AiSee の Windows アプリ作成マニュアル 【基本編】

第1版

株式会社ベータ・ネット

作成者:石毛、大山、増子、伊藤

作成日:2022/10/28

[目次]

1	この資料の目的	1
2	使用するアプリケーション	1
\$	2.1 AiSee の画面全体の説明	1
3	AiSee を使った Windows アプリ開発の全体像	4
ę	3.1 サンプルアプリの完成図	6
ę	3.2 このアプリの仕組み	8
4	AiSee Windows アプリの作成手順	9
4	1.1 環境設置手順	9
4	1.2 画面作成	9
4	I.3 アプリの作成手順	10
	4.3.1 プロジェクトファイルの作成	10
5	プロジェクト内容の作成手順	14
ł	5.1 シナリオの作成	14
ł	5.2 イベント定義の指定	17
ł	5.3 イベント毎のシナリオの内容サンプル	22
	5.3.1 CalculationTable_初期化 とテスト実行	22
	5.3.2 Calculation_初期化	29
	5.3.3 CalculationTable_Add	33
	5.3.4 Calculation_Add	34
	5.3.5 Calculation_Close	44
	5.3.6 CalculationTable_Del	53
	5.3.7 CalculationTable_Edit	55
	5.3.8 【共通】合計処理	72
	5.3.9 合計処理	89
	5.3.10 平均処理	93
	5.3.11 最大值処理	100
	5.3.12 最小值処理	113
ł	5.4 データベースについて	122
	5.4.1 AiSee における DB の使用方法	122
	5.4.2 DB を使ったシナリオ作成	123
	5.4.3 DB を作成したい場合(DB Browser)	139
6	NetBeans でアプリの画面を作成する方法	141
7	既存プロジェクトをインポートする方法	156
8	繰り返し(ループ)処理の説明	157
9	便利な Tips	160
10	Q&A	163

改訂履歴

版数	ページ番号	内容	作成日/改定日	作成者/改訂者
1			2022/10/28	

1 この資料の目的

この資料では、初めて AiSee に触れる方を想定し、AiSee の基本的な操作方法を紹介いたします。

実際の作成例を通して、Windows アプリケーションを作成するための手順を中心に記載しています。

2 使用するアプリケーション

必要なアプリケーション	用途
AiSee	AiSee の Windows アプリ開発ツール (IDE)
NetBeans	画面表示用 Jar ファイル作成のために使用するアプリ

2.1 AiSee の画面全体の説明

まずは、これから操作することになる AiSee の画面の全体像を紹介します。 細かい内容は次章以降、順番に説明していきますので、現状は理解ができなくても問題ありません。 そのままアプリ作成に進んでください。



①プロジェクトフォルダ

アプリごとに作成されるフォルダです。図には複数のプロジェクトのファイルが表示されています。

AiSee では何種類ものアプリを同時作成したり、内容を他のアプリに流用したりする事も可能です。

②画面定義

画面イメージが格納されます。

③シナリオ定義

ボタンが押された時や、アプリが表示された時に何が起こるか等、実際の処理内容が記載される部分

です。アプリが動く仕組みがここで記述されるため、この部分の作成が作業の大きなウェイトを占めます。

④メニュー定義

「右クリックで画面にはないメニューが表示される」といった設定を任意で加えることができます。

⑤イベント割当定義

「どのボタンを押すとどの処理が発動するか」など処理をボタンなどの部品に紐づけます。

⑥ライブラリ定義

別ソフトで作成した画面イメージのファイルである jar ファイル (OOOjar) を登録します。

⑦リソース定義

データベースを使うアプリを作る場合のみ、データベースのファイル(OOO.db)を登録します。

3 AiSee を使った Windows アプリ開発の全体像

アプリの開発から確認まで、全ての作業フェーズを紹介します。

開発環境設置

● 開発環境及びソフトをインストールする

AiSee ウインドウズの開発ツールをインストールする

NetBeans 及び JDK をインストールする

※NetBeans のインストールについては本資料では省略します。



アプリの画面設計作業

NetBeans を利用して、アプリの画面を作成する

※本資料では省略、アプリの画面は事前準備されています。



アプリの開発作業

AiSee のウインドウズ開発ツール(IDE)で、下記の手順を行う

- プロジェクトを作成する
- アプリ画面の Jar ファイルを登録する
- シナリオを作成する
- イベントを定義する

● メニューを定義する



デバッグ

- 起動設定を行う
- テスト実行して画面動作を確認する



Windows アプリの完成

3.1 サンプルアプリの完成図

この手順書では、下図の「Calculation」と「CalculationTable」の2つの画面を使用し 四則演算と、計算データを表へ格納する処理を行うアプリを作成します。

① 実行すると、「CalculationTable」が現れる (まだ数字等のデータがない、空の状態)

②「CalculationTable」の追加ボタンを押すと「Calculation」が現れる

③ 「Calcuation」の計算式の部分に数字を入力し、計算ボタンを押すと、下に計算結果が表示される

< Calculation Table>

#§ CalculationTa	able	-	
		追加 修	正 削除
1項	演算子	2 項	結果
1600	*	8	12800
2300	*	8	18400
1500	*	6	9000
1750	*	8	14000

< Calculation >

👭 Calcu	lation		-		\times
:	1750	*		8 計算	閉じる
14000					

【このアプリの仕組み】

①「CalculationTable」が現れる (まだ数字等のデータがない、空の状態)

②「CalculationTable」の追加ボタンを押すと「Calculation」が現れる

- ③ 「Calcuation」の計算式の部分に数字を入力し、計算ボタンを押すと、下に計算結果が表示される
- ④ 「Calculation」の閉じるボタンを押すと、「Calculation」に入力してあった数字、演算子、計算結果が

「CalculationTable」の方に追加され、表示される

※上記が完了しましたら、修正ボタンや削除ボタンへの機能の追加もお試しください。

4 AiSee Windows アプリの作成手順

4.1 環境設置手順

① AiSeeIDE をインストールする

② NetBeans をインストールする

※今回は作成済みの Jar ファイルをダウンロードしていただくため、不要です。

4.2 画面作成

用意されたサンプルアプリ画面のファイル(Jar ファイル)を、以下よりダウンロードしてください。

Jar ファイルのダウンロード

※ここではアプリの画面の作成や NetBeans についての詳細説明は行いません。

尚、NetBeans という別のソフトを使用してアプリの画面を作成する場合の操作については下記リンクを参照し

てください。

NetBeans でアプリの画面を作成する方法

4.3 アプリの作成手順

Windows アプリを作成する手順を紹介します。

4.3.1 プロジェクトファイルの作成

まずは、AiSeeIDE(以下 AiSee)を起動してください。



次にプロジェクト名を決定・入力し、作業スペースを作成します。

この段階で入力した「プロジェクト名称」は後から変更ができないため、慎重に検討されることをおすすめします。

(1)新しいプロジェクトの作成手順【図1】

①「ISeeProjectInfomation」右クリック

②プロジェクト名「AiSeeSample」を入力

③「確定」をクリック

【図1】



(2)ライブラリ定義【図2】 ①「ライブラリ定義」を右クリックし「ライブラリ登録」をクリック ②ダウンロードしたファイル「sampleLib.jar」を選択 ③「開く」ボタンクリック ライブラリ定義直下に「 sampleLib.jar 」が反映されていれば OK

【図 2】

 □ 画面定義 □ シナリオ定義 □ メニュー定義 □ イベント割当定義 ↑ □ ルール定義 □ MonadicRule 	2 samp	oleLib.jar				
1 - <u>ライブラリ定義</u> - ロリンテス定義	sampleLibjar	O	2022/09/14 10:19	Executable Jar Fi		
	< i(N): sampleLib.jar		3	> ~		

開く(O)

キャンセル

(3)画面の登録【図3~図4】

- ①「画面定義」を右クリック、「画面読み込み」クリック
- ②「sampleLib.jar」を選択して、右下「選択」ボタンを押下
- ③ 右下「実行」ボタン押下→「OK」押下
- ④「画面定義」直下に「Calculation」と「CalculationTable」が追加されていればOK

【図 3】



【図4】



5 プロジェクト内容の作成手順

ここからは、「AiSeeSample」というプロジェクトの内容を作成していきます。

もし作成中に困った事があった場合、Q&A に解決方法の記載がないかご確認ください。

5.1 シナリオの作成

「シナリオ」とは処理のことです。
シナリオ作成によって、計算する、変数を作る等、多種多様なアプリの動作を作っていきます。
1 つのシナリオの内部に、足し算や引き算、ループや条件付け分岐(もし○○なら△△)など
具体的処理内容が記載できます。複数のシナリオを組み合わせて、目的の処理全体が完了するよう作成します。
例えば、画面を初期化するシナリオであれば、アプリが開くときの画面のサイズを規定したり、
テキストフィールドを空の状態にリセットしておくといった処理を1つのシナリオの内部に複数記載します。

ここではまず、シナリオの「項目名」のみを先に作成します。

作成する内容は、以下の画像の通りです。

※シナリオの並び順は、処理の順番ではないため、この通りにならなくとも問題はありません。



シナリオの「内容」の作成手順については下記リンク先を参照してください。

イベント毎のシナリオの作成

シナリオ作成【図 5】

「シナリオ定義」右クリックして「シナリオ新規作成」を押下→シナリオ名を入力

「CalculationTable_初期化」とし「OK」ボタンを押下

【図 5】



5.2 イベント定義の指定



ここでは「Calculation の計算ボタンを押すと Calculation_Add が実行される」設定、

「Calculation の閉じるボタンを押すと Calculation_Close が実行される」設定、

「CalculationTable の追加ボタンを押すと Calculation が表示される」etc...の設定を行います。

下記の画像の通りに作成します。



イベント定義作成

「イベント割当定義」を右クリックして「イベント定義作成」

「イベント名」を入力して「OK」ボタン



【図7】

.

AS ISAD_ScreenAc	tionManager	_		×
画面ID:	(3) 🔛 🗹 初期割当 イベント	-	追加	削除
画面イメージ表示	画面部品一覧	ドント]	
	All ISAD_ISeeResourcesChooser ×	イベ	ント定義	有効
	リソースの選択 ■ 画面定義 ■ Calculation ■ Calculation as			
	 		生成	閉じる

E_CalculationTable を作成する際は①で「**CalculationTable**」を選択してください。

【図 8】Calculation の設定

☆ ISAD_ScreenActionManager 画面10: Calculation ■ 図 23集示 画面部品一覧 計算式: ■ 計算 開ごる	- □ X (ベント▼) 追加 神路 第84(ベント▼) 直面(ベント (面面)588-6 イベント定義 有効 cat_bh 快程実行 → → → → → → → → → → → → → → → → → → →	1 都商が品名 Calc_bin Cose_bin 別 別	☆ ISAD_ISeeResourcesChooser リンースの資料 シナリオ定義 CalculationTable_20#月化 CalculationTable_20d CalculationTable_2dd CalculationTable_Edit CalculationTable_Edit CalculationTable_Cole CalculationTable_Cole CalculationTable_Cole CalculationTable_Cole CalculationTable_Cole CalculationTable_Cole Calculation_Cole

CalculationTable_Add の章から来た場合は<u>こちら</u>から戻れます。





CalculationTable_Close の章から来た場合は<u>こちら</u>から戻れます。

同じ手順で「E_CalculationTable」を作成してください。

【図 10】CalculationTableの設定

As ISAD_ScreenActionManager	- 🗆 X
画面D: CalculationTable	イベント 🔻 追加 削除
画面イメージ表示 画面部品一覧 部品イベント	● 画面イベント
道加 遊加 修正 削除 edit calc btn edit calc btn	
	<u>Ⅳ</u> 2.柱关11
	//++

【図 11】

	部品イベント 画面イベン	F		As ISAD_ISeeResourcesChoo	ser
	画面部品名	イベント定義	有効	リソースの選択	i
① (odit_colo_bto add_calc_btn dot_calc_btn	处理定行 处理実行 加速失行		 シナリオ定義 ③ CalculationTable 初期化 ③ CalculationTable Add ③ CalculationTable Edit ⑤ CalculationTable Edit ⑤ Calculation 初期化 ○ Calculation 初期化 ○ Calculation Close 	
	画面ID : CalculationTable 画面部品名: add_calc_btn イベント : 処理実行 イベント定義 処理シナリオ: Calculation_	e 約期化			
4	 □ビ 画面遷移あり 遷移元の画面状態 遷移先画面のクラスパス: 遷移先の画面状態: 	フォーカス可 com.samplelib.screen.Calculatio 新規表示 ▼		ビリン-ス □ フォルダ 「面面定素 Calculation CalculationTable	後索 ♥ 紋込み 選択 閉じる
		<u>6</u> 生成	閉じる		

Calculation_初期化の章から来た場合は<u>こちら</u>から戻れます。

部品イベント 画面イベント	AS ISAD_ISeeResourcesChooser
画面部品名 イベント定義 有効 edit_calc_btn 処理実行 ピ add_calc_btn 処理実行 ピ del_calc_btn 処理実行 ピ ①	UV-スの選択 Structure Structure Structure Structure Structure Structure Structure Structure Structure Structure Structure Structure Structure Structure Structure Structure
画面D : CalculationTable 画面部品名: del_calc_btn イベント : 処理実行 … イベント定義 処理シナリオ: CalculationTable_Del ② … 画面速移あり	- 検索 ⊻ 絞込み ビリソース □ フォルダ ④ 選択 閉じる

【図 12】

CalculationTable_Del の章から来た場合は<u>こちら</u>から戻れます。

【図 ′	13】
------	-----

				AS ISAD_ISeeResourcesChooser	
		イベント 🔻 追加	削除	リソースの選択	
	部品イベント 画面イベント			□ シナリオ定義 □ □ □ CalculationTable 初期化	
	画面部品名	イベント定義	有効	- CalculationTable_Add	
	add_calc_btn	24-		- ColoulationTable Edit	
	edit_calc_btn 処理実	[行	V)
(1)	del_calc_btn 即理	ĘĨT		- 🗟 Calculation_Add	
	画面ID : CalculationTable 画面部品名 : edit_calc_btn イベント : 処理実行			- 급 Calculation_Close - 급 (土満급) 合計処理 - 급 슈計処理 - 급 最大値処理 - 급 最大値処理 - 급 最小値処理	
	イベント定義				検索 ビ 絞込み
	処理シナリオ: CalculationTable_Edi 「ビ 画面逆移あり	t (2)			選択 閉じる
	遷移元の画面状態 : フォーク			AS ISAD_ISeeResources	Chooser
	遷移先画面のクラスパス: com.sar	nplelib.screen.Calculation		リソースの選択	
	遭移先の画面状態 : 新規表示			自面定義 C <mark>Calculation</mark> Calculation Table	
		生成	閉じる		

CalculationTable_Edit の章から来た場合は<u>こちら</u>から戻れます。

5.3 イベント毎のシナリオの内容サンプル

ここからは、1つ1つのシナリオの内部に、さらに具体的な処理を記述していきます。 シナリオの内部に、様々な種類の「リレーション」や「代入処理」などを作成しますが、 これらは上から下へ順番に処理されていくため、並び順にも注意しながら作成してください。 ※並び順の変更:作成した処理を右クリックすると下や上へ移動できます。

5.3.1 CalculationTable_初期化 とテスト実行

「CalculationTable」のデフォルトで作成されているテーブルを削除する実装

AS ISAD_Scree	enlmageViewe	er —			AS Calculatio	onTable	_	
		追加(を正 削除				追加(修正 削除
1項	演算子	2項	結果		1項	演算子	2 項	結果
			I					
I				1				

