマウスを右クリックし、 [公開メソッド設定] を選択</p>  <p>結果 公開メソッド設定画面が表示される</p>
2	<p>操作 公開したいメソッドを持つコンポーネント名上でマウスを右クリックし、 [公開メソッド設定...] を選択</p>  <p>結果 メソッド選択画面が表示される</p>

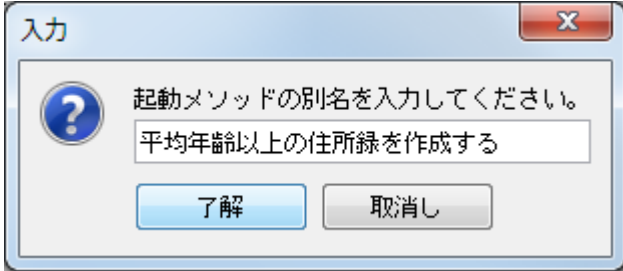
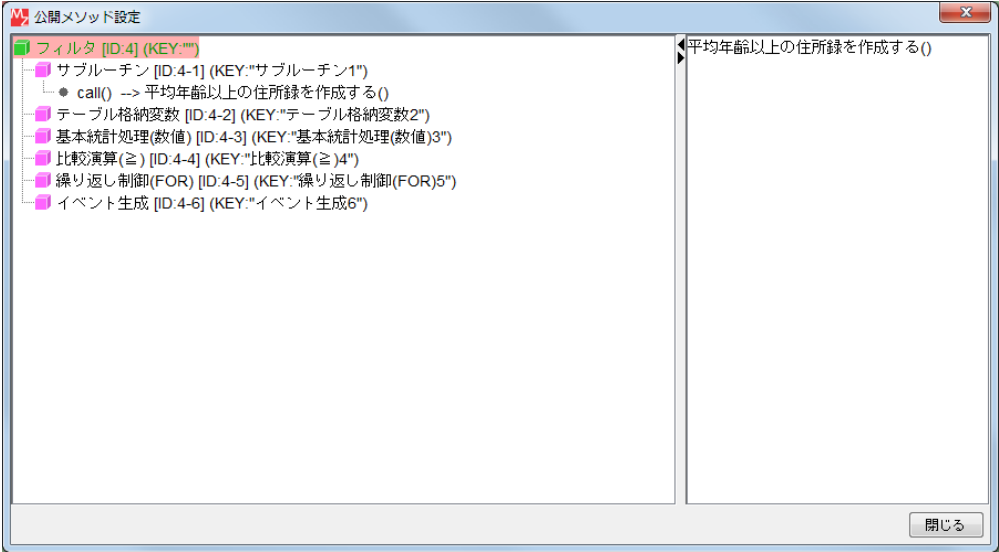
3 操作 メソッド一覧から公開したいメソッドを選択し、**追加（公開化）** ボタンをクリック



結果 公開メソッド一覧に選択したメソッドが追加される



<p><b>4</b> 操作</p>	<p><b>設定</b> ボタンをクリック</p> <p>結果 公開メソッド設定画面に公開メソッドが表示される</p> 
<p><b>5</b> 操作</p>	<p>公開メソッド名上でマウスを右クリックし、 [公開メソッド名変更...]を選択</p> 
<p>結果</p>	<p>公開メソッド名変更のダイアログが表示される</p> 

6	操作	公開メソッド名を入力し了解ボタンを押す 
	結果	公開メソッド名が変更される 
7	操作	同様にして、テーブル格納変数コンポーネントの“テーブルを設定する ()”メソッドも公開します。設定が終了したら、閉じるボタンをクリック
	結果	公開メソッドの設定が完了

