

OpenModelica超初級チュートリアル

5. オリジナルモデルの作成

Copyright (C) 2020 Shigenori Ueda
Released under the MIT license

<https://opensource.org/licenses/mit-license.php>

注意事項

- 本チュートリアルは以下の内容が理解できていることを前提としております。
「OpenModelica超初級チュートリアル1.解析モデルの作成と実行」
「OpenModelica超初級チュートリアル2.コーディング」
「OpenModelica超初級チュートリアル3.モデルのカスタマイズ1」
「OpenModelica超初級チュートリアル4.モデルのカスタマイズ2」
- OpenModelica1.14.1 (64bit – windows版)を利用して本チュートリアルは作成されています。

オリジナルモデルの作成

自分で一からモデルを作ってみましょう

オリジナルモデルが作れるようになる？

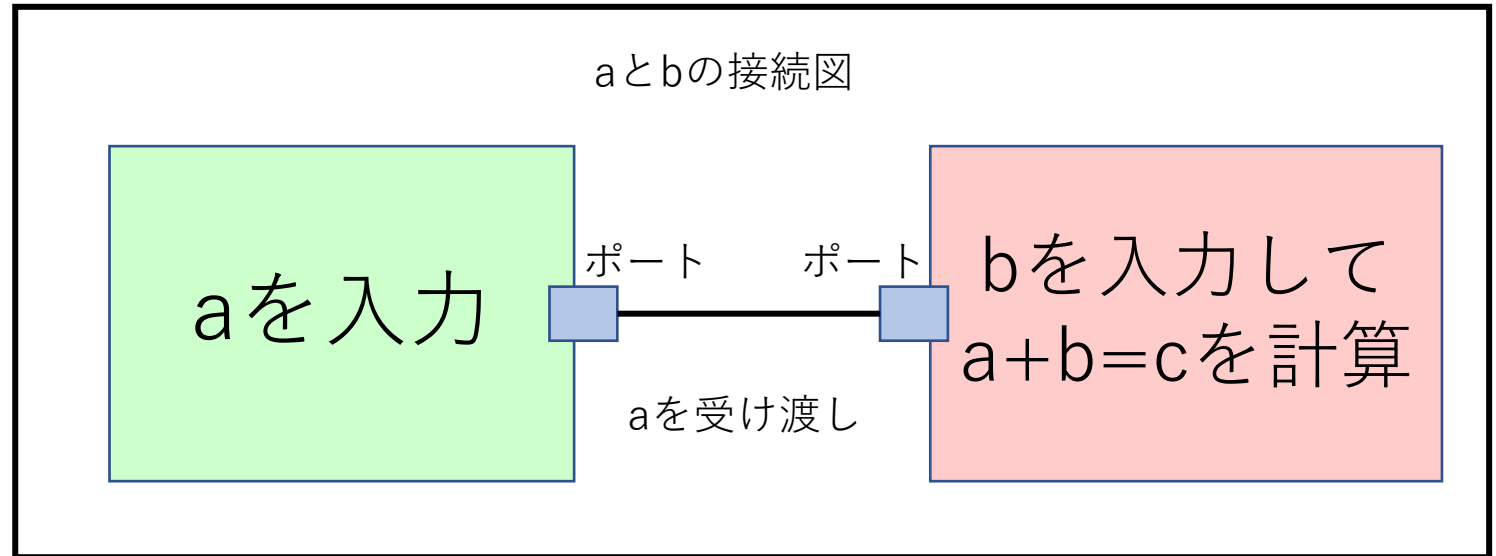
- ✓ 任意の数式を計算できるようになる
- ✓ オリジナルのライブラリを作れるようになる
- ✓ 既存ライブラリが何をしているか分かる

作成するモデル

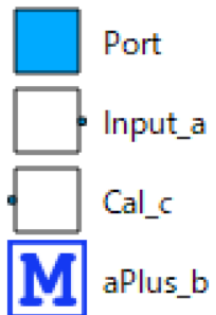
実数 a, b を入力し、 $a+b=c$ を計算するモデルを作成する

必要なクラス

- 実数を引き渡すポート
- a を入力するモデル
- b を入力して $a+b=c$ を計算するモデル
- a と b の接続図



作成するクラスとaとbの接続図



作成するクラスの詳細

各クラスを作成する際に必要なデータを以下に整理します
クラス名やパラメータ名などは任意です

表の上から順に作成します
まずは簡単に何が必要か確認してください

クラスの役割	クラスタイプ	クラス名	パラメータ名	変数名	コネクタ	equation
実数を引き渡すポート	connector	Port	実数 var	—	—	—
aを入力	model	Input_a	実数 a	—	Port	port1.var=a
bを入力 a+bを計算	model	Cal_c	実数 b 実数 a	実数 c	Port	a+b=c a=port1.var
aとbの接続図	model	aPlus_b	—	—	—	—

connectorの書式

connectorはクラス同士が参照する変数を定義したものです
以下の書式で表されます

connectorの書式

connector_コネクター名

受け渡ししたい変数の型_変数名1;