<u>手順1</u>:テーブルすべての行を削除する

<u>手順2</u>:テスト実行

手順1:テーブルすべての行を削除する【図14~図17】

【図 14】



「CalculationTable_初期化」右クリックで「修正」選択

【図 15】





「テーブルのすべての行を削除する」を選択。

ない場合は左下の検索欄で「テーブルのすべての行を削除する」を入力して検索ボタン。

【図 16】



①パラメータ定義の「画面部品を指定」を右クリック

パラメータ定義で入れるものは「パラメータ説明」のタブをクリックすると確認できます。 リレーションによってパラメータ定義で選択する内容が変わるので都度確認をお願いいたします。

【図 17】

ℳ リレーションマネージャ:CalculationTable_初期化			-		×
	[DL一覧追加			
代入 Section	1	シナリオ説明] 変調	款定 義	
	00000	変数名		種類	
	10000				
	00000				
	00000				
	10000				
	00000				
	10000				
	00000				
	10000				
	00000				
	10000				
	0.000				
	10000	タイトル:			-
	10000		1	株)ペータ	ネット
	10000		/. 🤇	Ser	
	10000	Copyright(c) 200	Betanet Cor	rparation. All Rights I	Reserv.
[I			Laboration		881. 7
			唯正	1717	開しる

手順2:テスト実行

1で行った初期化が反映されているかの確認を行います。

テスト実行は適宜行いエラーや反映漏れが無いか確認をしましょう。

以下【図 17~図 18】の起動設定が終わったあと、F5 キーを押すとテスト実行されます。

プロジェクトフォルダ「AiSeeSample」の直下フォルダ内のどこかを選択した状態で F5 キーを押して下さい。

【図 18】



「AiSeeSample」を右クリックで「起動設定」を選択

【図 19】



【図 20】

AS ISAD_Scree	enImageViewe	er —		_	AS Calculatio	nTable	_	
追加 修正 削除							追加的	『正 削除
1 項	演算子	2項	結果		1 項	演算子	2項	結果

テーブルの表示が消えていれば OK です

5.3.2 Calculation_初期化

Calculation 画面の1項・2項の欄(オレンジ枠)を起動時に空の状態で起動したいので下記の手順で定義する。

まずはイベント割当定義が出来ているか確認お願いします。

計算式:		+	•			
				計算	閉じる	,
		 				7.57.5×

手順1:1項・2項の欄を起動時に空の状態にする

「Calculation_初期化」を右クリックで「修正」

【図 21】



【図 22】



【図 23】



【図 24】

AS ISAD_Scenario	_		×			
タイトル : 複	数の代入処理を	行う				
説明・コメント:						
		(1)			
		左辺指定	右辺指定	追加i	削除 1	1
代入先	種類		代7	vī.	1	<u>重</u> 類
		<pre><=_</pre>		画面部品 定数の指 ローカル グローバ ISee関数 サブシキ オから割	の 指定 定 変数 エデル モデル ロード ゴー マび出	から指定
			2	NULLを EMPTY TRUEを FALSEを	指定 指定 指定 指定 指定	

EMPTY = 空 という意味です。

【図 25】

AS ISAD_ScenarioCellSubstitutionMulti	- 🗆 X	🗥 ISAD_ScenarioCellSubstitutionMulti – 🗆 🗙
タイトル : 複数の代入処理を行う 説明・コメント:		タイトル : 説明・コメント:
	定 右辺指定 道加 削除 ↑ ↓ 代入元 種類	左辺指定 右辺指定 追加 削除 ↑ ↓
Calculation/first_tdf/value 画面モデル = EMP	TY ISee定数	Calculation/first_tof/Val 画面モデル Calculation/second_tが画面モデル Calculation/second_tが画面モデル
通面節曲の指定 定数の指定 ローカル変数 グローバル変数 グローバル変数 ユーザーモデル ISee関数・共通DLから指定 サブシナリオ呼び出し	 	

ここで画面イメージから設定で選択するコンポーネントで命名されている「oo_txf」などは NetBeans で画面を作成時に作った名前が反映されています。
5.3.3 CalculationTable_Add

「サブシナリオ」機能を使ってリレーションを使いまわす方法。

手順1:サブシナリオで「Calculation_初期化」を実行するを設定

【図 26】



【図 27】



5.3.4 Calculation_Add

演算子の分岐設定の実装

まずはイベント割当定義が出来ているか確認お願いします。

<u>手順1</u>:分岐処理を行う

<u>手順2</u>:計算をする

手順3:計算結果を表示する

<u>手順4</u>:テスト実行

手順1:分岐処理を行う

計算の演算子の分岐点を作成します。

「+」「-」「*」「/」だったら~までを作成します。

③「条件追加」を4回押すとこの状態になります。

【図 28】



【図 29】

	All ISAD_ISeeResourcesChooser ×	
★☆ ISAD_FormulaSimpleCondition ー × 条件式入力 左辺指定 追加 卵除 ↑ 左辺 種類 - - - 室辺 種類 - - - 確定 閉じる	リソースの選択 ■ 画定表 ■ dacuation 3 Calculation 1 as	★SISAD_ScreenComponentSelecting × データモデル定義から設定 TECL Market ComponentSelecting 本 日本 1 年間 TECL Market ComponentSelecting 本 本前 イメージから設定 TECL Market ComponentSelecting 本 本前 イメージから設定 日本 1 年間 TECL Market ComponentSelecting 本 本前 イメージから設定 日本 1 年間 日本 1 年間 日
All ISAD_FormulaSimpleCondition - 🗆 ×		
条件式入力 左辺指定 右辺指定 追加 削除 ↑ ↓ 左辺 種類 石辺 種類 ・ ・ 2 画面部品の指定 ー ー ・ ・ ローカル変数 確定 閉じる		様在設定 類ドインテックス在設定 リストモデルを設定 1/2スタンスを設定 設定 ■面面減快 取定 酸にる

※この時「画面イメージから設定」直下の「設定」は押さないようにしてください

【図 30】

AS ISAD_FormulaSim	pleCondition (1) -						
条件式入力	左辺指定 右辺指定 追加	削除 ↑ ↓					
左辺 在類 Calculation/o 画面モデル	i EQ ▼ 右辺	種類					
		前部品の指定					
	2	gの指定 - カル変数					
4 ケース4		コーバル変数					
	_	-ザーモデル					
ŕ							
	AS ISAD_InputLiteralInfo	×					
1	+	参照	AS ISAD_FormulaSimple	eCondition	_		\times
		シナリオのパス 出力先ファイルのパス	条件式入力	左辺指定 右辺	2指定 追加	削除 1	· ↓
		入力元ファイルのパス ライブラリのパス	左辺 種類 Calculation/o 画面モデル	EQ ▼	右辺 +	種類	
		設定閉じる				確定	閉じる
L							

「-」「*」「/」も同様の操作で設定してください。

【図 31】

AS ISAD_Scenario	CellCase	_	- 🗆	×
タイトル : <u>分</u> 説明・コメント:	<u> 坂処理を行う</u>			
分岐ケース指定	条件追加	条件編集	コメント	条件削除
羅列順番 現在順番		分岐条件	:	
1 ケース1	Calculation/operator	_cmb/Value	EQ +	
2 ケース2	Calculation/operator	_cmb/Value	EQ -	
3 ケース3	Calculation/operator	_cmb/Value	EQ *	
4 ケース4	Calculation/operator	_cmb/Value I	EQ/	
<u>*</u> .				
	使用する ケースの乳	【行条件: (● 腓他実行	○ <u>冬重</u> 実行 確定 戻る



手順2:計算をする

画像内ではケース1の足し算の設定方法を行っています。

「+」「-」「*」「/」だったら~どうするを作成します。

ケース2~4も同様に操作を行います。出来上がると【図 32】のようになります。

選択するリレーションは下記です。

ケース1「足し算(加算)を行って、計算結果を返す」

ケース2「引き算(減算)を行う関数」

ケース3「掛け算(乗算)を行い、結果を返す」

ケース4「割り算(除算)を行い、計算結果を返す」

ローカル変数「計算結果」は未作成なので作成してください。【図35】

【図 32】

ዮ 🗔 分岐処理を行う	
- D ケース1:Crissian (second cmb/Value EQ +	分類階層表示 ■ 履歴
- □ ケース3:C4 新規作成 ▶ 選択した種類	- D.代入処理: 代入先は、変数や画面モデルやデータモー <u>ロビロン 4.00月/1012</u> (1)3000 日月40本で送り
- ウース4:C t 貼り付け 代入	- □ Thotily / Ferravo CES9 - □ ISeeリストに要素を追加する - □ アータベースレコードのカラムへ内容をセットを - □ 選択中テーブルの行を、行モデルで上書きする
上へ移動 リレーション	□ リストモナルから指定したコンテックスの要素と取得 □ マップを作成して返す。初期値の指定が可能 □ PSマップを作成して返す。初期値の指定が可能 □ PSマップを作成して返す。オートはない、トスセートオート
下へ移動 (」) サブシナリオ	□ Steeマップから、キーと対応したパリューを取得して返 □ ISeeマップから、キーをもとにバリューを取得して返 □ データベーステーブルのレコードを更新(Update)する
有効・無効の切替分岐	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
ループ	ランキング
ブロック	□ ランキング □ ブータベースレコードのカラムから内容を取得して返す
中断	− □ データベースレコードのカラムへ内容をセットする − □ テーブルのすべての行を削除する
リターン	- □ テーブル行モデルを作成して返す - □ 足し算(加算)を行って、計算結果を返す
共通データリンク	□ テーブルモデルのハンドル(テーブル行モデルのハンドル)を取得して返す □ □ 行干デルの内容をテーブルへ追加する

【図 33】



【図 34】



【図 35】

		AS ISAD_SelectLocalVariable	- 🗆 ×
<u>48</u> シナリオリレーション -		ローカル変数一覧	E
タイトル: 足し算(加算)を行って、計算結果を返す	処理選択	N0 51版のダイフ 200名	22 FUAMUE
説明・コメント:			
戻り値:ローカル変数 🔻	選択		
n - v .		追加 修正 削除 ↑ ↓ ■ 画面を常に表示	選択開じる
All ISAD_VariableInfoDefine ×			
名前 :計管結果		A ISAD_SelectLocalVariable	- 🗆 🗙
說明:		ローカル実験一覧 No 21時のタイゴ 実物名	20 2112日(市
		1 値渡し 計算結果	
利・「「奈小田」 「「四一」			
■ 「「「」」」「「「「「」」」」「「「「」」」」「「「」」」「「「」」」「「」」」「「」」」「「」」」「「」」」「「」」」「「」」」「」」「」」「」」「」」「」」」「」」」「」」」「」」」「」」」「」」」			
			jitte Burg

「パラメータとして使用する」にチェックを入れてください。

【図 36】

AS シナリオリレーション		- 🗆 X	P 🚍 Section
タイトル: 足し算(加算)を行って、計算結果。 説明・コメント:	<u>を返す</u>	処理選択	トロラ からします かいしょう やいしょう かいしょう かいしょう やいしょう かいしょう かい かいしょう かいしょう かい かいしょう かい かいしょう かい
			└──」足し算 ┣──二 ケース2:Calculation/operator
戻り値: ローカル変数 ▼ 計算結果		選択	
パラメーター定義	追加削除↑↓	•	ー L 掛け算
No 種類 f 1 画面モデル Calculation/first_bd/Value 2 画面モデル Calculation/second_bd/Value			← □ ケース4: Calculation/operator └ □ 割り算
数値などの足し算(加算)を行う関数。			
ID : additionFunction			
		登錄 戻る	

cmb/Value EQ +

cmb/Value EQ cmb/Value EQ *

cmb/Value EQ /

手順3:計算結果を表示する

【図 37】



ケース2以降のローカル変数はケース1のときに作ったものをそのまま使用してください。

ここで F5 キーを押し、 <u>テスト実行</u>を行いましょう。

問題なければ下図のように計算結果が表示されるようになります。

AS Calculation	—		×
計算式: 4	+ •] 3 計算 [閉	3 じる
7			

5.3.5 Calculation_Close

Calculation で計算した内容を Calculation Table に反映させる実装を行います。

まずはイベント割当定義が出来ているか確認お願いします。

<u>手順1</u>: 行モデルを作成する

<u>手順2</u>:「1項」「演算子」「2項」「結果」の内容をカラムへセットする

<u>手順3</u>:「行モデル」の内容をテーブルへ追加する

<u>手順4</u>:テスト実行

手順1:行モデルを作成する

「戻り値」の設定が必要になります。

戻り値とは呼び出した結果を保管する箱のようなものです。





【図 40】

	A8 シナリオリレーション	– 🗆 ×
48 ISAD VariableInfoDefine X	タイトル: テーブル行モデルを作成して返す	処理選択
名前: 行モデル		(1)
	戻り値: ローカル変数 ▼ 行モデル	jær
	パラスイター定義 通知 削除 1 4 No 種類 価値 1 画面モデル CalculationTable/calc_tbl	<u> </u>
型 : 任意の型 初期値 :		
□パラメータとして使用する ⑧ 値渡し ○ 😡 🎯 開じる	「機能説明」「バラメータ説明」 テーブル行モデルを作成する関数。戻り値で作成したモデルを戻す。 ID: createNewISeeTableRowModel_F	
	-	
		レを作成して返す

シナリオリレーション内①の「選択」をクリックして、「行モデル」を入力して「確定」をクリックしてください。

手順2:「1項」「演算子」「2項」「結果」の内容をカラムヘセットする

「演算子」「2項」「結果」は同じ手順で作成します。

下記は「1項」「演算子」の設定操作手順です。

「2 項」、「結果」に関しては【図 41】⑤の選択位置を「2 項」と「結果」で選択位置を変更してください。

コピー機能を使って修正を行うと短時間での作業が出来て効率的です。【図 43~45】

【図 41】



【図 42】



【図 43】



【図 44】



【図 45】



【図 46】

<u>48</u> シナリオリレーショ	ン						-		×	
タイトル: 🗅	實算子」の内容をカラムヘセットする	3						処	理選択	
説明・コメント:デー	- タベースレコードのカラムへ内容?	をセット	する							
パラメーター定義		追加	削除	1	Ţ	定数指定				Ŧ
No 種類	値					定数入力		▼	適用	8
1 ローカル変数 2 定数 3 画面モデル	行モデル 演算子 Calculation/operator_cmb					1項				
「螺能説明」パラメ パラメータ 1:値をセットするラ 2:値をセットしたし 3:セットする値	ー タ説明 ニーブルの行モデル いカラム名									
								28	れ 戻	3



手順3:「行モデル」の内容をテーブルへ追加する

【図 47】



【図 48】



【図 49】

No	種類			値
1	画面モデル	Ca	CalculationTable/calc_tbl	
3			画面部品の指定 定数の指定	-
			ローカル変数 グローバル変数	

パラ	メーター定義		追加
No	種類	値	
1	画面モデル	CalculationTable/calc_tbl	
2	変数	行モデル	
3			

P
│ │ │ テーブル行モデルを作成して返す
│ │ │ │ 「1項」の内容をカラムへセットする
│ │ │ │ 「演算子」の内容をカラムへセットする
│ │ │ │ │ │ 2項」の内容をカラムへセットする
- 🔁 「結果」の内容をカラムへセットする
□ 行モデルの内容をテーブルへ追加する

パラメータ定義3行目は任意項目の為、今回は未定義で問題ありません。

※図49に「ローカル変数」と「変数」がありますが、同じものです。

ここで F5 キーを押し、テスト実行を行いましょう。

問題がなければ「Calculation」で計算、閉じるボタンを押すと「CalculationTable」に反映されていると思います。

AS CalculationTa	ble	-	
		追加 (正削除
1項	演算子	2項	結果
1	+	1	2
1.			