上述した2つのメソッドが設定されていることを確認して、公開メソッドの設定は終了です。

次に生成したテーブルデータを外部から取得可能なように、テーブルデータを内包した「データ生成イベント」を発生し、複合コンポーネント外部に伝播します。データ生成イベントの発生にはイベント生成 (ID:4-6) のメソッド『データ生成イベントの発生(Object, int)』を実行します。引数の Object は生成データ、int はイベント番号です。イベント番号はとりあえず“1”とします。このイベント生成 (ID:4-6) から発生したデータ生成イベントを外部へ伝播するには、複合コンポーネント編集画面の一番上の複合コンポーネント「フィルタ」(ID:4)に接続する必要があります。コンポーネント間を次のように接続します。

表 4 コンポーネント接続関係（生成したデータの複合コンポーネント外部への伝播）

項目	内容
イベント発生元コンポーネント	■サブルーチン（ID:4-1）
発生イベント	アクションイベント
イベント番号	定常起動
接続先	◇接続先コンポーネント ■イベント生成（ID:4-6） ◇起動メソッド データ生成イベントの発生(Object, int) <引数：生成データ> 取得方法：『メソッド戻り値』 取得先コンポーネント：テーブル格納変数（ID:4-2） 取得メソッド：“テーブルを取得する（）” <引数：イベント番号> 取得方法：『固定値』 値：“1”

項目	内容
イベント発生元コンポーネント	■イベント生成（ID:4-6）
発生イベント	データ生成イベント
イベント番号	定常起動
接続先	◇接続先コンポーネント ■フィルタ（ID:4） ◇起動メソッド イベントを伝播させる（PFEvent） <引数：対象イベント> 取得方法：『イベント』