受け渡ししたい変数の型_変数名2;

⋮

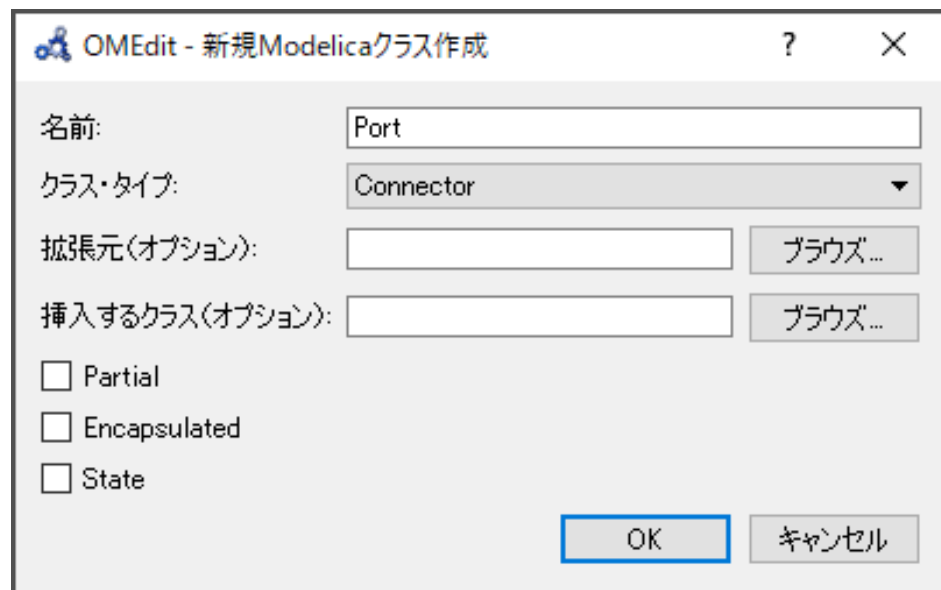
annotation(アイコン情報);

end_コネクター名;

connectorの作成1

connector「Port」を作成してみましょう。

- ① 「Modelicaクラス新規作成」から、名前を「Port」、クラス・タイプを「Connector」にしてモデルを新規作成してください



- ② テキストビューから以下のようにconnectorのコードを確認しましょう

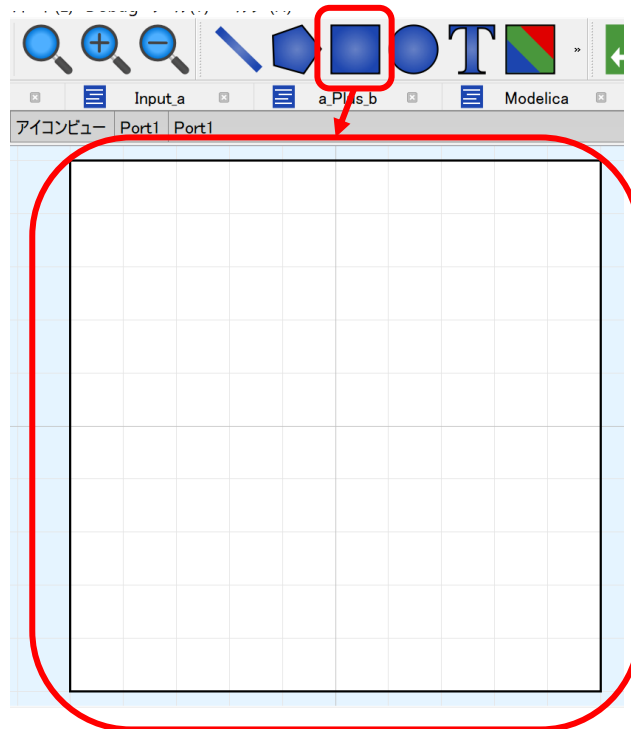


connectorの作成2

- ③ 受け渡しする変数を定義します
今回は実数を受け渡しするため、Real型で宣言し、変数名をvar(variableの略)にします

```
connector Port  
  Real var;  
end Port;
```

- ④ アイコンの設定を行うため、「アイコンビュー」から「長方形」を使って、表示範囲(coordinate System)全体に正方形を書いてください



connectorの作成3

- ⑤ 長方形を右クリックして、「特性」をクリックし
「塗りつぶしタイプ」を以下のように設定し、アイコンの色を定義してください。
これで「Port」が完成しました。