5.3.6 CalculationTable_Del

「CalculationTable」で選択した行の削除を行う実装

まずはイベント割当定義が出来ているか確認お願いします。

手順1:カレント行を削除する

【図 50】

🧥 リレーションマネージャ:CalculationTable_Del



【図 51】



【図 52】



ここで F5 キーを押し、テスト実行を行いましょう。

問題がなければ「CalculationTable」で選択した行を削除することができます。

5.3.7 CalculationTable_Edit

「CalculationTable」で選択した行の修正を行う実装

まずは<u>イベント割当定義</u>が出来ているか確認お願いします。

- <u>手順1</u>: 選択中である行のテーブル行モデルを取得
- <u>手順2</u>: 変数「edit 値」から「1 項」の値を取得する
- 手順3:「value」から値を取得して計算式の左側に代入する
- 手順4: 変数「edit 値」から「演算子」の値を取得する
- 手順5:コンボボックスを選択状態にする
- <u>手順6</u>: 変数「edit 値」から「2項」の値を取得する
- 手順7:「value」から値を取得して計算式の左側に代入する
- 手順8: 変数から「結果」の値を取得する
- 手順9:「value」から値を取得してテキストボックスに代入する

<u>手順10</u>:テスト実行

手順1: 選択中である行のテーブル行モデルを取得

ローカル変数「edit 値」は未作成なので作成してください。

【図 53】

一 D Sec 修正		┝─ 🗂 UI ┝─ 🗂 JAVA Swing
新規作成	▶ 選択した種類	 ・ ・ ・
セクション作い	成(上へ)(代入	
セクション作用	成(下へ) リレーション	〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
貼り付け	サブシナリオ	
前セクション。	ヒマージ 分岐	

【図 54】



【図 55】

AS ISAD_VariableInfoDefine X	× ⁴⁸ × × × × × × × × × × × × × × × × × × ×	- 🗆 X
名前 : editte	タイトル: 選択中である行のテーブル 説明・コメント: 選択中である行のテーブル行モデルを取得して返す	処理選択
説明 :	戻り値: ローカル変数 ▼ editi節	2
型 : 任意の型 設定	バラメーター定義 No 種類 値 1 画面モデル CalculationTable/calc_tbl な	
初期値: ③ ビリラメータとして使用する ◎ 値渡し ○ 参照渡し 確定 閉じる		
4	違振中である行のテーブル行モデルを取得する閲覧。戻り値でモデルを戻す。 ID:gelCurrentISeeTableRowModeLF	Ē
	۲ <u>ــــــــــــــــــــــــــــــــــــ</u>	
		,て返す

「パラメータとして使用する」にチェックを入れてください

手順2:変数「edit 値」から「1項」の値を取得する

ローカル変数「value」は未作成なので作成してください。

【図 56】

P □ Sec	修正	- 	
	新規作成	選択した種類	
	セクション作成(上へ)	代入	P-☐ UI
	セクション作成(下へ)	リレーション	
	貼り付け	サブシナリオ	
	前セクションとマージ	分岐	✓ □ テーダ・ スァラビス □ DBテーブルのカラム情報を取得する。
	2445万ショントラニミジ	11	

【図 57】





【図 58】

As ISAD VariableInfoDefine ×		<u>#8</u> シナリオリレーション —	
		タイトル: 変数「editliを」から「1項」の値を取得する	処理選択
名前 : value		説明・コメント: データベースレコードのカラムから内容を取得して返す	
説明 :			
		バラメーター定義 <u>19月1日時間 1 ↓</u> No 新聞 値	
型: 任意の型 設定			
初期値:			
□パニノークトレク体田オス ◎ 防盗」 ○ 条肥通し 協会 問じる			
	1	「想能説明」 パラメータ説明	
		データベースレコードのカラムから内容を取得する関数	
		ID : getTableColumnValueFunction	
			रो ।
			登録 戻る
			T
		🏱 🔲 Section	। - ₹` Б=≠
		──□ 選択甲 ごのる打切テーフル打モナルを取得 ──□ 変数「edit値」から「1項」の値を取得する	し ()必9

手順3:「value」から値を取得して計算式の左側に代入する

【図 59】



【図 60】



※図57に「ローカル変数」と「変数」がありますが、同じものです。

手順4:変数「edit値」から「演算子」の値を取得する

2:変数「edit 値」から「1項」の値を取得するをコピーして修正をしましょう

違うポイントは「画面部品」で指定するカラムです。

【図 61】



【図 62】



手順5:コンボボックスを選択状態にする

【図 63】



【図 64】



【図 65】

糸8 シナリオリレーション				_		\times
タイトル: コンボボックスを選択状	劇にする				処	理選択
説明・コメント: コンボボックス、リスト)	ほど、リスト系のコンボーネン	トで、指定したア・	イテムを選択状態に	:する 		
パラメーター定義	追加 削除 ↑ ↓	画面モデル指注	定			
No 種類	值 cmb	画面モデル	Calculation			III III
2 ローカル変数 value	_cmb		名前		種類	
		first_txf		JTextField		
		operator_cmb)	JComboBox		
		second_txf	-	JTextField		
		calc_btn		JButton		
		close_btn		JButton		
		result_txa		JTextArea		
		体水影会				
		10/2 BR/E	カフォージー			
機能説明 パラメータ説明		選択1 ノナッ	クスを設定			
		リストモテル	で設定			
ハラメニジ 1、溜切を指定する対象のコンポークント		インスタンス	を設定			
 ・	- ナ 避中な鄙烈全する					
2. ABI/(1/(28/C 9.5/108 NOLE/218/EO/C44	1C197 781/1/5 HHM 3 20					
p		- I'			- 	禄戻る

— 🗋 選択中である行のテーブル行モデルを取得して返す
— 🗋 変数「edit値」から「1項」の値を取得する
- 🗋 「value」から値を取得して計算式の左側に代入する
— 🗋 変数から「演算子」の値を取得する
- 🗋 コンボボックスを選択状態にする
- □ 安奴がら 「演算」」の値を収得する - □ コンボボックスを選択状態にする



手順6:変数「edit値」から「2項」の値を取得する

2:変数「edit 値」から「1項」の値を取得するをコピーして修正をしましょう

違うポイントは「画面部品」で指定するカラムです。

【図 66】



【図 67】



手順7:「value」から値を取得して計算式の左側に代入する

【図 68】



【図 69】



手順8:変数から「結果」の値を取得する

【図 70】



【図 71】


手順9:「value」から値を取得してテキストボックスに代入する

【図 72】



【図 73】



ここで F5 キーを押し、テスト実行を行いましょう。

問題がなければ「CalculationTable」の行を選択して「修正」を押下すると「Calculation」に選択した行の計算内容が反

映されます。

AS CalculationTa	ble		— C	x c
		追加	修正	削除
1項	演算子	2 項		結果
1	+	2	3	
3	+	2	5	
2	-	1	1	
	計算式: 2	-	▼ 1 計算	閉じる
	1			

5.3.8 【共通】合計処理

「CulculationTable」で「メニュー」画面から「合計」を押下すると「結果」カラムの合計数値が表示される実装

手順1:シナリオ定義

- 手順2:「メニュー」画面の作成
- <u>手順3</u>:「CalculationTable」で右クリックをすると「メニュー」が出てくる実装
- 手順4:テーブルの全ての行を取得する
- 手順5: 最終合計値に「0」を代入する
- <u>手順6</u>: すべての行に対して、下記の処理を繰り返し実行
- 手順7:行の「結果」カラムから値を取り出す
- 手順8: 取り出した値を合計値に足す

手順9:計算された合計値を返す

<u>手順10</u>:テスト実行

手順1:シナリオ定義

今回実装する「合計処理」のシナリオ定義を作成しましょう。

その後実装する「平均」「最大値」「最小値」「【共通】合計処理」のシナリオ定義作成も同時に行います。

作り方がわからなくなってしまった方はこちら。

手順2:「メニュー」画面の作成

【図 74】





【図 75】

#8 ISAD_MenuGenerator		- c) X	Л	力		\times
メニューIDMenu □ メニュー定義	項目表示名	】 ショートカッ	自加 肖明除 ト 国際化		?	項目キーを入力してください。 合計 OK 取消	
		生	成開じる				
メッセージ表示領域			完成: 3/4				

【図 76】

AS ISAD_MenuItemActionGenerator	─ □ ×
メニューD 東京順 項目キー 項目表示名 ● 合計 合計 合計 ● 平均 平均 平均 ● 最大値 最大値 最大値 ● 最小値 日本 二合計 項目キー 二合計 合計 ○ 銀小値 日本 二合計 項目キー 二合計 一項目表示名 ○ メニュー ● 処理シナリオ : ● 画面遷移あり	
メッセージ表示領域 ・ Ci イベント割当定義 - Ci E_Calculation	<u>生成</u> 開じる 完成: 3/4 ビリソース □ フォルダ 選択 関じる

【図 77】

表示順	項目キ	=·	項目表示名	ショートカー	ット	国際化	
1	合計		合計				
2	平均		平均				
3	最大値		最大値				
4	最小値		最小値				
項目キー	:	合計					
項目表示約	ሬ :	合計			[国際化	
ב⊐א⊘	- :						
◉ 処理シ	ナリオ :	合計処	理				
三 画面说	駆あり						
					生成	閉じる	

手順3:「CalculationTable」で右クリックをすると「メニュー」が出てくる実装

【図 78】



ここから「【共通】合計処理」のリレーション作成に移ります。

手順4:テーブルの全ての行を取得する

ローカル変数「すべての行」は未作成なので作成してください。

【図 79】

<i>ฝ</i> ร リレーシ	ョンマネージャ:【共通】合	計処理	
- 🗋 Secti	修正		□ テーブルモデルのハンドル
	新規作成	選択した種類	∳
	セクション作成(上へ)	代入	□ テーブルモデルのハンドル (テーブル行モデルのハンドル)を取得して返す □ なままルの内容をエニブルの追加する
	セクション作成(下へ)	リレーション	- アーブルの指定した行に、行モデルの内容を挿入する
	貼り付け	サブシナリオ	- □ 選択中テーブルの行を、行モデルで上書きする - □ 指定したテーブルの行を、行モデルの内容で上書きする
	前セクションとマージ	分岐	□ テーブル行モデルで指定した行を削除する □ テーブル行モデルで指定したインデックスの行を削除する
	後セクションとマージ	ループ	└ □ ᆕ - ゔルエデルのカレントに行た地やオス
	セクションの削除	ブロック	
	サブセクション化	中断	
	脱サブセクション	リターン	
	セクション順序変更	共通データリンク	





【図 81】



手順5:最終合計値に「0」を代入する

ローカル変数「合計結果」は未作成なので作成してください。

【図 82】



AS IS	A	D_VariableInfoDefine		×
名前	:	合計結果		
説明	:			
型	:	任意の型		設定
初期値	:			
		□ パラメータとして使用	する 🖲 値渡し	○ 参照渡し 確定 閉じる
				➡
H-7	л	永 就一見		
No		引数のタイプ		変数名
	'申日	目しない 用したい	すべての行	
2	野	用しない	合計結果	
				4

【図 83】



手順6:すべての行に対して、下記の処理を繰り返し実行

ローカル変数「一行」は未作成なので作成してください。

※ループ処理についての考え方(フローチャート図)はこちらを参照

【図 84】

	松 リレーションマネージャ:【共通	】合計処理				12.4 UZ K 1201401			
シイトル : すべての行に対して、下記の処理を繰り返し実行 第 新規作成 違訳した種類 セクション作成(上へ) 代入 セクション作成(下へ) リレーション 助け付け サブシナリオ 前セクションとマージ ノレーブ 後セクションとマージ ノレーブ グロック ブロック サブセクション化 中断 脱サブセクション加厚家変更 リターン 中断 山東一・ 東通データリンク (本) レクション加厚家変更 (本)				AS ISAD_Sce	enarioCellLoop			- 0	×
 ● 「● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●				タイトル	: すべての行に対して	、下記の処理を繰り返し実行			
● 1 ● 1 ● 4 ● 1 ● 5 ● 5 ● 5 ● 5 <t< th=""><th>P □ Sec</th><th>L</th><th>-</th><th>説明・コメント</th><th></th><th></th><th></th><th></th><th></th></t<>	P □ Sec	L	-	説明・コメント					
■ 新規作成 選択した種類目 セクション作成(上へ) 代入 セクション作成(下へ) リレーション 貼り付け サブシナリオ 前セクションとマージ 分岐 後セクションとマージ 小ーブ セクションの削除 ブロック サブセクション化 中断 脱サブセクション リターン セクション順序変更 共通データリンク		15	u	□繰返しの指定					
セクション作成(上へ) 代入 セクション作成(下へ) リレーション 貼り付け サブシナリオ 前セクションとマージ 分岐 後セクションとマージ 分岐 「OR NEXT 縦辺し変数 「なりションとマージ 小ーブ レーブ 「のR NEXT 「GR ACH 縦辺し変数 「安設 「日ーカル変数線 「日本 「日本 「日本	- □ 롸 新規作成	▶ 選択した種類			繰返し条件: 条件式			条件編集	詳細条件
セクション作成(下へ) リレーション 貼り付け サブシナリオ 前セクションとマージ 分岐 後セクションとマージ ルーブ マクションの削除 ブロック サブセクション化 中断 脱サブセクション リターン セクション順序変更 共通データリンク	セクション作成(上へ)	代入			· · · · · · · · · · · · · · · · · · ·				
貼り付け サブシナリオ 第83年 ・ 2年10 「 700400 ********************************	セクション作成(下へ)	リレーション		I FOR EACH	編述し変数: 支数石 集合要表 ・ 更去名		ローカル変数	ロー カル 生 今 選 捉	支勁選択
前セクションとマージ 分岐 1 定数 選択 開提 後セクションとマージ ルーブ 第週し変数期順値 1 定数 選択 開提 セクションの削除 ブロック ブロック 第週し変数記録 2 運取 選択 開提 サブセクション化 中断 6 第週し条件 : 条件式 条件編集 詳細条件 1 セクション順序変更 共通データリンク #	貼り付け	サブシナリオ			AD3676 . 35770		H 7570 3C3A		10175
1 1 定数 2	前セクションとフージ	公岐		U FUR NEXT	線返し変数	: 炎蚁名			/安凱選択
後セクションとマージ ルーブ 編進しステップ値 注 1 定数 遅根 膨振 セクションの削除 ブロック ジロック 編進し気テップ値 : [*] ▼ 1 定数 運根 調振 サブセクション化 中断 編進し条件 : 条件式 条件編集 詳細条件 脱サブセクション リターン <t< td=""><td>前にノノヨノヒマーノ</td><td>7) w<u>x</u></td><td></td><td></td><td>繰返し変数初期値</td><td>: 1</td><td>定数</td><td>選択</td><td>間接</td></t<>	前にノノヨノヒマーノ	7) w <u>x</u>			繰返し変数初期値	: 1	定数	選択	間接
セクションの削除 ブロック サブセクション化 中断 脱サブセクション リターン セクション順序変更 共通データリンク	後セクションとマージ	ルーブ			緑返しステップ値	: += 🔻 1	定数	選択	間接
サブセクション化 中断 ○編返し条件 : 条件式 条件編集 資編集 脱サブセクション リターン ほこ を ほこ を ほこ を セクション順序変更 共通データリンク	セクションの削除	ブロック			◉ 編返し終了変数値	: ≦ 💌 値・変数名・関数	定数	· 選択	間接
脱サブセクション リターン 確定 原る セクション順序変更 共通データリンク	サブセクション化	中断			◎ 繰返し条件	: 条件式		条件編集	詳細条件
セクション順序変更 共通データリンク	脱サブセクション	リターン							曜定 戻る
	セクション順序変更	共通データリンク							

【図 85】



手順7:行の「結果」カラムから値を取り出す

これは「すべての行に対して、下記の処理を繰り返し実行」の直下に作成したいので、「すべての行に対して、下記の処理を 繰り返し実行」を右クリックで「新規作成」→「リレーション」→「データベースレコードのカラムから内容を取得して返す」を選択