このステップでの複合コンポーネント内のコンポーネント間の接続は完了しました。参考のため、接続の様子を下図に示します。

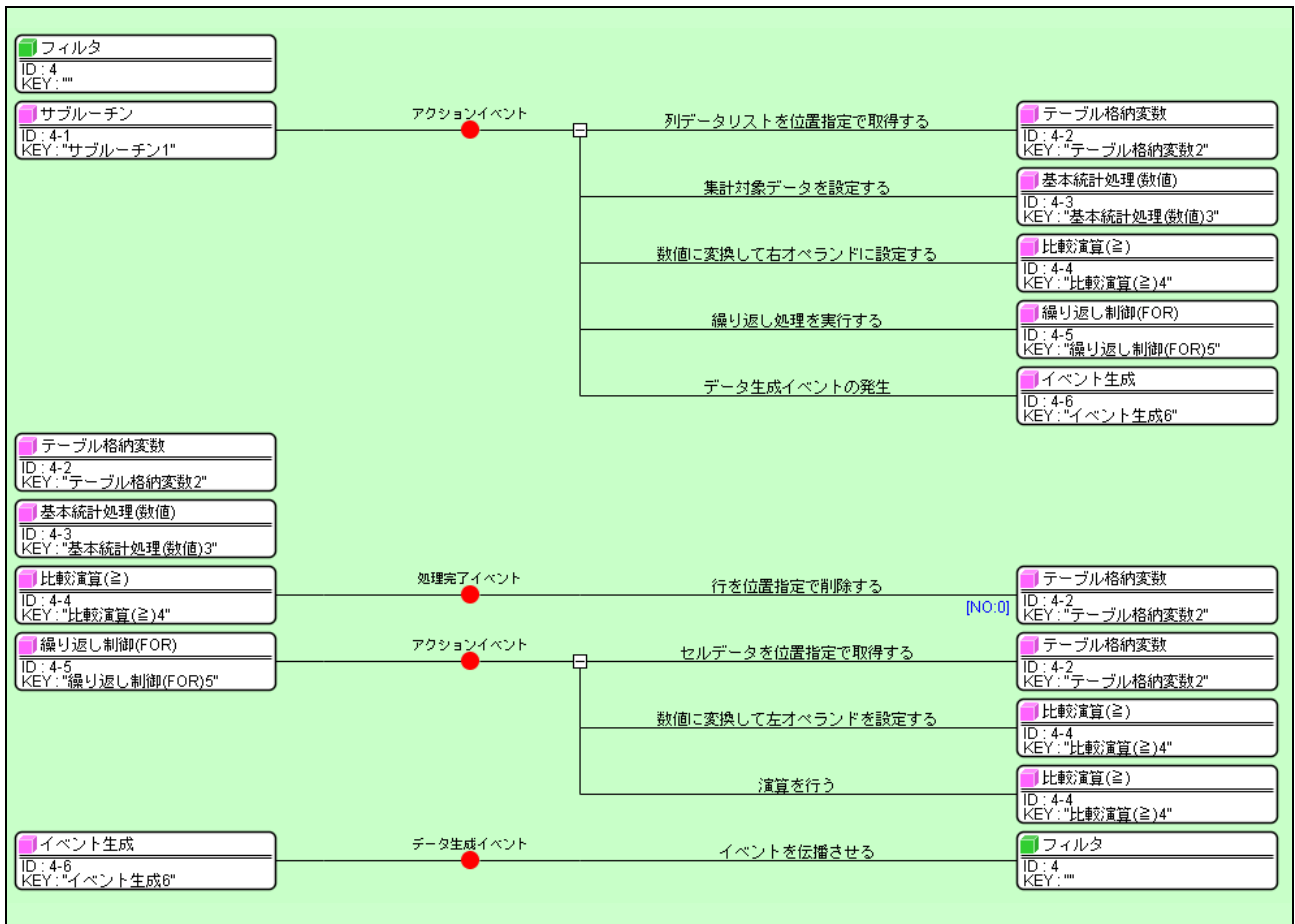


図 2 複合コンポーネントのコンポーネント接続関係 (その 1)


それでは、右上のを押して、アプリケーションのビルダー画面に移り、他のコンポーネントからテーブルデータの生成を指示してみましょう。その前に、その指示を与える『ボタン』と、テーブルを初期状態に戻す『ボタン』コンポーネントを追加します。次の図を参考にボタン(ID:5) とボタン(ID:6)をフレーム(ID:1)に配置し、ボタンのテキストに“平均年齢以上”、“全体”とそれぞれ設定します。



図 3 拡張したアプリケーションの画面イメージ (その 2)

[平均年齢以上]ボタンが押された際に、フィルタ (ID:4) にテーブルデータを設定して、平均年齢以上の住所録を生成し、生成されたテーブルデータをテーブル (ID:2) に設定するよう、次のようにコンポーネント間を接続します。

表 5 コンポーネント接続関係 (テーブルデータ生成と再設定) (その 1)

項目	内容
イベント発生元コンポーネント	■ ボタン (ID:5)
発生イベント	アクションイベント
イベント番号	定常起動
接続先	<p>①テーブルデータを設定する</p> <p>◇接続先コンポーネント</p> <p>■ フィルタ (ID:4)</p> <p>◇起動メソッド</p> <p>テーブルを設定する (PFObjectTable)</p> <p>&lt;引数: 設定するテーブル&gt;</p> <p>取得方法: 『メソッド戻り値』</p> <p>取得先コンポーネント: テーブル (ID:2)</p> <p>取得メソッド: “テーブルデータを取得する ()”</p> <p>②平均年齢以上の住所録を生成する</p> <p>◇接続先コンポーネント</p> <p>■ フィルタ (ID:4)</p> <p>◇起動メソッド</p> <p>平均年齢以上の住所録を作成する ()</p>

項目	内容
イベント発生元コンポーネント	■ フィルタ (ID:4)
発生イベント	データ生成イベント
イベント番号	1
接続先	<p>◇接続先コンポーネント</p> <p>■ テーブル (ID:2)</p> <p>◇起動メソッド</p> <p>テーブルデータを設定する (PFObjectTable)</p> <p>&lt;引数: テーブルデータ&gt;</p> <p>取得方法: 『イベント内包』</p> <p>取得メソッド: “イベント対象データ”</p>

また、[全体]ボタンが押された際に、テーブルを初期状態に戻せるように次のようにコンポーネント間を接続します。



表 6 コンポーネント接続関係 (テーブルデータの初期化)