OMEdit - Rectangle 特性

Rectangle 特性

変換

原点 X: 0.00 原点 Y: 0.00 回転: 0.00

範囲

点1のX座標: -100.00 点1のY座標: 100.00
点2のX座標: 100.00 点2のY座標: -100.00

境界のスタイル

パターン: BorderPattern.None 半径: 0.00

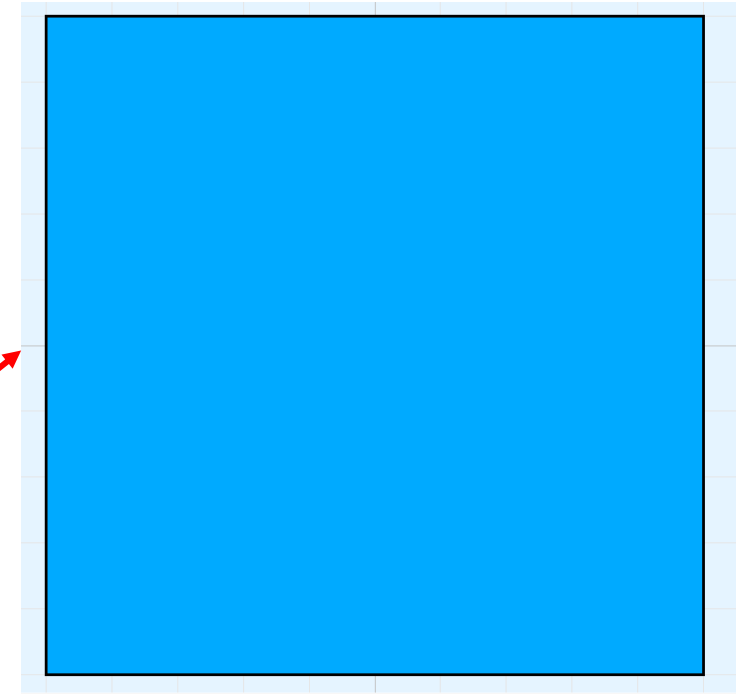
線のスタイル

色: ■ 色をピック
パターン: LinePattern.Solid
太さ: 1.50

塗りつぶしタイプ

色: ■ 色をピック
パターン: FillPattern.Solid

OK キャンセル 適用

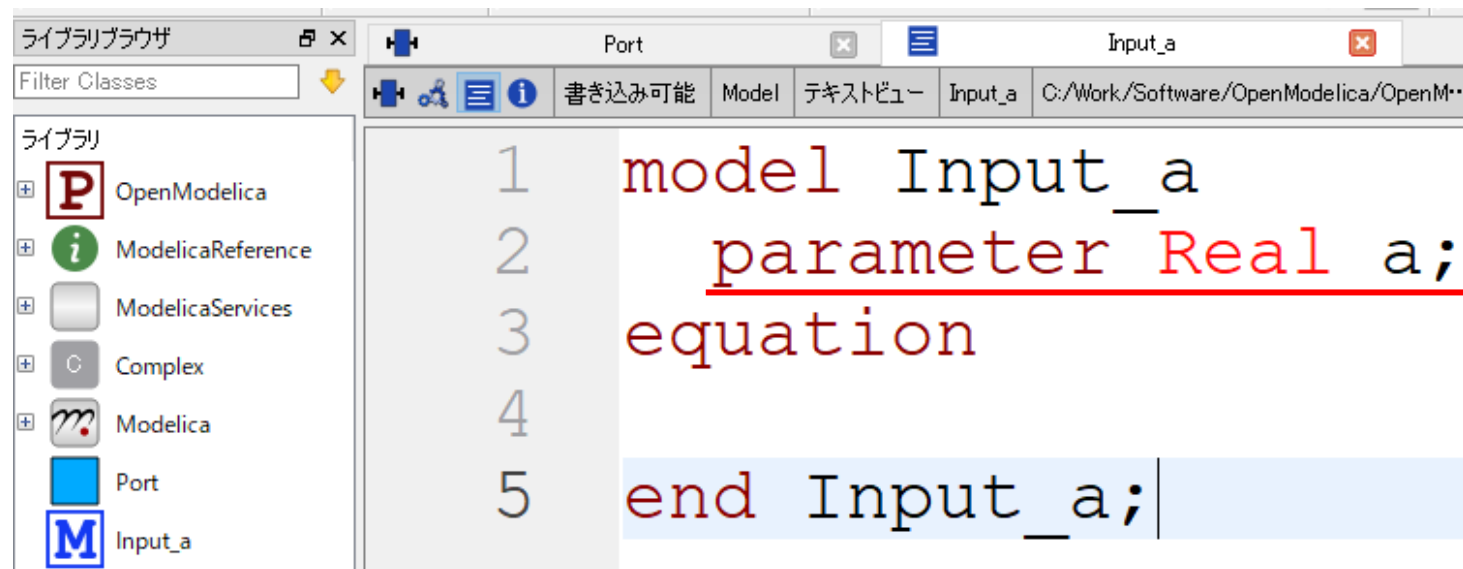


「aを入力」の作成1

以下の表をもとに「aを入力」するモデルを作成します