ローカル変数「結果カラムの値」は未作成なので作成してください。

【図 86】



【図 87】



パラメータ定義の No.1 はローカル変数「一行」を選択してください。

【図 88】



手順8:取り出した値を合計値に足す

これも「すべての行に対して、下記の処理を繰り返し実行」の直下に作成したいので、「すべての行に対して、下記の処理を 繰り返し実行」を右クリックで「新規作成」→「リレーション」→「足し算(加算)を行って、計算結果を返す」を選択

【図 89】



分類階層表示				
🗖 加賀				
└□□足し算	(加算)	を行って、	計算結果を返す	J

【図 90】

<u>#8</u> シナリオリレーション		-	
タイトル: 足し算(加算)を行	テって、計算結果を返す		処理選択
説明・コメント :			
戻り値: ローカル変数 🔻	合計結果		選択
パラメーター定義	追加 削除 ↑ ↓		
No 種類	値		
支数 吉訂稿未 2 変数 結果カラムの何	<u>.</u>		
(機能説明) パラメータ説明 数値などの足し算(加算)を行う関数。			
ID : additionFunction			
			登録 戻る





手順9:計算された合計値を返す

【図 91】





ここで F5 キーを押し、テスト実行を行いましょう。

問題がなければ「CalculationTable」内で右クリックをすると「メニュー」が表示され「合計」を押下すると「結果」カラムの合

計値がメッセージとして表示されます。

AS Calculation	Table		_		\times
		追加	修正	前小阪)î
1 項	演算子	2項		結果	
1	+	2	3		
3	+	2	5		
2	-	1	1		
	合計値の計	算結果			×
	(i	9			
		ОК]		

5.3.9 合計処理

手順1:「【共通】合計処理」を実行する

手順2:合計値の計算結果を表示する

ローカル変数「合計値」は未作成なので作成してください。

【図 93】

A8 リレーションマネージャ:合計処理	AS ISAD_ISeeResourcesChooser
	リソースの選択
・ ・ 。 。 。 。 。 。 。 。 。 。 。 。 』<	シナリオ定義 CalculationTable_初期化 CalculationTable_Add CalculationTable_Del CalculationTable_edit Calculation_Add Calculation_Add
☆ ISAD_ScenarioCellSubScenarioDirect - □ × タイトル: 「(共通) 合計処理」を実行する	□ 取入値処理 □ 最小値処理 □ 平均処理(バックアップ)
説明・コメント: シナリオ:【共通】合計処理 シナリオパ▼ 指定	
バラメーター定義 バラメータの削除 処理道択 No バラメータタ名 パラメータタイプ 種類 値	
jLabel2	

【図 94】

		AIS ISA	D_VariableInfoDefine	_	\Box ×
As ISAD_ScenarioCellSubScenarioDirect	- 🗆 X	名前:	合計値		
タイトル: 「【共通】合計処理」を実行する		説明:	:		
説明 - コメント:					
シナリオ:【共通】合計処理	シナリオパ ▼ 指定				
戻り値: □−カル変数 ▼ 合計値					設定
パラメーター定義	パラメータ更新 パラメータ削除 処理選択		· 11/2/07/12		BAAC
No バラメータ名 バラメータタイプ	種類 値	初期值:		(2)	
			🗌 パラメータとして使用する	◉ 値渡し ○ 参照渡し	確定閉じる
	A			** ** *	
jLabel2			Y Li Section L Li [【共通】合言	計処理」を実行する	1

手順2:合計値の計算結果を表示する

【図 95】



【図 96】



閉じる

設定

【図 97】

パラ	メーター定義		ボキャブラリ選択			
No	種類	値		ボキャブラリ選択		
1	変数	合計値		値		
2	定数	合計値				
3	定数	INFORMATION MESSAGE		PLAIN_MESSAGE		
		_		INFORMATION_MESSAGE		
			QUESTION_MESSAGE			
				WARNING_MESSAGE		
				EDDOD MERRACE		

← □ Section
 □ 「【共通】合計処理」を実行する
 □ 合計値の計算結果を表示する
 □

5.3.10 平均処理

手順1:「【共通】合計処理を実行する」

- <u>手順2</u>: テーブルモデルの行数を取得して返す
- <u>手順3</u>:合計÷行数
- 手順4:合計値の計算結果を表示する
- <u>手順5</u>:テスト実行

手順1:「【共通】合計処理を実行する」

合計処理の手順1と同じです

手順2:テーブルモデルの行数を取得して返す

ローカル変数「行数」は未作成なので作成してください。

【図 98】



【図 99】



【図 100】



手順3:合計÷行数

【図 101】

የ-⊡ Sectio - [] [- [] ד	修正 新規作成	テする、 選択した種類				
	セクション作成(上へ)	代入	1 除質			
	セクション作成(下へ)	リレーション	「「割り算((除算)	を行い、	計算結果を返す
	貼り付け	サブシナリオ	1			
	前セクションとマージ	分岐				

【図 102】



手順4:合計値の計算結果を表示する

【図 103】



【図 104】



ここで F5 キーを押し、テスト実行を行いましょう。

問題がなければ「CalculationTable」内で右クリックをすると「メニュー」が表示され「平均」を押下すると「結果」カラムの平

均値がメッセージとして表示されます。(図)

AS CalculationTa	ble	_	\square \times
		追加 修	E 削除
1項	演算子	2項	結果
1	+	2	3
3	+	2	5
2	-	1	1
	平均値の計算	算結果	×
	i 3		
		ОК	

5.3.11 最大值処理

「CulculationTable」で「メニュー」画面から「最大値」を押下すると「結果」カラムの最大数値が表示される実装

- <u>手順1</u>: Calculation Table のテーブル全ての行を取得する
- 手順2:「最大値」に比較する固定の値を代入する
- 手順3:繰り返し処理を行う
- 手順4:「一行」から「結果」の値を取得する
- <u>手順5</u>:分岐処理を行う
- 手順6:「結果の値」を「最大値」へ代入
- 手順7:「最大値」を表示する
- <u>手順8</u>:テスト実行

手順1: CalculationTable のテーブル全ての行を取得する

ローカル変数「すべての行」は未作成なので作成してください。

【図 105】



【図 106】



【図 107】

÷

					AIS IS	SAC				×
AR シナリオリレーション		-								
タイトル: CalculationTableのテーブル全ての行を取得する 説明・コメント: デーブルモデルのハンドル(テーブル行モデルのハンドル)を取得して返す		処理選択]	名前 説明	:	<u>すべての行</u>			
戻り値: □ーカル実数 ▼ すべての行		(1) jär							
パラメーター定義 追加 削除 ↑ ↓	画面モデル指定									
No 種類 値 1 画而モデル CalculationTable/calc tbl	画面モデル /CalculationTable		E¥%⊞		뽀	-	任意の型			設定
	名前 add_calc_btn	Rutton	種類		初期値	1:				
	edit_calc_btn del_calc_btn calc_tbl	JButton JButton JTable					□ パラメータとして使用する	◉ 値渡し	○ 参照渡し 隆定	閉じる
	値を設定									
標能説明 パラメータ説明	テーブルモナルを設定									
										_
1:ハンドルを取得したいSeeテーラルモテル					የ- 🚅	3 S	ection			
<u> </u>					1 1	— C) CalculationTableのテー	ブル全ての)行を取得する	
		3	登録戻る]						

手順2:「最大値」に比較する固定の値を代入する

ローカル変数「最大値」は未作成なので作成してください。

※ここでは、テーブルに入る値と比較して充分に小さい任意の値(-100000)を設定しています。 AiSee で設定できる最小値は「-9223372036854775808」

【図 108】



【図 109】



ローカル変数「一行」は未作成なので作成してください。

【図 110】


手順4:「一行」から「結果」の値を取得する

これは「繰り返し処理を行う」の直下に作成したいので、「繰り返し処理を行う」を右クリックで「新規作成」→「リレーション」→

「データベースレコードのカラムから内容を取得して返す」を選択

ローカル変数「結果の値」は未作成なので作成してください。

【図 111】



【図 112】



【図 113】



これも「繰り返し処理を行う」の直下に作成したいので、「繰り返し処理を行う」を右クリックで「新規作成」→「分岐」を選択

【図 114】

CalculationTableのテーブル全ての行を取得する CalculationTableのF F F F F F F F F F F F F F F F F F F	All ISAD_ScenarioCellCase — 🗆 🗙
 ● 「最大値」に比較する固定の値を代入する ● 「 繰返し処理を行 修正 ら内容を取得して返す 新規作成 ● 選択した種類 新規作成 (上へ) ● 代入 新規作成 (下へ) ● リレーション 削除 サブシナリオ コピー ① 分岐 切り取り ルーブ 	タイトル : 分岐処理を行う 説明・コメント: 3 分岐ケース指定 条件追加 条件編集 コメント 条件削除 羅列順番 現在順番 分岐条件
	 ▲ □「その他」処理を使用する ケースの実行条件: ● 排他実行 ○ 多重実行 □ Ga

【図 115】



【図 116】

タイトル : 条件を満たせば処理を実行	
説明・コメント:	
分岐ケース指定 条件追加 条件編集 コメント 条件削除	
羅列順番 現在順番 分岐条件	
1 ケース1 結果の値 > 最大値	
▲	
□ 「その他」処理を使用する ケースの実行条件 : ⑧ 排他実行 ○ 冬重実行	 Section CalculationTableのテーブル全ての行を取得する 協大値に比較する固定の値を代入する 最大値に比較する固定の値を代入する 最少ベースレコードのカラムから内容を取得して返す データベースレコードのカラムから内容を取得して返す 金件を満たせば処理を実行

これは「ケース1:結果の値>最大値」の直下に作成したいので、「ケース1:結果の値>最大値」を右クリックで「新規

作成」→「代入」を選択

【図 117】



手順7:「最大値」を表示する

【図 118】



【図 119】



<最大値処理の全体像>

🎌 リレーションマネージャ:最大値を表示		_		×
	DL一覧追加			
Ŷ ☐ Section	シナリオ説明	変数定義		
- D テーブルの全ての行を取得 - D '是太値' に 100000など 3	密勒名		種類	1
▶ □ 取べ値 に10000021(へ) ▶ □ すべての行'の中の各'→行'に対して順に以下の処理を繰返す	すべての行	任意の	<u>기포~주</u>	
- □ '─行' の"結果" カラムから値を取得	→行	任意の	型	
┡ 🗔 条件を満たせば処理を実行	合計値	任意の	型	
? □ ケース1:結果カラムの値 > 最大値	最大値	任意の	型	
-□'最大値を表示			-	
	タイトル: テーブルの全ての	行を取得		
	説明・コメント: 最大値を探すテー	ブル全体を	取得	
	バラメータ:			
	P1 画面モ CalculationTable/	デル: calc_tbl		
	Copyright(c) 2009 Bets	net Corporation.	#) << - 9 * • C C NI Rights Res	y h serv.
	6	確定 (保	存月	閉じる
1.000		_	_	

※'最大値'の初期値設定は Java 言語の long 型で扱える最小の値(-9223372036854775808)までであれば代入が可能

ここで**テスト実行**を行いましょう。

問題がなければ「CalculationTable」内で右クリックをすると「メニュー」が表示され「最大値」を押下すると「結果」カラムの

最大値がメッセージとして表示されます。(図)

AS Calculation	Table	_	
		追加 修正	E 削除
1項	演算子	2項	結果
5	+	2	7
5	-	2	3
1	+	1	2
	最大値の検索結	果	×
	(i) 7		
		OK	

5.3.12 最小值処理

「CulculationTable」で「メニュー」画面から「最小値」を押下すると「結果」カラムの最小数値が表示される実装

- 手順1:最大値処理からリレーションをコピーしてくる
- 手順2:「最小値」に比較する固定の値を代入する
- 手順3:条件を満たせば処理を実行
- 手順4:「結果の値」を「最小値」へ代入
- 手順5: 「最小値」を表示する
- <u>手順6</u>:テスト実行

手順1:最大値処理からリレーションをコピーしてくる

【図 120】





「最小値処理」に貼り付けをした後に右下の「確定」を押してください。

警告文が出てきますが、「はい」で一度確定をして未定義の変数を自動で定義しましょう。

手順2:「最小値」に比較する固定の値を代入する

※ここでは、テーブルに入る値と比較して充分に大きい任意の値(100000)を設定しています。 AiSee で設定できる最大値は「9223372036854775807」

【図 121】



手順3:条件を満たせば処理を実行

これも「繰り返し処理を行う」の直下に作成したいので、「繰り返し処理を行う」を右クリックで「新規作成」→「分岐」を選択

【図 122】



【図 123】

AS ISAD_ScenarioCellCase − □ ×				
タイトル : 条作 説明・コメント:	キを満たせば処理を実行			
分岐ケース指定	条件追加	:件編集 🗆	マント	条件削除
羅列順番 現在順番		分岐条件		
1 ケース1	結果の値 < 最小値			
🗌 「その他」 処理を停	使用する ケースの実行	条件: ⑧ 协	非他実行 (曜	<u>) 多重実行</u> 定 戻る

手順4:「結果の値」を「最小値」へ代入

これは「ケース1:結果の値く最小値」の直下に作成したいので、「ケース1:結果の値く最小値」右クリックで「新規作

成」→「代入」を選択

ローカル変数「最小値」は未作成なので作成してください。

【図 124】

	AS ISAD_Scenari	oCellSubstitu	tionMulti		-		\times
- □ 一行から「結果」の値を取得する • □ 条件を満たせば処理を実行	タイトル :	結果の値'を'最小(直へ代入				
←□ ケース1:結果の値 < 最小値 □ 注里の値(を長大値)	説明・コメント:						
- □ '最大値'を表示する							
新規作成(上へ) 🕨			左辺指定	右辺指定	追加	削除 ↑	
	代入先	種類		代入元	Ċ	種類	
	最小値	変数	. <=	結果の値		ローカル変数	. <u> </u>
AS ISAD_VariableInfoDefine X						確定	戻る
名前 : 最小值							
説明 :							
			•				
	🛉 🗐 គឺអ៊	返し処理を	行う				
		〕`─行`から	「結果」	の値を取得	导する		
型 : 任意の型 設定	P-□	■ 条件を満7	こせは処地 1・24里	里を実行 の値 < 長	ヨル値		
初期值:			雨木	の過入れ 最小値へ位	をつう 七入		
□パラメータとして使用する ⑧ 値渡し ○参照渡し 確定 閉じる	日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日	大値を表示	する		47 V		
□ ハラメータとして使用する ● 値渡し ○ 参照渡し 唯定 励しる	一口 垢	大値を表示	する				

手順5:「最小値」を表示する

【図 125】



<最小値処理の全体像>

州 リレーションマネージャ:最小値を表示	– 🗆 X
	DL一覧追加
 P □ Section □ F - ブルの全ての行を取得 □ '最小値に1000002 代入 ● 'マイでの行の中の各'行'に対して順に以下の処理を繰返す ● '-行'の"結果" カラムから値を取得 ● □ '-行'の"結果"カラムの値 < 最小値 ● (□ ケース1: 結果カラムの値 < 最小値 □ '品果カラムの値 'を 最小値'に代入 □ '最小値を表示 	シナリオ説明 変数名 種類 支べての行 任意の型 一行 任意の型 最小値 任意の型 最小値 任意の型 結果力うムの値 任意の型 行鼓 任意の型 201 任意の型 第二 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
<u>↓ … ↓</u> 录小値を表示	P1 国田モデル: Calculation able/Calc_ton 株)ハーウネット Copyright(c) 2009 Betanet Corporation. All Rights Reserv. 確定 保存 閉じる