項目	内容
イベント発生元コンポーネント	■ ボタン (ID:6)
発生イベント	アクションイベント
イベント番号	定常起動
接続先	◇接続先コンポーネント ■ テーブル (ID:2) ◇起動メソッド テーブルデータを設定する (PFObjectTable) <引数: テーブルデータ> 取得方法: 『メソッド戻り値』 取得先コンポーネント: 住所録 (Lesson. 6) (ID:3) 取得メソッド: “住所録データを取得する ()”

この時点でアプリケーションを保存し実行してみましょう。平均年齢以上の住所録が表示されましたか？ テーブルデータを初期状態に戻せますか？ うまく表示されない場合には、もう一度コンポーネント間の接続を確認してください。参考まで、ここまでのコンポーネント間の接続の様子を示します。

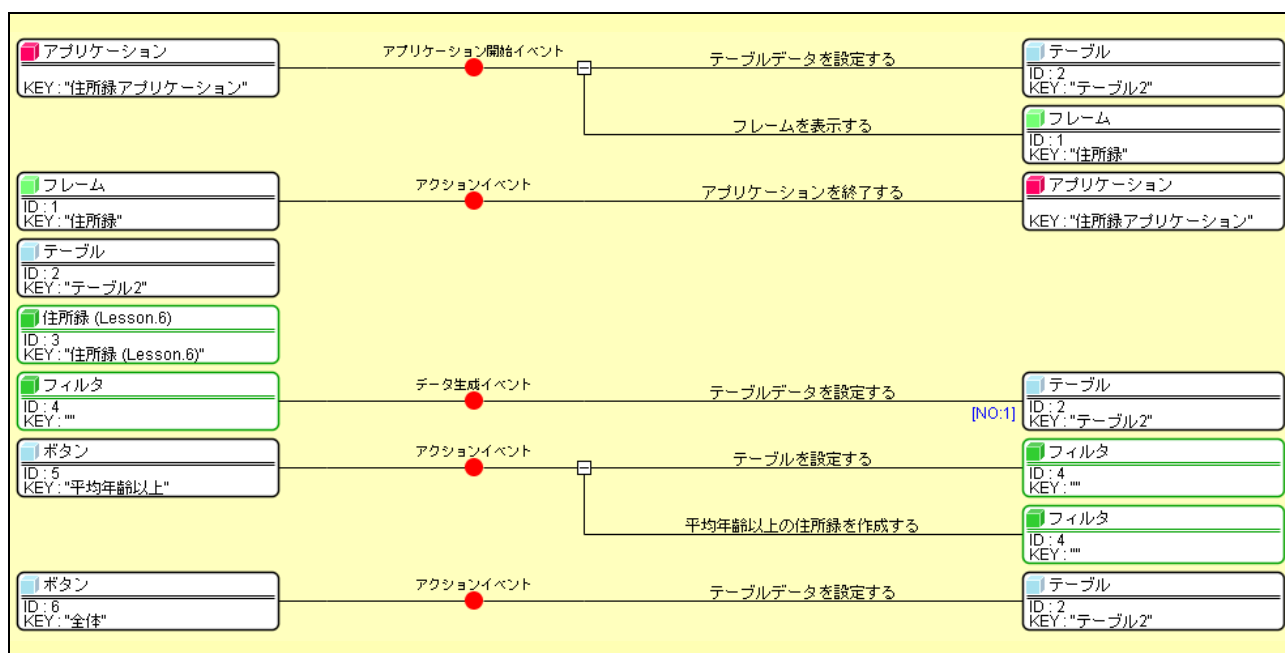


図 4 拡張したアプリケーションのコンポーネント接続関係 (その 1)

ここまでアプリケーションを作成してきて、『複合コンポーネント』、『サブルーチン』、『イベント生成』コンポーネントの使用方法が理解できたと思います。次のステップでは、『ファンクション』コンポーネントの使用方法を習得します。