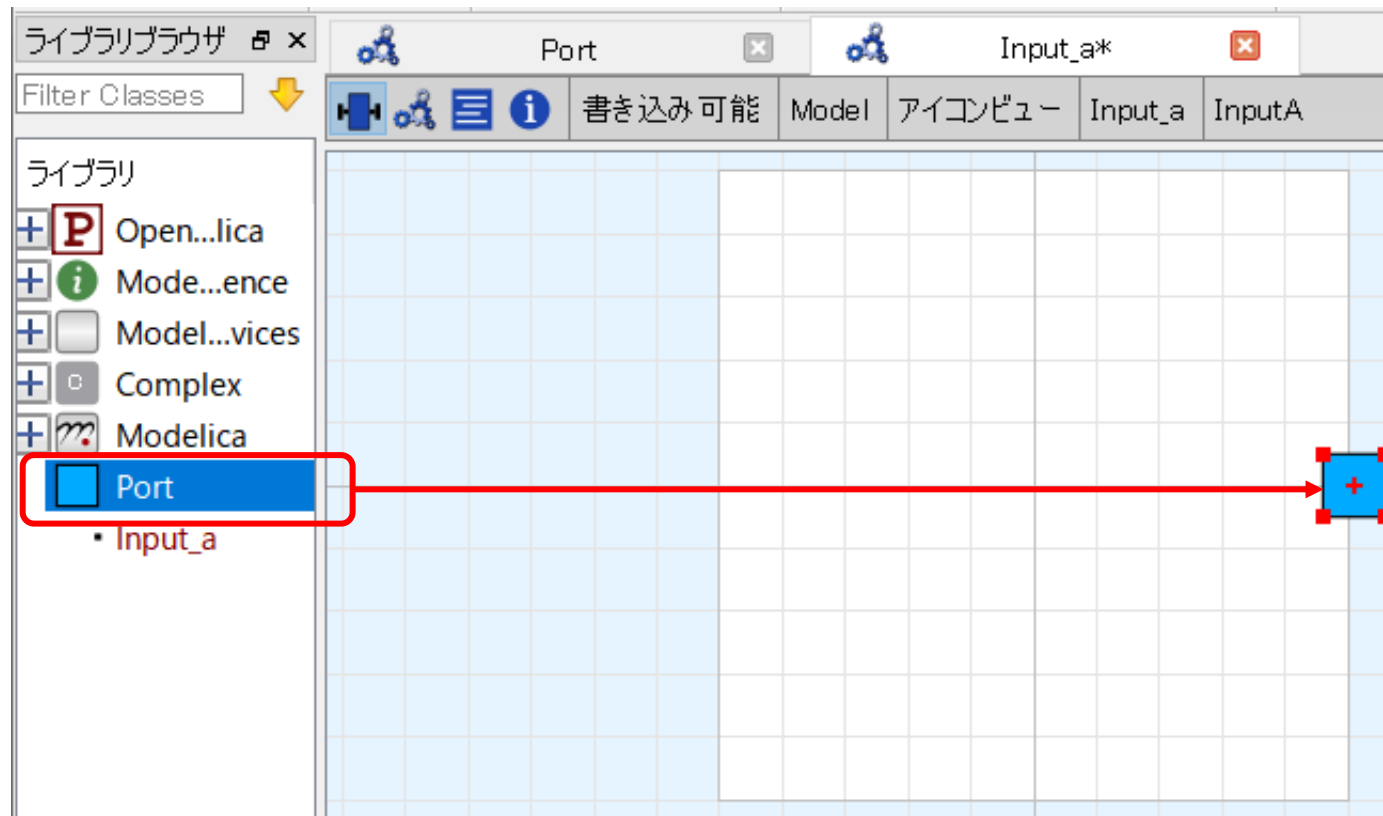
クラスの役割	クラスタイプ	クラス名	パラメータ名	変数名	コネクター	equation
aを入力	model	Input_a	実数 a	—	Port	a=port.var

- ① クラスタイプをModel、クラス名をInput_aにして新しいクラスを作成してください
- ② パラメータとなる実数「a」を追加してください



「aを入力」の作成2

③ tutorial4で行ったように、「アイコンビュー」から「Port」を追加してください



テキストビューを確認すると
「Port」のインスタンスが以下のように
作成されています

```
1  model Input_a
2      parameter Real a;
3  +  Port port annotat
5  equation
6
7  end Input_a;
```