※'最小値'の初期値設定は Java 言語の long 型で扱える最大の値(9223372036854775807)までであれば代入が可能

ここで**テスト実行**を行いましょう。

問題がなければ「CalculationTable」内で右クリックをすると「メニュー」が表示され「最小値」を押下すると「結果」カラムの

最小値がメッセージとして表示されます。

AS CalculationTa	able	_	\Box \times
		追加 修	正削除
1 項	演算子	2項	結果
1	+	2	3
3	+	2	5
2	-	1	1
	最小値の検索		×
	- i) 1		
		ОК	

5.4 データベースについて



<リレーショナルデータベースのイメージ図>

↓テーブル

ユーザーID	名前	PASSWORD	
0001	山田太郎	111	←レコード
0002	田中花子	222	
0003	鈴木次郎	333	
↑フィールド/ナ	і э д		

5.4.1 AiSee における DB の使用方法

基本的な DB を使用した大まかなフローは以下の通りです。



以降の章では具体的な例を見ながら実際に作成していきましょう。

5.4.2 DBを使ったシナリオ作成

今回は、以下のような DB のテーブル(SampleDB.db)を使用し、このテーブルに AiSee アプリからデ ータを書き込んだり、データの検索や削除ができるようにシナリオを作成します。 テーブルの名前が「UserInfo」で、「UserID」「Value」「Password」の3つのカラムを作成済です。

テーブル(T): 📃 UserInfo 🛛 🗸 🗸				
UserID	Value	Password		
ליייי	77	フィルター		

1.データを DB へ追加する

(1)リソース定義に、作成済の DB ファイルを登録



		- 0 đevo	(To the set	
→ * ↑ * * PC * 59720-F *	~	0 2 2000	- ト の快楽	
! ▼ 新しいフォルダー			🖽 🕶 🔟 🕐	
clip ^ 名前	更新日時	種類	サイズ	
Valorant > 昨日 (2)				
video				
会社資料用 () () () () () ()				
Creative Cloud File				
OneDrive Percer	2022/09/22 17:07	DTL ファイル	19 KB	
SampleDB.db	2022/09/22 14:52	Data Base File	12 KB	
PC Alsee Windowsアジリ作成例_v1.05.docx	2022/09/21 10:20	Microsoft Word	2,797 KB	
J 3D オブジェクト AiSee_Windowsアプリ作成例_v1.04.docx	2022/09/20 15:39	Microsoft Word	4,075 KB	
ダウンロード > 今月に入って(先週は含めず)(6)				
」デスクトップ				
ドキュメント > 元月(6)				
E ビクチャ				
ビデオ				
5ユージック				
Windows (C:)				
ローカル ティスク (C				
ネットワーク 🗸				
ファイル名(N): SampleDB.db			~	📗 🔶 🗂 リソース定
		20 (10)	Au suball	

①リソース定義を右クリック

②リソースを登録をクリック

③事前にダウンロードした「 SampleDB.db 」を選択→開くをクリック

(2)シナリオ作成

手順1:リソース定義フォルダパスを取得する

<u>手順2</u>:リソース定義フォルダパスに DB ファイル名を結合して DB ファイルパスを作成する

<u>手順3</u>:データベースへのコネクションを作成して返す

- <u>手順4</u>:挿入マップを作成する
- <u>手順5</u>:データベースを挿入する
- <u>手順6</u>:データベースをコミットする
- <u>手順7</u>:データベースのコネクションを切断する

※詳しい手順はリンク先を参照

手順1:リソース定義フォルダパスを取得する

<u>₩</u> シナリオリレーション	- 🗆 X
タイトル: リソース定義フォルダパスを取得する	処理選択
説明・コメント: 2	
戻り値: ローカル変数 ▼ ファイルパス	選択
パラメーター定義 追加 削除 1 ↓	ボキャブラリ選択 ▼
	ボキャブラリ選択
	値 APP_DIR LICENCES_DIR CAE_DIR LIE DIR RESOURCE_DIR
機能説明 パラメータ説明 パラメータ パラメータ 1:取得したいプロパティ名を指定	リソース定義フォルダ
	(3) 登録 戻る

②戻り値の指定(変数を作成してない場合はここで作成する) ③登録

<u>係</u> シナリオリレーション		_		×		
タイトル: リソース定義フォルダバスにDBファイル名を結合してDBファイルバスを作成する 説明・コメント:						
2 戻り値: ローカル変数 ▼ ファイルバス				選択		
パラメーター定義 追加 削除 ↑ ↓	変数選択			-		
	ローカル変数一覧	追加	修正	削除		
1 ローカル変数 ファイルバス 2 定数 /SampleDB.db	変数名		型			
\bigcirc	リスト Value コネクション					
	挿入マップ ファイルパス					
【 螺能説明 「 パラメータ説明	-					
 パラメータ 1: 結合したい文字列の1個目。 2: 結合したい文字列の2個目。3個以上結合したい時には、パラメーターを追加 Uて指定する。 						
		3	登録	戻る		

No.2 のところに定数「/SampleDB.db」と入力します ※任意で用意した場合はそのファイル名を入力してください ②戻り値の指定(変数を作成してない場合はここで作成する) ③登録

手順3:データベースへのコネクションを作成して返す

<u>₩</u> シナリオリレーション			- 🗆 X	
タイトル: データベースへのコネクシ		処理選択		
説明・コメント:			٦	
戻り値: ローカル変数 ▼ コネ	クション		選扎	R
パラメーター定義	追加 削除 ↑ ↓	ボキャブラリ選択		•
No 種類	値	ボキャブラリ選択		_
		值 ISQLITE		╈
3 ローカル変数 ファイルバス 4 定数 SQLITE		ORACLE		
5		SULSERVER		╞
		SQLitez中接続		
機能説明 パラメータ説明				
パラメータ 1・フーザーID				
2: パスワード				
3:ホスト名 4:データベースの種類				
5:データベースのインスタンス名 6:ポート番号				
F		(2	登録一戻る	>

No4 のところを左クリック→右の欄から「SQLITE」を選択 ②戻り値の指定(変数を作成してない場合はここで作成する) ③登録

※コネクションとはデータベースにアクセスするチャンネルのようなもの。 パラメータ定義の No3 と No4 にのみ上記変数を挿入することで OK

手順4:挿入マップを作成する

<u>#</u> \$ シナリオリレーション							- 0	×
タイトル:	挿入マップを作成する						処王	聖選択
説明・コメント:								
(2)								
戻り値:	コーカル変数 ▼ 挿入マッカ	ĵ						選択
パラメーター定義		<u>ا</u> ز	含力D	削除	î	Ļ		-
No 種類		値						
1							N	
2 定数	UserID							
3 画面モデル	TopPanel/user_txf/Value							
4 定数	Value							
<u>5</u> 画面モナル	TopPanel/value_txt/value							
0 定数	Password							
「定数	123450							
\bigcirc								
機能説明 パラ	メータ説明						-	
パラマータ	L						1	
1:指定したキーカ	『存在しなかったときに戻す値』	、指定しない地	易合に	コま、空々	*指定			
2:セットする値。	対応するキー					=		
3:セットする値								
						•	<u></u>	
								戻る

②戻り値の指定(変数を作成してない場合はここで作成する)

3登録

※挿入マップを作成することでデータベースとシナリオを紐づけることができる。

手順 5:データベースを挿入する

<u>₳</u> ₨ シナリオリレーション							_		×
タイトル: デー	ータベースを挿入する							処理	里選択
説明・コメント:									
パラマーター定差		追加	削除	1	1	変数選択			-
No 種類	(庙					ローカル変数一覧	追加	修正	削除
1 ローカル変数	コネクション					変数名		型	
2 定 設 2 2 二 定 数 3 ローカル変数	Userinfo 挿入マップ					行モデル			
	11777 22								
(1)						コネクション			
						挿入マップ			
						ファイルバス			
						-			
機能説明 パラメ・	一夕説明					1			
バラメータ					-				
1:コネクジョンID 2・テーブルタ									
2: 3: キーがカラム名、	マップ値で構成されたマップ。力き	ラム名に	:it. & ((半角ア)	ンパー				
ーサント)で区切って、	一文字のデータ型を指定する。デ	ータ型が	が未指定	時は、 🛛	′ (V				
ARCHAR)型と判断す	る。以下、指定可能なデータ型。括	弧内は	型の説明	。 V (V	AR 🖵		6	`	
-							4		
								登録	見る

①パラメータをセットする

②登録

手順6:データベースをコミットする

<u>₩</u> \$ シナリオリレーション						_		×
タイトル: データベースをコミットする	ò						如王	聖選択
説明・コメント :								
パラメーター定義	追加	削除	1	Ļ	変数選択			•
No	佰				ローカル変数一覧	追加	修正	削除
1 ローカル変数 コネクション					変数名		型	
\square					行モデル			
					value			_
					コネクション			_
					挿入マップ			
					ファイルバス			
					<u></u>			
一根能説明 パラメータ説明					-			
バラメータ				Ì				
1:コネクションID								
						(2)	_	
						Ŭ	登録	戻る

①パラメータをセットする

②登録

手順7:データベースのコネクションを切断する

AS シナリオリレーション								×
タイトル: データベースのコネクションを切断する								聖選択
説明・コメント : データベースのコネクションを切断す	3							
パラメーター定義	追加	削除	1	Ļ	変数選択			-
No 種類 (i		_	_		ローカル変数一覧	追加	修正	削除
1 ローカル変数 コネクション					変数名		型	
					行モデル			
					value			
					コネクション			
					挿入マッフ ファイルパフ			
						1		
機能説明 パラメータ説明								
バラメータ								
1: コネクションID								
						2	登録	展る

①パラメータをセットする

②登録

<追加のリレーション全体像>

▲ リレーションマネージャ:TopPanel_Addボタン	—		×
	DL一覧追加		
 		変数定義 値 仕意の型 仕意の型 仕意の型 仕意の型 仕意の型 仕意の型 仕意の型 日 (日意の型 日 (日意の型 日 (日意の型 日 (日意の型 日 (日意の型 日 (日意の型 日 (日本のの型 日	бр
		唯正 1未1子	- Com

2.DBを使用してデータを削除する

- (1)リソース定義に DB ファイルを登録※すでに登録している場合は省略
- (2)シナリオ作成
- 手順1:リソース定義フォルダパスを取得する
- 手順 2:リソース定義フォルダパスに DB ファイル名を結合して DB ファイルパスを作成する 手順 3:データベースへのコネクションを作成して返す
- <u>手順4</u>:条件マップを作成する
- <u>手順5</u>:データベースを削除する
- 手順6:データベースをコミットする
- 手順7:データベースのコネクションを切断する
- <u>手順8</u>:テーブルのカレント行を削除する

※手順1~3・6・7に関しては「追加」と一緒なのでこの章では省略

その他の詳しい手順はリンク先を参照

手順4:条件マップを作成する

糸 シナリオリレーション	-		×
タイトル: 条件マップを作成する		処理違	鈬
説明・コメント:			
戻り値: ローカル変数 🔻 条件マップ			選択
パラメーター定義 追加 削除 ↑ ↓			-
No 種類 值			
1 2 定数 UserID			
3 画面モデル TopPanel/user txf/Value			
\square			
一根能説明 パラメータ説明			
1:11年にしたキーの特徴しなかったとうに戻す他。11年じない場合には、生き指定 2:セットする値に対応するキー			
3:セットする値			
,,,,,,,,,,,	2)	70.03	=-

①パラメータをセットする

②戻り値の指定(変数を作成してない場合はここで作成する) ③登録

手順 5:データベースを削除する

(A					
一般 シナリオリレーション			_		×
タイトル: データベースを削除する				処理	選択
説明・コメント: データベーステーブルのレコードを削	I除(Delete)する				
パラメーター定義	追加 削除 ↑ ↓	変数選択			-
No 種類 (ii	i .	ローカル変数一覧	追加	修正	削除
1 ローカル変数 コネクション		変数名		型	
2 定数 Userinfo		コネクション			
<u>3 ローガル変数 条件マッフ</u> 4 字数 AND		条件マップ			
4 JEST AND		ファイルバス			
1					
パラメータ					
1: コネクションID					
2:削除の対象テーブル名					
3:削除する条件を、カラム名と値のISeeマップで指定	。条件が無い時には指定は 🦷				
不要。					
4: 条件を複数指定した時の項目間の条件の「OR」また	Elは「AND」を指定する。未 🚽				
		,	0	30.03	=7
			4	登赫	戻る

②登録

手順8:テーブルのカレント行を削除する

糸 シナリオリレーション			-	
タイトル: デーブルのウ 説明・コメント:	コレント行を削除する			処理選択
パラメーター定義	追加 削除 ↑ ↓	画面モデル指定		-
No 種類	値	画面モデル /TopPanel		言非細
1 画面モデル TopPan	el/value_tbl	名前	ħ	<u> 重換</u>
L		delete_btn	JButton	
		user_txf	JTextField	
		value_tbl	JTable	
		value_txf	JTextField	
		add_btn	JButton	
		値を設定		
		テーブルモデルを設定		
│ 【 機能説明 │ パラメータ説明		テーブルインスタンスを調	定	
バラメータ				
1:対象のテーブルモデル				
2:任意指定。自動付番の列が	「ある場合IIIは、列名を指定。			
		<u> </u>		
			2	登録 戻る

①パラメータをセットする

②登録

<削除のリレーション全体像>

パンションマネージャ: TopPanel_Delボタン	_		×
	DL一覧追加		
 	シナリオ説明 変数名 コネクション バリュー ファイルバス ユーザーID レコードセット レコードセット レコード指照 条件マップ 行モデル	変数定義 種 任意の型 任意の型 「た意の型 日本市の型 「たきの型 日本市の	2月 - フネット であ Reterv.
TopPanel_Debボタン			

3.DBを使用してデータを検索する

(1)リソース定義に DB ファイルを登録※すでに登録している場合は省略

(2)シナリオ作成

<u>手順1</u>:画面テーブルを削除する

<u>手順2</u>:リソース定義フォルダパスを取得する

<u>手順3</u>:リソース定義フォルダパスに DB ファイル名を結合して DB ファイルパスを作成する

<u>手順4</u>:データベースへのコネクションを作成して返す

<u>手順5</u>:挿入マップを作成する

<u>手順6</u>:データベースを検索する

<u>手順7</u>:レコードセットで繰り返し処理を行う

<u>手順8</u>:データベースのコネクションを切断する

手順1:画面テーブルを削除する

タイトル: テーブルのすべての行を削除する 説明・コメント: テーブルのすべての行を削除する パラメーター定義 追加 No 種類 1 画面モデル指定 画面モデル TopPanel/value_tbl 名前 JButton User_bf JTextField	
パラメーター定義 追加 削除 1 画面モデル指定 No 種類 値 画面モデル 「のPanel 1 画面モデル TopPanel/value_tbl 名前 4 4 4 4 1 国面モデル ブロPanel/value_tbl 1	処理選択
user_txf JTextField	▼ ● <
value_tbl JTable value_tvf JTextField add_tvn JButton	
1:対象のISeeテーブルモデル	

手順2:リソース定義フォルダパスを取得する

▲ シナリオリレーション	- 🗆 ×
タイトル: リソース定義フォルダバスを取得する 説明・コメント: ISeeが管理するブロバティを取得して返す	処理違択
戻り値: ローカル変 数 ▼ ファイルパス	選択
パラメーター定義 追加 削除 ↑ ↓ No 通知 (ホ 1 1 定数 RESOURCE_DIR	ボキャブラリ選択 ▼ ボキャブラリ選択
パラメータ 1:取得したいプロパティ名を指定 ■ ■	