### Step.3 複合コンポーネントの拡張（ファンクション）

次に、指定した年齢以上の住所録を生成する機能を複合コンポーネントであるフィルタ (ID:4) に追加してみましょう。この機能の実現には『ファンクション』コンポーネントを使用します。それでは、フィルタ複合コンポーネントの構築画面に移り、『ファンクション』コンポーネントを追加します。このコンポーネントのカテゴリは次のとおりです。

コンポーネント名	必要数	
■ ファンクション	1	[処理部品]－[サブルーチン]－[ファンクション]

『ファンクション』コンポーネントのメソッド『ファンクションの呼出し（0引数）()』、『ファンクションの呼出し（1引数）(Object)』等を実行すると、「処理要求イベント」が発生します。処理要求イベントは最後に接続されているメソッドの戻り値を取得する仕様になっており、この値は上述の実行したメソッド『ファンクションの呼出し(...)』の戻り値となります。また、接続先の処理では、上述のメソッドに引数として指定されたオブジェクトを取得し、処理に使用することができます。最初の引数のオブジェクトを取得する場合『第1引数の取得()』、二つ目を取得する場合『第2引数の取得()』、三つ目を取得する場合『第3引数の取得()』、を用います。今回の機能では『ファンクションの呼出し（1引数）(Object)』を使用し、引数のオブジェクトには指定年齢である整数が入力されると仮定します。

『ファンクションの呼出し（1引数）(Object)』が呼ばれた際の処理は、前のステップとほぼ同じですが、比較演算(≥) (ID:4-4)の右オペランドに、一つ目の引数である指定年齢を設定するところが異なります。また、生成されたテーブルデータを戻り値として取得可能なように、ファンクション(ID:4-7)の接続先の最後にテーブル格納変数(ID:4-2)のメソッド『テーブルを取得する()』を指定します。コンポーネント間の接続関係は次のようになります。

表 7 コンポーネント接続関係（指定年齢以上の住所録の作成）

項目	内容
イベント発生元コンポーネント	■ ファンクション (ID:4-7)
発生イベント	処理要求イベント
イベント番号	定常起動
接続先	①指定年齢を比較演算子に設定 ◇接続先コンポーネント ■ 比較演算 (≥) (ID:4-4) ◇起動メソッド 数値に変換して右オペランドに設定する (String) <引数：右オペランド> 取得方法：『メソッド戻り値』 取得先コンポーネント：ファンクション (ID:4-7) 取得メソッド：“第1引数の取得 ()”

②繰り返し処理(テーブルデータの最終行から開始行の繰り返し)

◇接続先コンポーネント

■繰り返し制御(FOR) (ID:4-5)

◇起動メソッド

繰り返し処理を実行する (int, boolean, int, boolean, int)

<引数: 開始値>

取得方法: 『メソッド戻り値』

取得先コンポーネント: テーブル格納変数 (ID:4-2)

取得メソッド: “行数を取得する”

<引数: 開始値を含めるかどうか>

取得方法: 『固定値』

値: “false”

<引数: 終了値>

取得方法: 『固定値』

値: “0”

<引数: 終了値を含めるかどうか>

取得方法: 『固定値』

値: “true”

<引数: 増減値>

取得方法: 『固定値』

値: “-1”

③生成されたテーブルデータを戻り値とする

◇接続先コンポーネント

■テーブル格納変数 (ID:4-2)

◇起動メソッド

テーブルを取得する()