入出力の関係式

ここでconnectorで受け渡しする値を受け取る方法を解説します。
インスタンス化されたconnectorの変数と、
宣言したパラメータや変数を等式で結びつけることで値を受け渡します。

connectorの変数と宣言した変数の接続

変数の型 `_` 変数名A;

connector名 `_` インスタンス名;

equation

インスタンス名.変数名B=変数名A

インスタンスの中身

connectorの書式

connector `_` コネクター名

受け渡ししたい変数の型 `_` 変数名B;

annotation(アイコン情報);

end `_` コネクター名

Memo

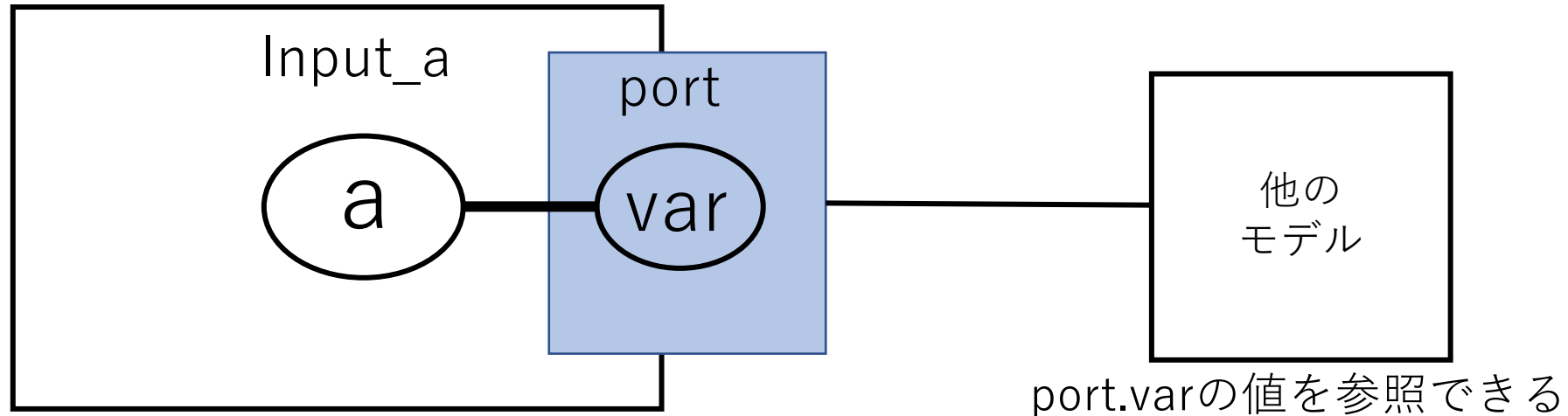
Modelica言語では、階層を「.(ドット)」で表します

「aを入力」の作成3

- ④ パラメータ「a」と、「port1」の変数「var」を結び付けるため、以下の式をequationセクションに記述します。

```
1 model Input_a
2   parameter Real a;
3   Port port annotation( ... );
5 equation
6   port.var = a;
7 end Input_a;
```