手順 3:リソース定義フォルダパスに DB ファイル名を結合して DB ファイルパスを作成する

<u>48</u> シナリオリレーション			(×
タイトル: リソース定義フォルダにDBフ: 説明・コメント: 文字列を結合して返す	ァイル名を結合してDBファイルバスをf	作成する		QL.I	型選択
戻り値: ローカル変数 ▼ ファイル	íz				選択
パラメーター定義	追加 削除 ↑ ↓	変数選択			-
No 111*10	値	ローカル変数一覧	追加	修正	削除
1 ローカル変数 ファイルバス 2 定数 (Sample DB db		変数名		型	
Z JEXX Joanniebb.du		条件マップ			-
		コネクジョン			_
		行モデル			
		ユーザーID			
		ファイルバス			_
		レコード情報			-
一根能説明 バラメータ説明 パラメータ 1: 結合したい文字列の1個目。 2: 結合したい文字列の2個目。3個以上結合し して指定する。	たい時には、バラメーターを追加				
				登録	、戻る

手順 4:データベースへのコネクションを作成して返す

👭 シナリオリレー	ション				—		×
タイトル: ファ 説明・コメント: デー	ァイルバスからデータベースコネク - タベースへのコネクションを作成	?ションを作 にて返す	成して返す				処理選択
戻り値: 口 [、]	-カル変数 ▼ コネクション						選択
パラメーター定義		追加	削除 1	Ļ			-
No 種類 1 2 2 二 3 ローカル変数 4 定数 5 6 6 - パラメータ 1: ユーザーID 2: パスワード 3: ホスト名 4: データベーフの超 -	ファイルパス SOLITE ーク説明						
						3	登録 戻る

手順 5:挿入マップを作成する

#8 シナリオリレーション	- 0	×
タイトル: 条件マップを作成する		処理選択
説明・コメント: マップを作成して返す。初期値の指定が可能		
戻り値: ローカル変数 💌 条件マップ		選択
パラメーター定義 追加 削除 ↑ ↓		-
No 種類 值		
2 定数 Value		
3 画面モデル TopPanel/value_txf/Value		
【 機能説明 パラメータ説明		
パラメータ 4. 北方たしかり、たちまに声力は、北京したい組合には、夜を北京		
1:11年したキーが存在しなかったときに戻す他。指定しない場合には、生き指定		
3:セットする値		
P		登録 戻る

手順6:データベースを検索する

☆ シナリオリレー	VEV	_	(×
タイトル : デ- 御・コメント : デ-	タベースを検索する タベーステーブルを、テーブル名、カラム名と、カラム名と条件値のマップをF	用いて検索して	返す	如王	里選択
戻り値:[[]	· カル 変数 ▼ レコードセット				選択
パラメーター定義	追加 削除 ↑ ↓ 変数選択				-
No 新酒	<u>値</u> ローカル変	ご数一覧	追加	修正	削除
1 ローカル変数 2 空料	コネクション 変	数名		型	
2 正奴 3 定数	<u>条件マップ</u>				-
 レーカル変数 	条件マップ コネクション	2			
定数	AND についていたい しついてん しつい たたい しつい たたい しつい たたい しつい たたい しつい たたい しつい しつい しつい しつい しつい しつい しつい しつい しつい しつ	ット 			
		2			_
•	ド情報	Fil			-
機能説明 パラメ・ パラメータ 1:コネクションID 2:テーブル名 3:取得するカラムを ろ.	- 夕説明 - ク説明 カンマ区切りで指定する。全件を取得する時には、*を指定す ▼				
	,			登録	戻る

手順7:レコードセットで繰り返し処理を行う

🛚 🏄 リレーションマネージャ:TopPanel_検索	_		×
	DL一覧追加		
	シナリオ説明 変数名 コネクション パリュー ファイルパス ユーザーID レコード性ット レコード情報 条件マップ パモデル Copyright(c) 2008 Beac Langle (Copyright(c) 2008 Beac	変数定義 運 任意の型 任意の型 任意の型 任意の型 任意の型 任意の型 任意の型 任意の型 任意の型 任意の型 「意の型 「「意の型 「「意の型 「「「」」」 「「」」」 「「」」」 「「」」」 「「」」」 「「」」」 「「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」」 「「」」」 「」」 「「」」 「」」 「「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」」 「」」 「」」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 「」」 </td <td>977 h</td>	977 h
TopPanel_ 検索			

手順8:データベースのコネクションを切断する

▲ シナリオリレーション	_	C	כ	×
タイトル: コネクションを切断			処理	選択
説明・コメント: データベースのコネクションを切断する				
パラメーター定義 追加 削除 ↑ ↓ 変	変数選択			-
	コーカル変数一覧	追加	修正	削除
1 ローカル変数 ロネクション	変数名		型	
&	ミ件マップ			-
	コージョン			_
	フロードビッド			
	1-ザーID			
2	ファイルバス			
	ノコード情報			-
祝鮮能説明 パラメータ説明 パラメータ 1: コネクションID				
			登録	戻る

<検索のリレーション全体像>

ℳ リレーションマネージャ:TopPanel_検索	_		×
	DL一覧追加		
 ▶ C Section ▶ テーブルのすべての行を削除する ▶ リンース定義フォルダバスを取得する ▶ リンース定義フォルダバスを取得する ▶ ファイルバスからデータペースコネクションを作成して返す ▶ 条件マップを作成する ▶ データペースを検索する ♥ □ レコードセットで積返し処理を行う ▶ レコードセットででしを取得する ▶ レコード世報からユーザーDを取得する ▶ ローレード情報からエーザーDを取得する ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ロードセットで積返し処理を行う ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ローンドロを取得する ▶ ローンドロ・ドロを取得する ▶ ローンドログルを住成して近いまする ▶ ロージーロ・「情報からエーザーDをセットする ▶ 「行モデルへマリーロをセットする ▶ 「行モデルへValuaをセットする ▶ 「行モデルへValuaをセットする ▶ 「行モデルへValuaをセットする ▶ 「行モデルへを可じん」 ▶ コネクションを切断 	シナリオ説明 変数名 コネクション バリュー ファイルパス ユーザーID レコードセット レコード性ット レコード情報 条件マップ 行モデル ・ ・ <td>変数定義 (注意の型 (注意の型</td> <td></td>	変数定義 (注意の型 (注意の型	
TopPanel_検索			

5.4.3 DB を作成したい場合(DB Browser)

上記で作成したアプリでは、事前に DB の作成を「SQLite」を用いて行っています。 ここでは簡単に、SQLite のデータが作成、編集できる「DB Browser」について紹介します。

DBを直接作成・編集したい場合には、以下の2つをインストールしてください。

SQLitehttps://www.sqlite.org/download.html/DB Browserhttps://sqlitebrowser.org/blog/version-3-12-2-released/

DB Browser をインストールする際に、デスクトップにショートカットを作成するオプションを選 択すると、このようなアイコンが表示されます。 ※DB Browser(SQL Cipher) はインストール不要です。DB Browser(SQLite)のみ使用します。



下図は DB Browser の画面です。作成済みの「○○○.db」ファイルを開いた場合、「データ閲覧」のタブから、データベースのテーブルの内容が以下のように表示されます。 (「○○○.db」ファイルをドラッグ&ドロップで開けます)

このページ上でテーブルのデータを直接編集することが可能です。 変更した後は「変更を書き込み」ボタンを押して内容を更新してください。



DBを新規作成したい場合には、現在開いている **DB** があれば一旦接続解除してから、画面左上の「新しいデータベース」ボタンをクリックし作成してください。
6 NetBeans でアプリの画面を作成する方法

ここでは、NetBeansを用いて、アプリの画面の作成に必要な Jar ファイルを作成します。

最初に Java1.8(JDK8)と、NetBeans IDE15 をインストールしてください。

< Java1.8(JDK8) > http://www.betanet.co.jp/download/jdk-8u192-windows-x64.exe

< NetBeans IDE15> <u>https://netbeans.apache.org/download/nb15/</u>

上記をインストール後、以下の手順で作成していきます。

手順1:NetBeans を起動

<u>手順 2</u>:パッケージとクラスを作成

- 手順 3: パネルの name にクラス名を入力
- 手順4:画面に部品を並べる
- <u>手順 5</u>: AiSee で操作する全てのコンポーネント(ボタン等の部品)の名前を設定
- <u>手順6</u>: JDK の設定
- 手順7:プロジェクトをビルドする
- 手順8: Jar ファイルが作成できたことを確認

手順1: NetBeans を起動

「File」→「New Project」 で新しいプロジェクトを作成します。





プロジェクト名(ここでは ScoreBoard)を入力、Create Main Class のチェックは外し

Finish を押下します。

🜔 New Java Application					×
Steps	Name and Locatio	n			
 Choose Project Name and Location 	Project <u>N</u> ame:	ScoreBoard			
	Project <u>L</u> ocation:	C:¥Users	¥Documents	€NetBeansProjects	Browse
	Project Fol <u>d</u> er:	sers¥terumin	¥Documents¥Net8	BeansProjects¥Score	Board_
	Use Dedicated	Folder for Stor	ing Libraries		
	L <u>i</u> braries Folder	r:			Bro <u>w</u> se
		Different use compilation	rs and projects ca libraries (see Help	n share the same for details).	
	Create Main Cl	ass scoreboa	rdScoreBoard_		
		< <u>B</u> ack	Next >	<u>Finish</u>	Cancel <u>H</u> elp

手順2:パッケージとクラスを作成

下図の通り「Source Package」の上で右クリック →「New」→「Java Package」を作成します。

Package 名は任意の名前で作成が可能です。

(Package は複数作る事ができますが、ここではまず1つ作成します)

Image:	o
Projects × Files Services _ Start	
	Ρ
V 🧶 ScoreBoard	
Figure Part Source Part Source Part (defau New) Folder	
> 📑 Libraries Find	
Paste Ctrl+V 📑 Java Package	
History > Java Interface	
Tools > Entity Class	
Properties 🔯 Entity Classes from Database	
👳 Web Service Client	
Other	

Steps	Name and Loca	tion
 Choose File Type Name and Location 	Package Name:	newpackage
	Project:	ScoreBoard
	Location:	Source Packages
	Created Folder:	C:¥Users¥terumin¥Documents¥NetBeansProjects¥ScoreBoard¥src¥newpack

次に、名前を付けたパッケージを右クリックして、「New」→「JPanel Form」をクリック。

分かりやすい Class Name (クラス名)を決め、Finish を押下し JPanel クラスを作成します。

ここで決めたクラス名は、以下手順3でも入力します。

🔾 File Edit Vie	w Navigate Sou	rce Refac	tor Run Debug Profile T	eam Tools	Wi			
: 🔁 🕋 🎙	1 · • •	<defau< td=""><td>ılt config> 🗸 🌚 - 🚏</td><td>€ ا</td><td></td><td>New JPanel Form</td><td></td><td>×</td></defau<>	ılt config> 🗸 🌚 - 🚏	€ ا		New JPanel Form		×
Projects × Files	Services		_	Start Page 🔾	<	Steps	Name and L	ocation
✓	kages					Choose File Type Name and Location	Class Name:	NewJPanel
package	New	>	Folder				Project:	ScoreBoard
	Find	Ctrl+F	🔠 Java Package				Location:	Source Packages V
	Cut	Ctrl+X	JFrame Form				Package:	package1 V
	Сору	Ctrl+C	Java Class				Created File:	$C: {\sf FU} sers {\sf Fterumin} {\sf FD} ocuments {\sf FN} et {\sf BeansProjects} {\sf ScoreBoard} {\sf Fsrc} {\sf Fpackage1} {\sf FN} of {\sf Formula} and {\sf F$
	Paste	Ctrl+V	JPanel Form					
	Delete	Delete	🚳 Entity Class				Superclass:	Browse
	Refactor	>	Entity Classes from Database	e			Interfaces:	Browse
	Compile Packag	ge F9	Web Service Client	_				
	Test Package	Ctrl+F6	Other					
	Run Selenium I	ests						
Navigator $ imes$	Tools	>	_					
			-					< Back Next > Finish Cancel Help

手順 3: パネルの name にクラス名を入力

下図の左側の四角は、初めから作成されているパネルで、この上に部品を並べていきます。

まずはこのパネルを選択した状態で、JPanelの「プロパティ」内、「name」欄にクラス名を入力します。

任意の名前ではなく、手順2で決めたクラス名と一字一句同じように入力してください。

(ここでは「TopPanel」と入力しています。)

※下図では事前に NetBeans の日本語化を行っております。

	開始ペー	-ジ×回To	pPanel	.iava ×			$\leftrightarrow \bullet \Box$	パレット ×		-
	1-7	デザイン	履麻			TTE la la	1 H	- Swingコンテナ		^
		5512	版正			· · · · · · · ·	· ·	□パネル	□タブ付きペイン	
	♀ 「ツー」	レ」>「パレッ	ŀ]>[2w	ving/AWT	「コンポー	ーネント」メ	ニュー項	■分割ペイン	💷 スクロール・ペイン	
									<i>℡デス</i> クトップ・ペイン	
								🗆 内部フレーム	■ レイヤー・ペイン	
								- Swingコントロール		
								wher ラベル	◎ ボタン	
								∞ トグル・ボタン	☞ チェック・ボックス	
手順々、両声に或りま								<u>⊶ ラジオ・ボタン</u>	≋ボタン・グループ	
于順4:回国に即四%								□ コンボ・ボックス	■リスト	
								ロテキスト・フィールド	≤テキスト領域	
								∞ スクロール・バー	◎ スライダ	
								■ 進捗バー	፼書式付きフィールド	-
								□ パスワード・フィールド	□ スピナー	
								■セパレータ	エテキスト・ペイン	
画面左のパネ								■ エディタ・ペイン	יויש 🔄	
								三		~
								[JPanel] - ブロパティ ×		-
下の例では「E								プロパティ バインド 1	ベント コード	
								focusTraversalPolicyProvider		
								font	MS UI Gothic 24 プレーン	
								inheritsPopupMenu		•••
								insets		
								maximumSize	32767. 32767	
							ſ	name	TopPanel	
							. L	aartEegyaableComponent]
また、石下の								opaque		•••
								preferredSize	400 3001	
								requestFocusEnabled		
								verifyInputWhenFocusTarget		•••
								アクセンビリティ アクセンビリティ対応の名前		
								アクセシビリティ対応の説明		
								アクセシビリティ対応の親		
								name		?
								(iava.lang.String) name		

Start Page 🗙 📄 Board.java 🗙 💎 🗆	Palette ×
Source Design History III IIII IIIIIIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	 ✓ Swing Containers Panel ☐ Tabbed Pane ☐ Split Pane ☐ Scroll Pane ☐ Tool Bar ☐ Desktop Pane
1 2 3 4 5 6 7 8 9 合計 Team A 1<	Internal Frame Swing Controls Weel Label Weel Label Weel Controls Weel Label Weel Control W
Team B 1 1 1 1 1 1 1 1 カウント	✓ Text Area Im Scroll Bar Im Scholl Bar Im Progress Bar Im Formatted Field Im Password Field Im Spinner Im Separator Im Text Pane Im Editor Pane Im Tree Im Table
	Swing Vindows Swing Fillers [JPanel] - Properties ×
	Properties Events Code
	background [242,242,242] border (No Border)
	foreground [0,0,0]
	Other Properties
	alignmentY 0.5

※レイアウト形式について、今回は「Null Layout」で作成します。

画面左のパネル部分を右クリック→「Set Layout」 → 「Null Layout」 を選択します。

Team A II 1 1 1 1 1 1 1 1 Team B 1 1 1 1 1 1 1 1	合計 1 d1 1 d2	
	Preview Design > Events >	
(Set Layout >	Free Design
	Add From Palette > Design This Container Set to Default Size Edit Layout Space	Absolute Layout Border Layout (Default) Box Layout Card Layout Flow Layout
	Change Order	Grid Bag Layout
	Copy Paste Ctrl+V	Null Layout
	Customize Code	Overlay Layout
	Properties	