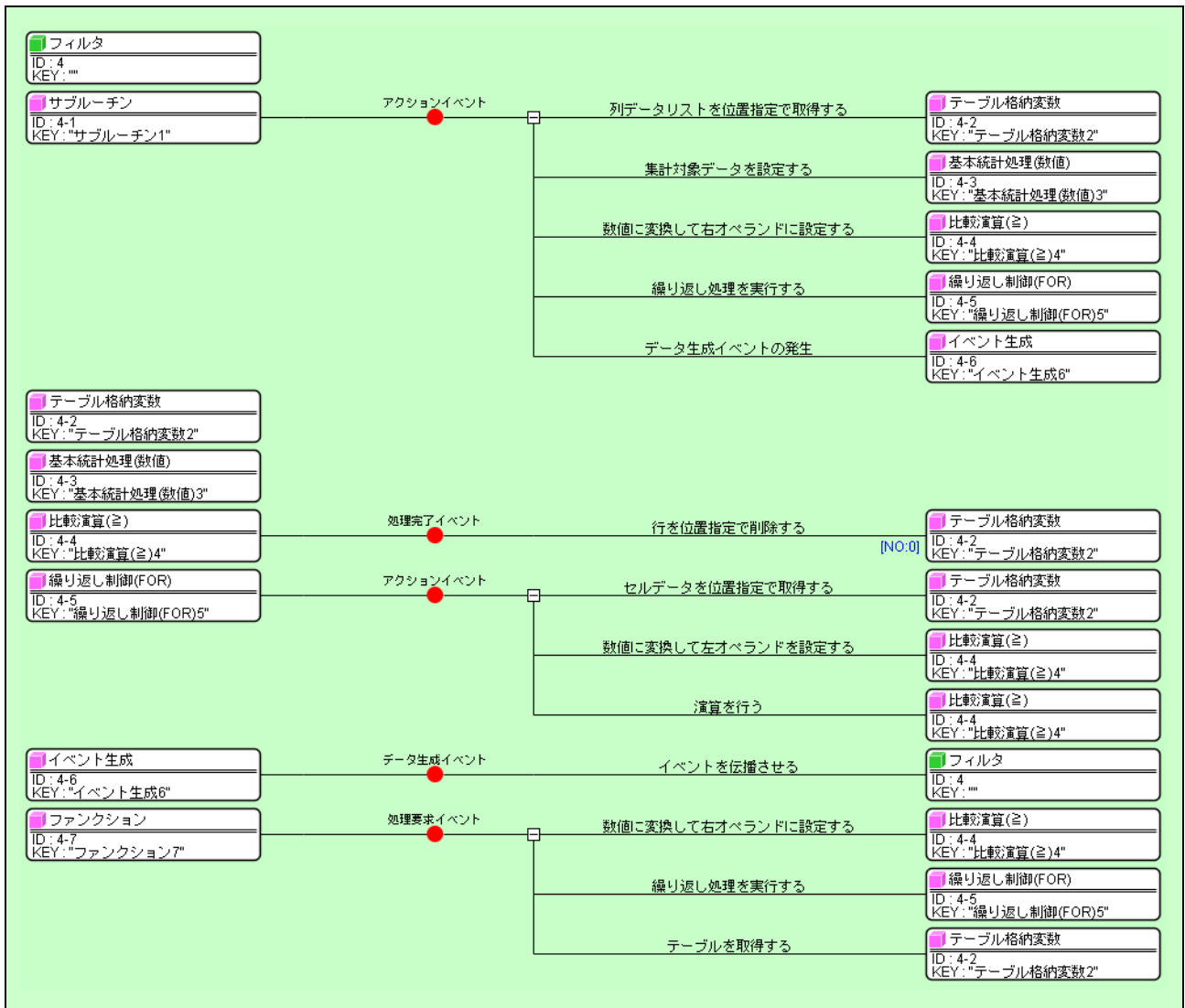



図 5 複合コンポーネントのコンポーネント接続関係 (その 2)

これらの接続で、ファンクション (ID:4-7) のメソッド『ファンクションの呼出し (1 引数) (Object)』が実行された際に、テーブルデータが取得できるようになりました。複合コンポーネントの外部からこのメソッドを呼び出し可能なように、公開メソッドを設定します。

表 8 複合コンポーネントの公開メソッド (その 2)

コンポーネント名	メソッド名
■ ファンクション (ID:4-2)	ファンクションの呼出し (1 引数) (Object)

また、メソッド名を「指定年齢以上の住所録を取得する」に変更します。それでは、右上の  を押して、アプリケーションのビルダー画面に移り、他のコンポーネントからこの公開メソッドを実行してみましょう。その前に、その指示を与える『ボタン』と、年齢を入れる『テキストフィールド』コンポーネントを追加します。次の図を参考にボタン (ID:7) とテキストフィールド (ID:8) をフレーム (ID:1) に配置し、ボタンのテキストに“指定年齢以上”と設定します。