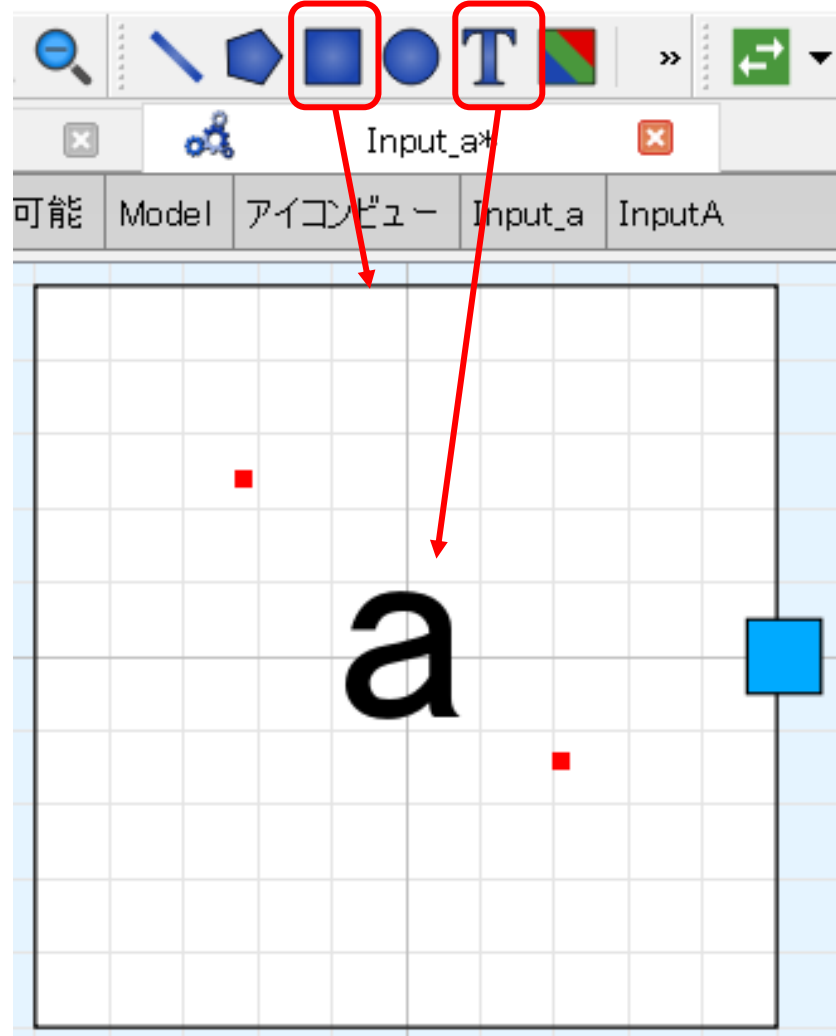
Input_aの値をPortによって他のブロックへ受け渡せるようになります。



コネクタの役割のイメージ

「aを入力」の作成4

⑤ アイコンを以下のように設定します。これで「Input_a」の完成です。



チェックしてモデルに問題が無いか確認することをお勧めします。
今後作成するモデルについても同様にチェックすることでデバッグが楽になります。

「bを入力、a+b=cを計算」の作成1

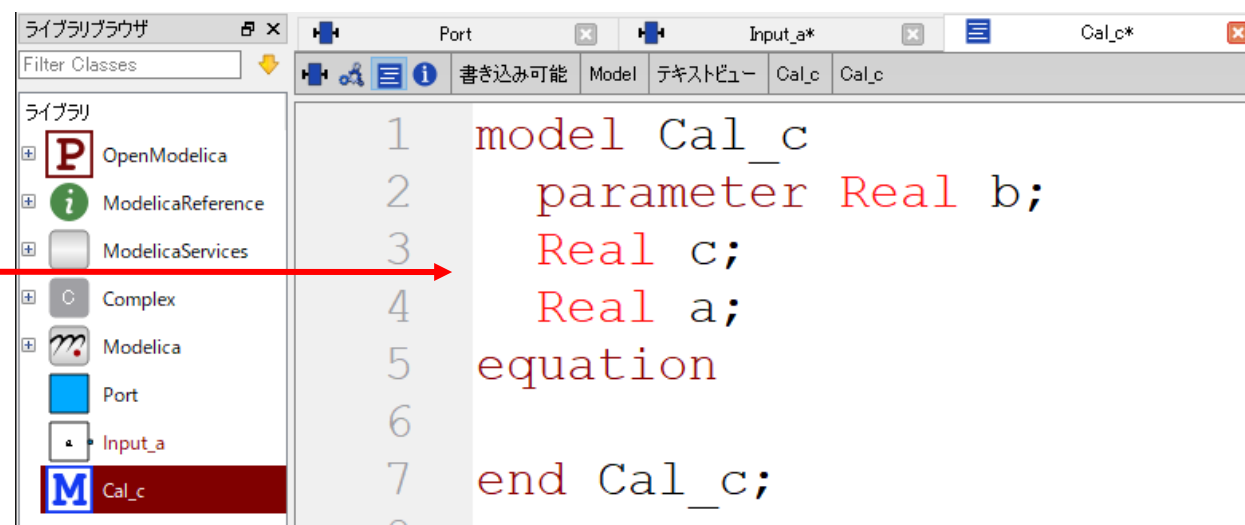
これまでと同様に、表を参考にモデルを作成します

クラスの役割	クラスタイプ	クラス名	パラメータ名	変数名	コネクター	equation
bを入力 a+bを計算	model	Cal_c	実数 b	実数 c 実数 a	Port	a+b=c a=port1.var

① 表を参考に新しいクラス「Cal_c」を作成してください

② パラメータとなる実数「b」を追加してください

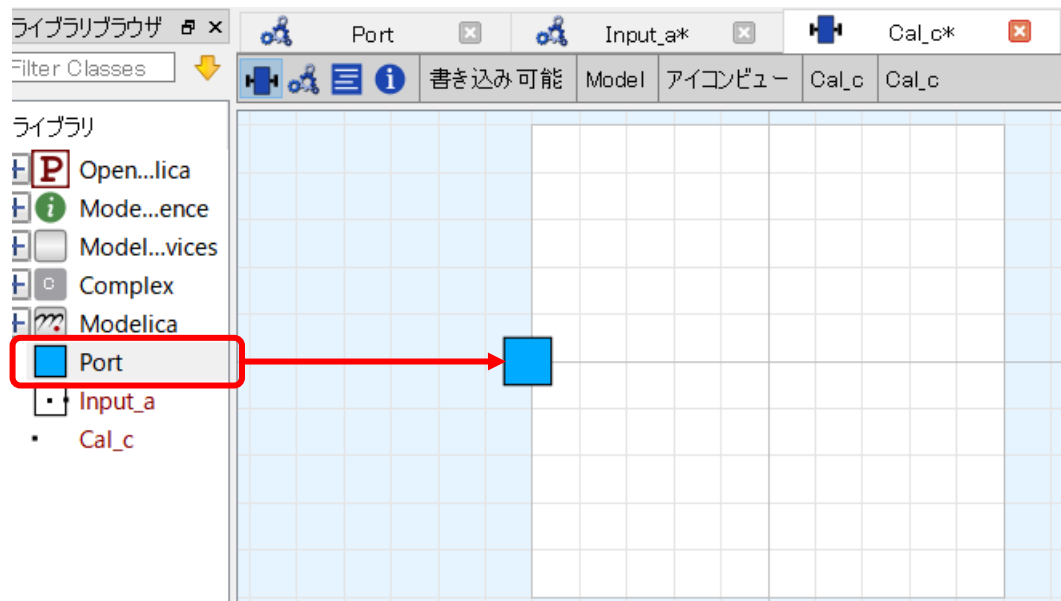
③ 変数となる実数「c」、「a」を追加してください
右図のように記述されると思います



```
1 model Cal_c
2   parameter Real b;
3   Real c;
4   Real a;
5   equation
6
7 end Cal_c;
```