 \sim \Box Board.java \times Palette \times History 🛛 🔛 🚰 🔛 🔛 🖽 🕂 🖽 🛤 Swing Containers Panel 🛅 Tabbed Pane 🛛 📃 Split Pane E Scroll Pane it manager of a container use Set Layout submenu from its 🗁 Desktop Pane 🧮 Internal Frame 🔳 Layered Pane 💷 Tool Bar Swing Controls label Label Image: Button Toggle Button 8[™] Button Group Check Box Radio Button
 1 2 3 4 5 6 7 8 9 合計 🖃 Combo Box 📑 List Text Field 1 1 1 1 1 1 1 1 1 1 d1 t× Text Area D Slider Scroll Bar Password Field Progress Bar Formatted Field 1 1 1 1 1 1 1 1 1 1 d2 💷 Spinner Separator T Text Pane 🖄 Editor Pane ੀਰੀ Tree Table カウント 消 Swing Menus Swing Window jButton2 [JButton] - Properties × Properties Events Code Properties action <none> Control background font Yu Gothic UI 12 Plain foreground [0,0,0] icon <none> mnemonic text 消 toolTipText Other Properties UIClassID ButtonUI jButton2 [JButton] •

表示されるボタン上の文字は「Properties」の「text」欄へ入力すると変更可能です。

手順 5: AiSee で操作する全てのコンポーネント(ボタン等の部品)の名前を設定

部品を並べ終わったら、コンポーネントをひとつずつ選択して「name」欄に名称を入力します。

例えばボタンなど、後でシナリオを付ける部品は必ず「name」の入力が必要となります。

図左の「合計」のように、表示するだけで他に機能がない部品については「name」の入力は任意です。

下図では部品に「A1」という名前を付けています。入力後、Enterを押して変更を確定させてください。

Board.java \times \bigcirc \checkmark \Box	Palette \times	-
History	Swing Containers	
, , , , , , , , , , , , , , , , , , , ,	Panel 🛅 Tabbed Pane	📜 Split Pane 🔜 Scroll Pane
le components in an area hold Shift and drag mouse over the	💷 Tool Bar 🛛 🛱 Desktop Pane	🛅 Internal Frame 🔳 Layered Pane
	V Swing Controls	
	label DK Button	Toggle Button Check Box
소란		
	jTextField1 [JTextField] - Properties $ imes$	-
1 1 1 1 1 1 1 d1	Properties Events Code	
	margin	[2, 6, 2, 6]
	maximumSize	[2147483647, 2147483647]
	minimumSize	[64, 22]
	name	A1 .
	navigationFilter	<none></none>
	nextFocusableComponent	<none></none>
	opaque	
	paintingForPrint	
	preferredScrollableViewportSize	[71, 22]
	preferredSize	[71, 22]
	requestFocusEnabled	
	scrollOffset	0
	scrollableTracksViewportHeight	
	scrollableTracksViewportWidth	
	selectedText	
	selectedTextColor	[255,255,255]
	selectionColor	[38, 117, 191]
	selectionEnd	11
	selectionStart	11
	verifyInputWhenFocusTarget	
	jTextField1 [JTextField]	Ø

(左図) NetBeans 上のアプリ画面作成例

(右図) 出力した Jar ファイルを取り込み、AiSee 上で表示させたもの

Team A	1 2 3 4 5 6 7 8 9 合計	[AS Board		_		×
Team B	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 カウント 消		Team A Team B	1 2 3 4 5 6 7 8 9 0 3 0 0 2 0 0 0 0 0 1 0 2 0 0	合計 5 4		
				カウント	消]	

手順6:JDK の設定

プロジェクト名の上で右クリック → プロパティ (Properties) を選択



下の画面左の「Sources」と「Libraries」2項目で設定が必要になります。

まずは「Sources」の中の「Source/Binary Format」のプルダウンで JDK8 を選択します。

※元々JDK8 が選択されている場合には手順6の操作は不要です。手順7に進んでください。

Categories:					
 Sources 	Project Folder:	C:¥Users¥terumin¥	Documents¥NetBeansP	rojects¥ScoreBoard	
Libraries	Source Package	Folders			
 Build Compiling 	Pa	ckage Folder		label	Add Folder
 Packaging Deployment 	src	ekage rolder	Source Packages	Laber	Remove
 Documenting Run 					Move Up
 Application Web Start License Headers 					Move Down
 Formatting Hints 	Test Package Fo	lders:			
	Pa	ckage Folder	1	Label	Add Folder
	test		Test Packages		Remove
					Move Up
					Move Down
	Course (Bisson F				In aluadas (Fueluadas
	Source/ Binary F	ormat: JDK 16			includes/ Excludes
	Profile:	JDK 2			
	Encoding:	IDK 8			
		JDK 9		_	
		JDK 10		OK	Cancel Help
	or of the test of test	JDK 11			
		JDK 12			
ור: C:#Users¥terumin¥Doo לום to C:¥Users¥torum	cuments#NetBeansProjects#Sco in#Documents#NetBeapsProject	reBoard¥ JDK 13			
) COPY.	In Francis Francisco Ballsri Oject.	JDK 14			
ar: C:¥Users¥terumin¥Do	ocuments¥NetBeansProjects¥Sc	oreBoarc JDK 15			
is application from the	command line without Ant, t	ry: JDK 16			

次に「Libraries」を選択し「Manage Platformes...(プラットフォームの管理)」をクリック、

次画面で「Add Platform」をクリックします。

	Jources	Java Platform:	JDK 16 (Default)	Manage Platforms
0	Libraries	Libraries Folder:		Browse
0	Build Compiling Packaging	Compile P	ocessor Run Compile Tests Run Test	5
	 Deployment 	Compile-time L	braries:	
0	 Documenting Run 	Modulepath		+ <u>E</u> dit
٥	Application	Classpath		+ Remo <u>v</u> e
0	License Headers			Move Up
ő	Formatting			Move Dowr

🟮 Java Platform Manager			\times
Use the Javadoc tab to register the A Click Add Platform to register other J	PI documentation fo ava platform version	or your JDK in the IDE. ns.	
<u>P</u> latforms:			
🗁 Java SE	Platform <u>N</u> ame:	JDK 1.8	
🖳 JDK 1.8	Platform Folder:	C+Program Filer¥lava¥idk1.8.0.192	_
JDK 16 (Default)	-	C++Togram Thestbava+juk 10.0_152	
	Classes Sou	urces Javadoc	
	Platform Classpa	ath:	
	C:¥Program	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥resources.jar	
	🗟 C:¥Program	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥rt.jar	
	C:¥Program	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥sunrsasign.jar	
	C:¥Progran	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥jsse.jar	
	C:¥Program	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥jce.jar	
	C:¥Program	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥charsets.jar	
	C:¥Program	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥jfr.jar	
	C:¥Progran	n Files¥Java¥jdk1.8.0_192¥jre¥classes	
	C:¥Program	n Files¥Java¥jdk1.8.0_192¥jre¥lib¥ext¥access-bridge-64.jar	
	C#Program	n Files#Java#jdk 1.8.0_192#jre#liD#ext#cldrdata.jar	
Add Platform			
		Close <u>H</u> elp	,

「Java Standard Edition」を選択し Next をクリックします。

🔾 Add Java Platform		×
Steps	Select platform type	
 Select platform type Choose Platform Folder 	Select <u>p</u> latform type to install: Java Standard Edition Remote Java Standard Edition Download OpenJDK (via Foojay.io Disco API)	
	< <u>Back</u> Next > Einish Cancel H	elp

「jdk1.8.0_192」を選び、Next \rightarrow 次の画面で Finish を押します。

🔇 Add Java Platform							×
Steps	Choose Platforn	n Folder					
Select platform type Choose Platform Folder Distform Name		Look In:	Java			× 1	15
5. Platform Name	最近使った項	> jdk-16.0	0.1				
		> jre1.8.0	_192				
	デスクトップ						
	לעג‡אא						
	PC						
		File Name:	C:¥Program Files¥lay	/a¥idk1.8.0_192			
	ネットワーク	Files of Type:	Platform Folder	langen nore_rsz		~	-
			< <u>B</u> ack	Next >	<u>F</u> inish Cance	! <u>H</u> e	lp

「Libraries」画面に戻ると、Java Plattforme で「JDK1.8」が選べるようになっているので、 「JDK1.8」を選択し OK を押します。JDK の設定は終了です。※JDK8 と 1.8 は同じものです。

Sources	Java Platform: JDK 1.8	✓ <u>M</u> anage Platforms
Libraries	Libraries Folder:	Browse
 Compiling Packaging Deployment 	Compile Processor Run Compile Tests Run Tests	5
Documenting Run	Modulepath	+ Edit
 Application Web Start 	Classpath	+ Remove
License Headers Earmatting		Move <u>U</u> p
 Formatting Hints 		Move Down
	Compile-time libraries are propagated to all library categories.	
	Ruild Decendencies	

手順7:プロジェクトをビルドする

画面左上のプロジェクト名の上で右クリックし「ビルド(Build)」を押します。

ビルドが成功したことを確認してください。

※既に1度でもビルドしたことがある場合は「消去してビルド(Clean and Build)」を推奨します。

プロジ	ェクト × ファイル	サービス	——————————————————————————————————————
	CSSample ^{III} ソース・ノ	新規	>/
		ビルド	
		消去してビルド	
	▶ フイノフラー 主■ JDK 1	消去	Α
+	WebSamp	Javadocの生成	
		実行	
		デバッグ	
		プロファイル	
出力	- CSSample (iar) × HTTPサーバー・モニター	
	ant -f C:¥¥U	sers¥¥TEST¥¥Documents¥¥NetBean	
	init:		
	deps-jar:		
	Created dir:	C:¥Users¥TEST¥Documents¥NetBe	ansProjects¥CSSample¥build
-	Updating pro	perty file: U:#Users#IESI#Docu	ments≇NetBeansProjects≇USSample≇build≇built-jar.properties ===Dusisst=¥CSS===ls%build%slasses
	Created dir:	C:¥Users¥TEST¥Documents¥NetBe	ansprojects#co3ample#build#classes ansDrojects#CSSample#build#empty
	Created dir:	C:¥Users¥TEST¥Documents¥NetBe	ansProjects¥CSSample¥build¥generated-sources¥ap-source-output
	Compiling 2	source files to C:¥Users¥TEST¥	Documents¥NetBeansProjects¥CSSample¥build¥classes
	compile:		
	Created dir:	CI#Users#IESI#Documents#NetBe	ansProjects≇USSample≇dist Natoananova inataYCCCanalaYbuild
	Nothing to c	ne to c.#osers#iEsi#bocoments#	Nerpeanski olecis#cosamkie#polio
	Building jar	<pre>- C:¥Users¥TEST¥Documents¥NetB</pre>	eansProjects¥CSSample¥dist¥CSSample.jar
	To run this	application from the command I	ine without Ant, try:
	java -jar "C	:¥Users¥TEST¥Documents¥NetBean	sProjects¥CSSample¥dist¥CSSample.jar"
	jari 尼山尼武市(本	計時間・ (1秒)	
		al estilate (449)	

手順 8 : Jar ファイルが作成できたことを確認

エクスプローラー→「ドキュメント」→「NetBeansProjects」フォルダ→ 作成したプロジェクト下の「dist」フォルダ内に「〇〇

O.jar」ファイルが作成されていることを確認します。

この Jar ファイルを 3.3.1 章の【図 2】【図 3】の手順で、AiSee で使用できるように取り込みます。

📙 » PC » ドキュメント » NetBeansProjects »	CSSample 🕨 dist
│ 名前	更新日時
CSSample.jar	2020/08/3 2020/08/3

補足: Jar ファイルを AiSee に取り込み、ボタン等にシナリオをセットした後に画面の修正やカスタマイズが必要となった場合には、 AiSee で作成中のプロジェクトの「画面定義」下のファイルと、「ライブラリ定義」下の Jar ファイルの 2 点を削除し、改めて更新した Jar ファイルを取り込めば問題なく作成済みのシナリオが適用されます。

7 既存プロジェクトをインポートする方法

AiSee のプロジェクトファイルは、「〇〇〇.isp」という拡張子のファイルです。 この章では既存の「〇〇〇.isp」ファイルをインポートする方法を説明します。

1:「ISeeProjectInformation」を右クリックし、「プロジェクトのインポート」をクリックする。



2:対象の Isp ファイルを選択する。

※本資料と同列の「AiSeeProject」フォルダ内にサンプルファイル(CSSample.isp)があります。

Import			×
← → × ↑ 🖡 >	PC > デスクトップ > demo v む	demoの検索	P
整理・ 新しいフォルタ	-	88 · 🔳	0
	名前	更新日時	^
🔓 3D オブジェクト	CSSample.isp	2020/08/31 15:16	
🎍 ダウンロード	WebSampleAiSee.isp	2020/08/31 13:06	
デスクトップ	WebSampleAiSee_0731.isp	2020/07/31 16:53	
ひ ドキュメント	✓ <		>
771	ル名(N): CSSample.isp ~ 〕	プロジェクトExportファイル(*.isp) 開く(Q) マ キャンセル	×
AS ISAD	– 🗆 X		
プロジェクト名称 プロジェクトルートディし デフォルト暗号化キー	: CSSample ンクトリ: C/Users/TEST/Documents/ISeeApp :		

3:インポートされたことが確認できたら完了です。



8 繰り返し(ループ)処理の説明

例1:

「結果」の合計を求めたい場合のループ処理を流れ図(フローチャート)で表すと以下のようになります。



例2:

「結果」の平均値を求めたい場合のループ処理を流れ図(フローチャート)で表すと以下のようになります。



例3:

DBを使用して「検索」したい場合のループ処理を流れ図(フローチャート)で表すと以下のようになります。

9 便利な Tips

• F1キー

プロジェクトマネージャーを表示し、アクティブな状態にしてから「**F1**キー」を押すとブラウザが 起動しヘルプが表示されます。ショートカットキーの一覧などが確認できます。

• F5 キー

テストを実行する場合に使います。テスト実行したいプロジェクトのフォルダや、任意のシナリオ を選択した状態で「**F5**キー」を押すと、簡単にテスト実行ができるショートカットキーです。

• **F4**キー

デバックしたい場合に使います。 まずは、デバックしたいシナリオ上で右クリック→「デバックデータリンクに登録」をクリック します。



次に、実際にテスト実行(F5 キー)してください。

テスト実行が終わったら、デバックしたいプロジェクトのフォルダや、任意のシナリオを を選択した状態で「F4 キー」を押すと、以下のような画面が表示されます。 矢印の部分をクリックすると、テスト実行結果の細かい内容を確認することができます。 (上が最新のログ、下が古いログです)

AS ISAD_LogViewer		_		×
プロジェクト名称 : ISeeSample2				
Iseel og(HTML) Iseel og(Text) JavaErrorl og				
isceedginney isceedgickity outdenticeog				
No 名前	サイズ	軍	新日時	
1 20221004 171939 full html	0k	2022/10/	04 17:19:3	a 🔺
2 20221004_171939.html	0k	2022/10/	04 17:10:3	<u>q</u>
3 20221004 171539 full html	0k	2022/10/	04 17:15:3	<u>,</u>
4 20221004 171539 html	0k	2022/10/	04 17:15:3	9
5 20221004 171425 full html	0k	2022/10/	04 17:14:2	5
6 20221004 171425 html	0k	2022/10/	04 17:14:2	5
7 20221004 165919 full html	1k	2022/10/	04 16:59:2	3
8 20221004 165919 html	1k	2022/10/	04 16:59:2	3
9 20221004 165700 full.html	0k	2022/10/	04 16:57:0	8
10 20221004 165700.html	0k	2022/10/	04 16:57:0	8
11 20221004 165403 full.html	17k	2022/10/	04 16:54:4	0
12 20221004 165403.html	13k	2022/10/	04 16:54:4	0
13 20221004 164708 full.html	4k	2022/10/	04 16:47:1	5
14 20221004 164708.html	3k	2022/10/	04 16:47:1	5
15 20221004 164501 full.html	4k	2022/10/	04 16:45:0	8
16 20221004 164501.html	3k	2022/10/	04 16:45:0	8
17 20221004_164427_full.html	4k	2022/10/	04 16:44:3	4
18 20221004 164427.html	3k	2022/10/	04 16:44:3	4
19 20221004_164215_full.html	4k	2022/10/	04 16:42:2	3
20 20221004_164215.html	3k	2022/10/	04 16:42:2	3
21 20221004_163835_full.html	4k	2022/10/	04 16:38:4	2
22 20221004_163835.html	3k	2022/10/	04 16:38:4	2
23 20221004_163333_full.html	4k	2022/10/	04 16:33:4	0
再読込 削除			閉	じる