氏名	住所	性別	年齢
山田太郎	茨城県つくば市並木1丁目	男	30
山田次郎	茨城県つくば市並木1丁目	男	28
山田三郎	茨城県つくば市並木1丁目	男	26
山田竹子	茨城県つくば市並木1丁目	女	24
山田梅子	茨城県つくば市並木1丁目	女	22
鈴木太郎	茨城県つくば市竹園1丁目	男	30
鈴木次郎	茨城県つくば市竹園1丁目	男	28
鈴木三郎	茨城県つくば市竹園1丁目	男	26
鈴木竹子	茨城県つくば市竹園1丁目	女	24
鈴木梅子	茨城県つくば市竹園1丁目	女	22
伊藤太郎	茨城県つくば市梅園1丁目	男	30
伊藤次郎	茨城県つくば市梅園1丁目	男	28
伊藤三郎	茨城県つくば市梅園1丁目	男	26
伊藤竹子	茨城県つくば市梅園1丁目	女	24
伊藤梅子	茨城県つくば市梅園1丁目	女	22
佐藤太郎	茨城県つくば市千現1丁目	男	30
佐藤次郎	茨城県つくば市千現1丁目	男	28

平均年齢以上    指定年齢以上    28    全体

図 6 拡張したアプリケーションの画面イメージ (その 3)

[指定年齢以上]ボタンが押された際にテキストフィールド(ID:8)に記述された年齢を取得して、フィルタ(ID:4)の上述の公開メソッドを実行し、指定年齢以上の住所録のテーブルデータをテーブル(ID:2)に設定するように、次のようにコンポーネント間を接続します。

表 9 コンポーネント接続関係（テーブルデータ生成と再設定）（その 2）

項目	内容
イベント発生元コンポーネント	■ ボタン (ID:7)
発生イベント	アクションイベント
イベント番号	定常起動
接続先	<p>①テーブルデータを設定する</p> <p>◇接続先コンポーネント</p> <p>■ フィルタ (ID:4)</p> <p>◇起動メソッド</p> <p>テーブルを設定する (PFObjectTable)</p> <p>&lt;引数: 設定するテーブル&gt;</p> <p>取得方法: 『メソッド戻り値』</p> <p>取得先コンポーネント: テーブル (ID:2)</p> <p>取得メソッド: “テーブルデータを取得する ()”</p> <hr/> <p>②指定年齢以上の住所録を生成し、取得する</p> <p>◇接続先コンポーネント</p> <p>■ フィルタ (ID:4)</p> <p>◇起動メソッド</p> <p>指定年齢以上の住所録を取得する (Object)</p> <p>&lt;引数: 指定年齢&gt;</p> <p>取得方法: 『メソッド戻り値』</p> <p>取得先コンポーネント: テキストフィールド (ID:8)</p> <p>取得メソッド: “テキストを取得する ()”</p> <hr/> <p>③取得した住所録をテーブルに設定する</p> <p>◇接続先コンポーネント</p> <p>■ テーブル (ID:2)</p> <p>◇起動メソッド</p> <p>テーブルデータを設定する (PFObjectTable)</p> <p>&lt;引数: 指定年齢&gt;</p> <p>取得方法: 『メソッド処理結果』</p> <p>取得メソッド: “指定年齢以上の住所録を取得する”</p>

この時点でアプリケーションを実行してみましょう。指定年齢以上の住所録が表示されましたか？うまく表示されない場合には、もう一度コンポーネント間の接続を確認してみてください。参考まで、ここまでのコンポーネント間の接続の様子を示します。