「bを入力、 $a+b=c$ を計算」の作成2

- ④ 「アイコンビュー」から「Port」をドラッグ&ドロップして追加してください



- ⑤ equationセクションに、ポートからの値を受け取る等式「 $a = \text{port.var}$ 」を追加してください

```
1 model Cal_c
2   parameter Real b;
3   Real c;
4   Real a;
5   Port port annotation( ... );
7 equation
8   a=port.var;
9 end Cal_c;
```


「bを入力、 $a+b=c$ を計算」するモデルの作成3

⑥ 「 $a + b = c$ 」を追加してください

```
1  model Cal_c
2      parameter Real b;
3      Real c;
4      Real a;
5      Port port annotation( ... );
7  equation
8      a+b=c;
9      a=port.var; ← 入出力の関係式
10 end Cal c;
```

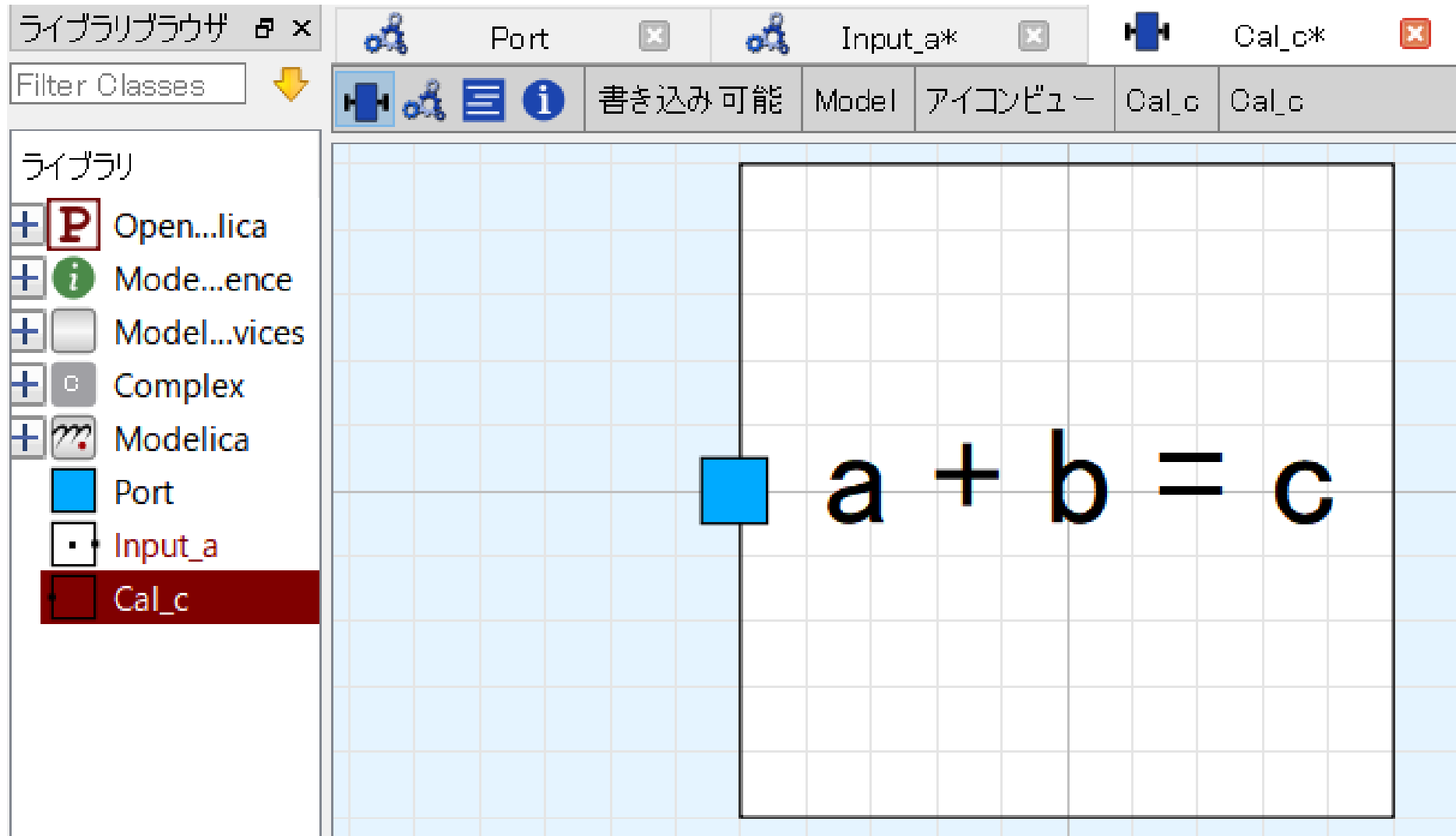
Memo

Modelica言語は、非因果的モデリングのため
方程式の左辺、右辺
等式の順番
は計算結果と関係ありません。

一般的に入出力の関係式は、equationセクションの一番下に
記述すると計算式と混在せず可読性の高いコードになります

「bを入力、 $a+b=c$ を計算」するモデルの作成4

⑦ Cal_cモデルのアイコンを以下のように作成してください。これでモデルは完成です。

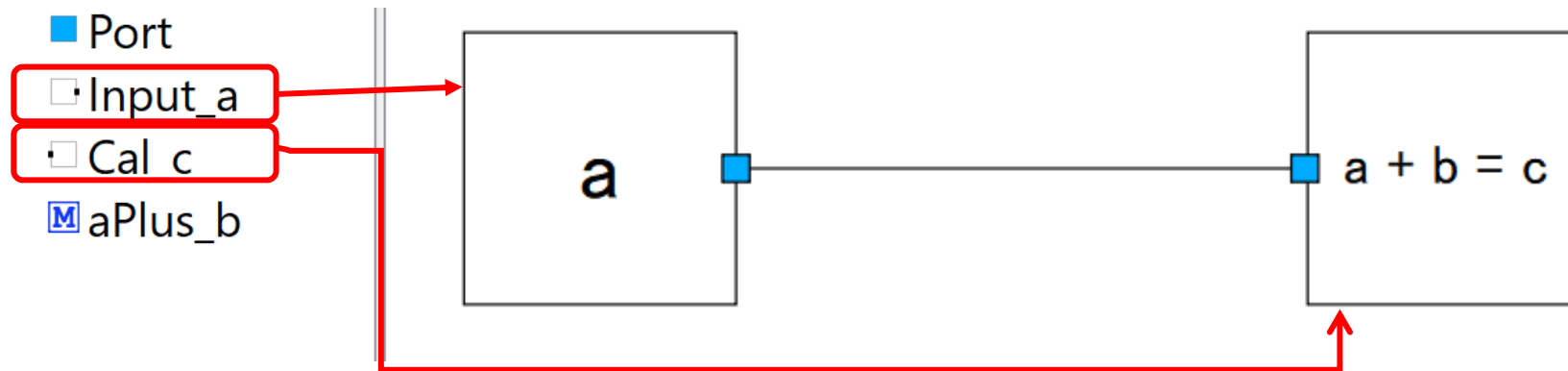


「a+bの接続図」の作成1