2022/10/04 Datalink START : ISeeSample2/ISeeDataLink/Calculation_Close
Step:1テーフル行モテルを作成して番号を付ける
Return:@行モデル Value:M{_LThis_Instance_=jp.co.betanet.isee.common.beans.DataModelBean@813f2f8, No=M{},種類=M{}, 相類=M{}, 相類=M{}
=M{}}
Step:3 文字列を結合して返す
Return:@結合文 Value:3+3
Step:4 「結合文」の内容をカラムヘセットする
Step:7 文字列を結合して返す(=結果)
Return:@イコール結果 Value:=6
Step:8「結果」の内容をカラムヘセット
PatternMatch:9 分岐処理を行う
Step:11 「加算」をカラムへセットする
Step : 12 文字列を結合して返す
Return:@計算種類 Value:加算!
Step: 22 行モデルの内容をテーブルへ追加する
Step: 23 リソース定義フォルダパスを取得する
Return: @ファイルパス Value: C:¥Users¥terumin¥AppData¥Roaming¥ISeeIDE¥ISeeRun¥ISeeSample2/ISeeResources/
Step:24 リソース定義フォルダにDBファイル名を結合してDBファイルパスを作成する
Return: @ファイルパス Value: C:¥Users¥terumin¥AppData¥Roaming¥ISeeIDE¥ISeeRun¥ISeeSample2/ISeeResources//Oct_DB.db
Step: 25 ファイルパスからデータベースコネクションを作成して返す
Return: @コネクション Value: 0bd1d72b-2d28-4873-8a4e-11a0e679d3a2
Step: 26 データベースを検索する
Return: @レコードセット Value : operator, contents, result
Step: 27 挿入マップを作成する
Return:@挿入マップ Value:M{result==6, contents=3+3, operator=加算!}
Step: 28 データベースへ挿入する
Return : Parameter Value : null
Step: 29 コミット
Step: 30 コネクションを切断
2022/10/04 Datalink FND: I SeeSample2/ISeeDatal ink/Calculation. Close

デバック登録したシナリオを解除したい場合や、どのシナリオを登録したか確認したい場合は 「シナリオ定義」の上で右クリックし、「デバックデータリンク一覧表示」を開くと確認や編集が 可能です。

※データリンク=シナリオを指します。

• シナリオ作成画面のアイコン

シナリオを作成する際に、通常は「Section」を右クリックして作成しますが、「Section」の上にある アイコンからもリレーションやサブシナリオを新規作成する事ができます。(下図)

⊀8 リレーションマネージャ:TopPanel_行選択		-		×
	DL一覧追加			
P □ Section Sec	シナリオ説明	変数定義		
	変数名		種類	
│ │ │ データベースレコードのカラムへ内容を取得して返す	バリュー	任意の	<u> </u>	
│ └ │ 複数の代入処理を行う	<u>ユーザーID</u>	任意の	<u> 번</u>	
	<u> ITTTル</u>	性恵の	꼬	
	タイトル: Copyright(c) 2008 Betane		集)ペータネッ シーマ VI Rights Reso	h H
	瞳	定 保	存開	じる
TopPanel_行選択				

10 Q&A

- ◆ **Q.** シナリオ内にリレーションを作成したいが、何度「登録ボタン」を押しても反映されず、作成が できません。
 - A.リレーション作成前に、下図の「Section」の部分を左クリックで選択してください。 Section の周囲に細い青枠が表示されたことを確認してから、リレーションの新規作成を行って ください。(下図)

糸 リレーションマネージャ:TopPanel_行選択			_		×
		DL一覧追加			
♀ □ Section ○ 認知中のある行のニーゴル行ちごりた取得して返す	₹	シナリオ説明	変数定	義	
	1000	変数名		種類	
− □ データベースレコードのカラムへ内容を取得して返す	1000	バリュー	任意の	型	
□ 複数の代入処理を行う	1000	ユーザーID 行モデル	<u>仕</u> 意の 任音の)型 刑	
	1000	11 0 7 10	112,22,00	· ⊥	
	1000				
	1000	タイトル:			Ê
		Section			-
		Ai	S	株)ベータネ eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee	•
)	_ 9	copyright(c) 2009 Bea		777 B	BI" Z
		DE	ERE 14	k1£ 6	ຍບລ
TopPanel_行選択					

- ◆ Q.「Section」をクリックしてからシナリオを作成していますが、それでも作成できません。
 - A. 念のため、AiSee の再起動をお試しください。リレーションマネージャ画面を閉じて、 プロジェクトを保存してから、「閉じる」で終了し再起動してください。(下図)

AS プロジェクトマネージャ					-	_		\times
ISeeProjectInformation	ファイル情報							
← □ 面面定義	名前	種類	サイズ		更新日	時		
 □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	SecondPanel_OK.dtl	dtl	12k	2022/	09/13 1	4:20:20	1	
▶ 🗖 メニュー定義	SecondPanel_初期化.dtl	dtl	5k	2022/	09/13 1	2:42:06	1	
● □ イベント割当定義	TopPanel_Addボタン.dtl	dtl	54k	2022/	09/22 1	7:55:56	1	
● □ ルール定義	TopPanel_Delボタン.dtl	dtl	41k	2022/	09/22 1	7:32:37	1	
● □ ワイノフリ定義	TopPanel_初期化.dtl	dtl	61k	2022/	09/22 1	7:39:11	1	
► ISeeSample2	TopPanel_更新.dtl	dtl	44k	2022/	09/26 1	1:01:43	1	
	TopPanel_検索.dtl	dtl	58k	2022/	09/26 1	0:09:31	1	
	TopPanel_行選択.dtl	dtl	17k	2022/	09/26 1	4:35:16	1	
	▲ ▼ データリンク編集履J		ブックマ	ィーク	情報]		
	プロジェクトター		タロンス	7夕		117		
	CSSample T	opPan	el 行選	///山 択	TopPa	anel ífi	選択	
	CSSample T	opPan	el_Add	ボタン	TopPa	anel_Ac	idボタ:	ンヨー
	CSSample T	opPan	el_初期	íŁ 🔄	TopP	anel_初	期化	
	CSSample	opPan	el 更新		TopP:	anel 🛡	新	
th) x-bxyr								
3	Copyright(c)	2009 Beta	net Corpora	tan. All Ri	ghts Re.	s vet.		
検索キー:	▲ 検索 ■ 絞込み	言羊約田:	検索			保存	毘	じる
Ver.1.0.22.901					C			

◆ Q. シナリオの検索がうまくいきません。

A.下図のように「文章対応」のチェックを外して検索をお試しください。

「文章対応」にチェックがあると、あいまい検索も可能であるため、基本的にはチェック有での 検索を推奨いたしますが、チェックの有無で検索結果が異なる場合がございます。

併せて、例えば「データを削除する」と検索されている場合には「データ 削除」など キーワードのみでの検索もお試しください。(下図)

また、複数回の検索を試されたい場合には、一度検索した後に「絞り込み」のチェックを外せば 再度検索ができる状態になります。

AS ISAD_DataRelationMethod_History	×
分類階層表示	▼【履歴
 ■削除 ■ JAVA Swing ■ テーブルのカレント行を削除する ■ テーブル行モデルで指定した行を削除する ■ テーブル行モデルで指定したインデックスの行を削除する ● テーブル行モデルで指定したインデックスの行を削除する ● 「ジータ構造 ■ ISee ツリーエレメントを切り離して、ほかのツリーモデルへ結合 ■ のリーノードを削除する ■ カレントのツリーノードを削除する ■ すべてのツリーノードを削除する ■ 1See リストの内容(要素)をすべて削除する 	 隔歴
 ビ 文章対応 検索 ビ 絞込み Top10 ▼ 詳細 削除 	



Q. テスト実行したら、このようなエラーが出てしまいました。

ISee	×
x	起動時にエラーが発生しました、 ネットワーク環境、及び、ISeeライセンスが有効か確認してください。
	OK

A1. テスト実行前に、起動設定の見直しを行ってください。

が

特に、プロジェクトフォルダの上で右クリック→「起動設定」→ 画面中央の「画面 ID」の部分

正しく設定されているかどうかご確認ください。<u>5.3.1 章</u>【図 17】 【図 18】をご参照ください。

- A2. もし上記の方法で解決できなかった場合は、一度プロジェクトをエクスポートして、新たな プロジェクトとしてインポートしてください。
- ◆ Q. テスト実行(F5)を押しましたが、何の反応もありません。

A.起動設定も正しいのに、テスト実行ができなくなってしまった場合、一度 AiSee を 終了し、念のため PC の再起動も行い、再度 AiSee を起動してテスト実行をお試しください。 改善する場合があります。

◆ **Q.** いくつかのシナリオをフォルダ分けする方法はありますか?

A. シナリオを複数のセクションに分ける事ができます。

「Section」の上で右クリック→「セクションの作成(下へ)」で、新たなセクションが作成 できます。

下図では、「データベース関連」というセクションを作成しました。

作成した新たなセクションの下にシナリオを追加していくと、新たなセクション名の頭に フォルダのマークが現れ、複数のセクションが見やすくなります。(下図)



◆ Q.文字列を結合するシナリオを作成しましたが、実行結果に「null」が余分に入ってしまいます。

A.シナリオリレーション作成時、不要なパラメータが入ってしまった可能性があります。下図の パラメーター定義の No3 の部分を選択し、削除すれば、null が消えます。

10 × 111+111 × ×							
<i>州</i> S ジナリオリレージョン					-	Ц	×
タイトル: 文字列を結	拾して返す(=結果)					処理違	訳
説明・コメント:							
戻り値: ローカル3	変数 ▼ イコール結果						選択
パラメーター定義	jên jî	166 1	t	画面モデル指定			-
No 種類	値			画面モデル /Calculation	ı		詳細
1 定数 = 2 画面モデル Calcul	lation/result_txa/Value		_	名前	租	(漢頁	
3				first_txf	JTextField		
· · · · · · · · · · · · · · · · · · ·				operator_cmb	JComboBox		
				second_txf	JTextField		_
				calc_btn	JButton		_
				close_btn	JButton		_
				result_txa	JTextArea		
				値を設定			
機能説明 パラメータ説!	ад			インスタンスを設定			
1・44~」たい文字列の1個	Ξ		F				
 1: 福音 O/C いく デバット (個) 2: 結合したし文字別の2個 	ロ。 日、3個以上結合したし時には パラッ	/ークー友追hul					
て指定する。			'HI				
						登録	戻る

◆ Q.どこを見ても間違いがないように思うのですが、ほかに探すところはありますか?

A.定数で不要な改行が入っていたり、漢字の「二」をカタカナの「ニ」などの書き方に 間違いがないか一度確認をしてみましょう。

◆ Q.NetBeans で作成したボタンにシナリオを付けたが、シナリオは正しいのに動かないです。

A.作成したボタンなどの部品の名称に、「.(ピリオド)」など使用できない文字が入っていないか ご確認ください。ピリオドは消去するか、アンダーバーに変更してから再度お試しください。

- ◆ Q.NetBeans で画面を作成し、その後画面を修正したが、AiSee 上で見ると修正が反映しないです。
 - A. AiSee の中で、同じクラス名の画面がある場合に競合してしまう可能性があります。 そのため NetBeans 上で、画面の名前を別のものに変更して、上手くいくかお試しください。

下の例であれば、変更部分は Project 下の「Board.java」と画面右下の Properties 内の name にある「Board」です。変更後も、2 つが同じ名前になるよう変更をお願いします。



Design History	Internal Frame		hand the second s		
		Layered Pane			
	✓ Swing Controls				
e the context menu to access available useful actions for the selected component	abei Label (EK Button	Toggle Button	Image: Image: Box ■ Check Box	
	®- Radio Button	8– Button Group	Combo Box	📑 List	
	Text Field	t× Text Area	Scroll Bar	ѿ Slider	
Score Board	Progress Bar	Formatted Field	Password Field	💷 Spinner	
1 2 3 4 5 6 7 8 9 合計	Separator [T Text Pane	🖄 Editor Pane	🖽 Tree	
	Table				
	Swing Menus				
	File Menu Bar		🖃 Menu		
	- Menu Item		Menu Item / Che	eckBox	
メンで 成	[JPanel] - Properties >	×			_
	Properties Event	ts Code			
🕏 Terumi 2022.	inheritsPopupMenu				
	inputVerifier		<none></none>		<u>~]</u>
	insets		[0, 0, 0, 0]		
	maximumSize		[32767, 32767]		
	minimumSize		[0.0]		-
	name		Board		
_	nextrocusableCompor	nent	<none></none>		<u> </u>
	paintingForPrint				
この部分を選択	preferredSize		[0.0]		
	[JPanel]				•

◆ Q.データベースにデータを書き込んだはずだが、DB Browser を見ても反映していないです。

A.DB Browser の最上部に、PC の中のどのデータベースを表示しているのか、場所が表示されています。もしこの表示が、

「 C:¥Users¥ユーザー名¥Documents¥ISeeAppl¥プロジェクト名¥ISeeResources 」 (以下、Documents下)

になっているようであれば、また別の、以下の場所のデータベースを確認し、更新時刻が 編集した時刻になっているかご確認ください。

「 C:¥Users¥ユーザー名¥AppData¥Roaming¥ISeeIDE¥ISeeRun¥プロジェクト名 」

(以下、AppData下)

※上記 URL は通常、編集が必要ない場所であるため隠しファイルになっています。まず Windows で 隠しファイルを表示する設定にしてから、アクセスをお試しください。

【 AppData 下の DB ファイルが、最新の更新時刻となっている場合】

DB は正常に更新されていますので AiSee 上のパス指定は正しいと考えられます。

通常は、Documents 下の DB を DB Browser で参照すれば更新されているはずですが、もし更新 されていなければ AppData 下の DB ファイルを DB Browser で開いて、DB が更新されているか ご確認ください。

【 AppData 下の DB ファイルが、最新の更新時刻となっていない場合】

AiSee 上で DB の指定した DB のパスに誤りがないかご確認の上、正しいパスをご指定下さい。 パスが正しいように見えるのに更新が反映しない場合には、下図のように絶対パス指定で DB の 場所の指定をお試しください。

<i>▲</i> S シナリオリレーション	– 🗆 ×
タイトル: リソース定義フォルダにDBファイル名を結合してDBファイルパスをf	作成する 処理選択
説明・コメント: 文字列を結合して返す	
戻り値: ローカル変数 ▼ ファイルバス	選択
パラメーター定義 追加 削除	↑ ↓ 定数指定 ▼
NO 催焼 I世 1 完計 Cill Isare) I世	定数入力 適用
2 定数 Oct_DB.db	C:\Users\ in\Documents\ISeeAppl\ ISeeSample_viSeeResources\
1:結合したい文字列の1個目。	
2:結合したい文字列の21回日。31回以上結合したい時には、ハラメーターを追加して引 。	
J	
	登録 戻る

AiSee の画面から DB ファイルを追加した際には、 エクスプローラーでは Documents 下に DB が 追加されますが、<u>更新や削除など実際の処理は、</u> AppData 下にコピーされた DB で実行されます。 ただし、AiSee 上で絶対パスで DB の場所を指定したい場合は、Documents 下の DB を指定して ください。