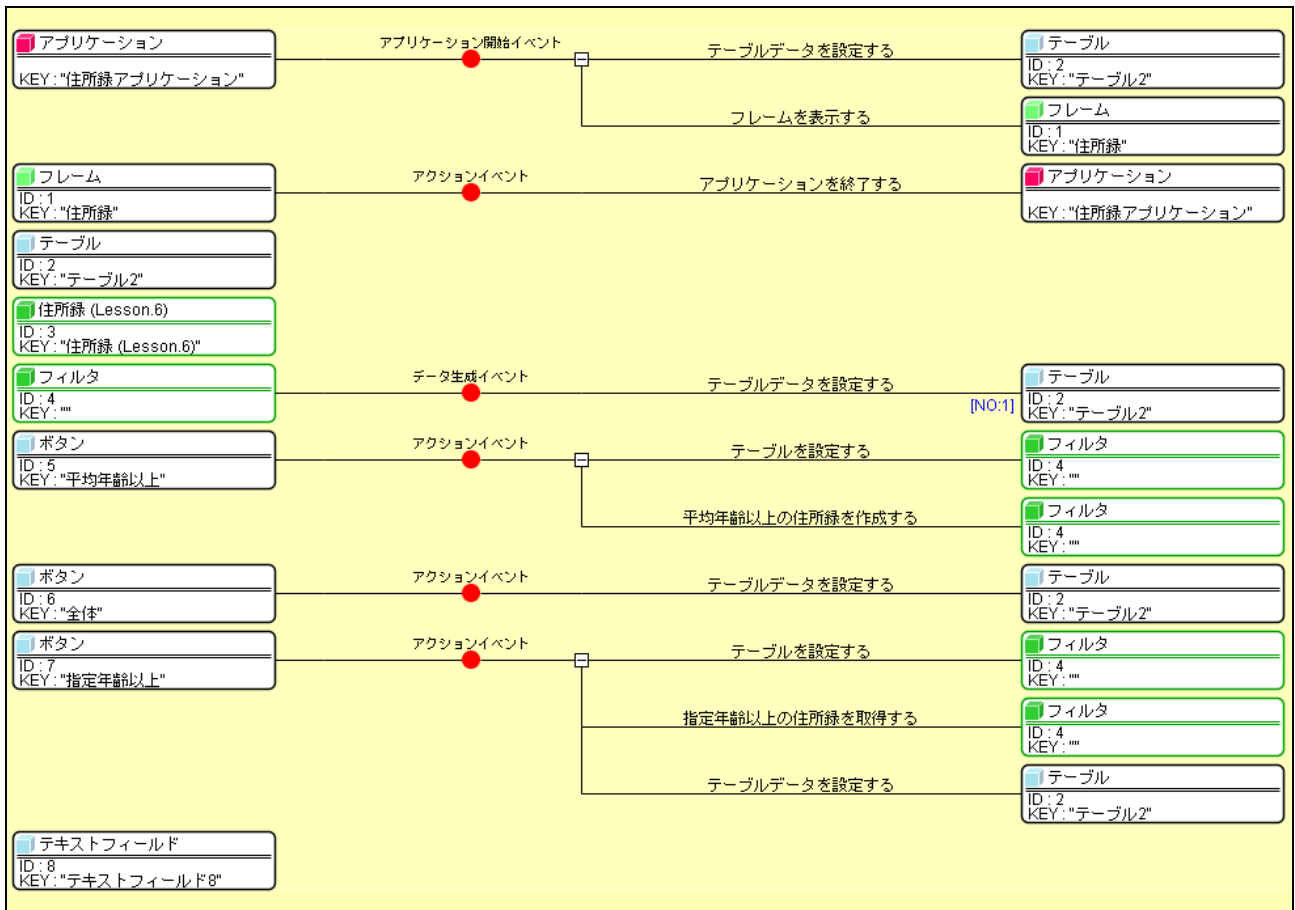


図 7 拡張したアプリケーションのコンポーネント接続関係 (その 2)

このレッスンを通して、処理をまとめる方法として、『複合コンポーネント』だけではなく、『サブルーチン』『イベント生成』『ファンクション』の使用方法、相異が理解できたと思います。これらを用いることにより、格段にコンポーネントの接続関係、処理の流れが理解しやすくなったはずです。