最後に、これまで作成した「Input_a」と「Calc_c」を接続するためのクラスを作成します

クラスの役割	クラスタイプ	クラス名	パラメータ名	変数名	コネクター	equation
aとbの接続図	model	aPlus_b	—	—	—	—

- ① 新規クラス「aPlus_b」を作成し、ダイアグラムビューに切り替え
「Input_a」と「Calc_c」のインスタンスを作成し、接続してください



- ② 「Input_a」のパラメータ「a」と「Calc_c」のパラメータ「b」に1を代入してください

- ③ チェックを実行し、問題なければシミュレートを実行し「c」の値が2になっていることを確認してください

Exercise

- ・チュートリアル内では「bを入力、 $a+b=c$ を計算」を一つのモデルで作成しました。しかし「bを入力」と「 $a+b=c$ を計算」のモデルをそれぞれ別に作成することもできます。

試しに作成してみてください。

- ・今回は4つのファイルを作成しました。これらを一つのファイルとしてまとめる方法があります。

チュートリアル6の「01_MakePackage」を参照して一つのパッケージとしてまとめてみてください。

その他—Modelicaにおける命名規則

以下のように命名規則を定めることで可読性が高く、変数のバッティングを防ぐことができます。

1. functionクラス以外のclass名は大文字で始める
2. 変数名は小文字で始める。しかし、一文字の場合は大文字にする場合もある
例. 自然長s_rel0, 温度の変数 T
3. 複数の単語からなる名前は、最初の文字は上記 1～2に基づいて決定し
残りの単語を大文字にする
例. 変数 onePlus,
4. アンダーバーは名前の最後に使う。
5. 同様のコネクタインスタンスが二つある場合は、正負を表すpとnやaとbなどを使い区別する
例. flange_a, flange_b