

OpenModelica超初級チュートリアル

3. モデルのカスタマイズ



“OpenModelica tutorial for beginner 3 CustomizeModel1” by UedaShigenori is licensed under [CC BY 2.0](https://creativecommons.org/licenses/by/2.0/)

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」
- 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。

モデルのカスタマイズ

既存のモデルを自分の欲しいモデルになるように改造してみましょう。

カスタマイズができるようになる？

- ✓ 既存ライブラリを自分好みに改造できる
- ✓ オリジナルのライブラリを作る足掛かりになる
- ✓ Modelica言語に詳しくなる

モデルの種類

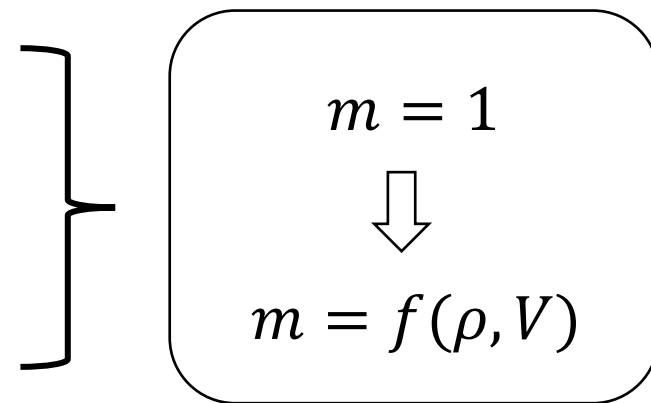
モデルは用途に応じて大きく以下に分類できます*。

- ・ モデル同士の接続関係を表す接続図(Connection Diagram, システムモデル)
- ・ モデルの構造を表す部品図

今回のカスタマイズはこちら

まずは接続図を対象に
ある変数を他の変数の関数にカスタマイズ
してみましょう。

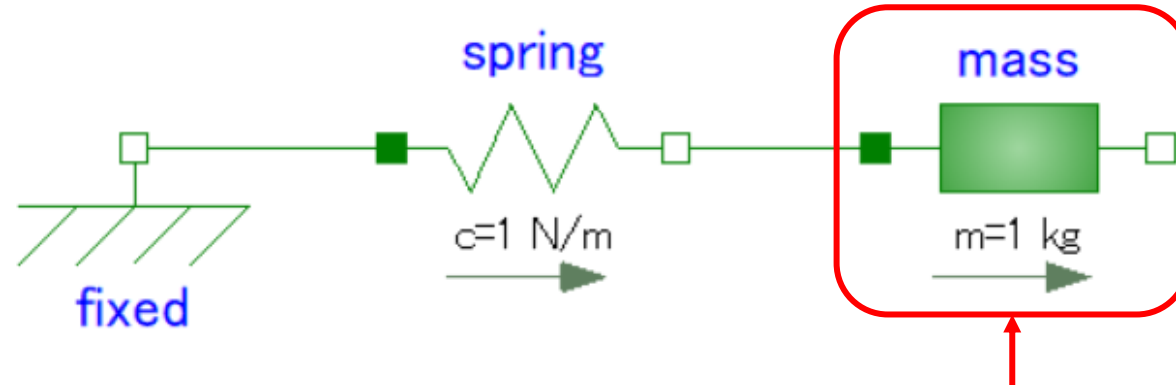
質量を密度と体積の関数に



* 接続図を使ってモデルの構造を表現する場合もあるため
あくまで便宜上の分類です。
ただ業務などで、部品図のことを話しているのか、接続図のことを話しているのか
混乱を生じる場合があるため分類しました。

カスタマイズ内容

チュートリアル1で作成したバネマスモデルのmassモデルの質量 m を密度 $\rho \times$ 体積 V にしてみましょう。



現在・・・質量を直接入力

カスタマイズ・・・質量を密度 $\rho \times$ 体積 V で表す

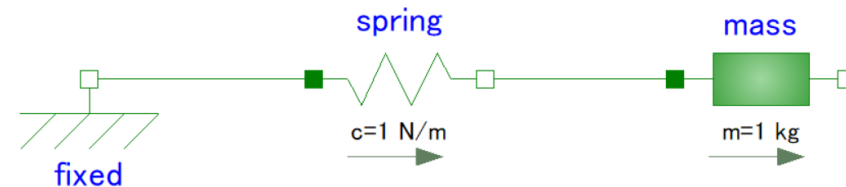
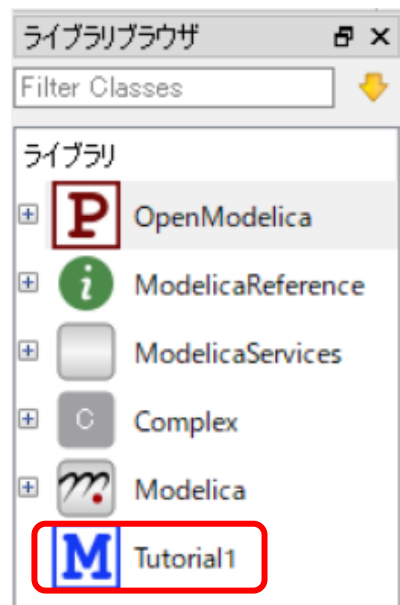
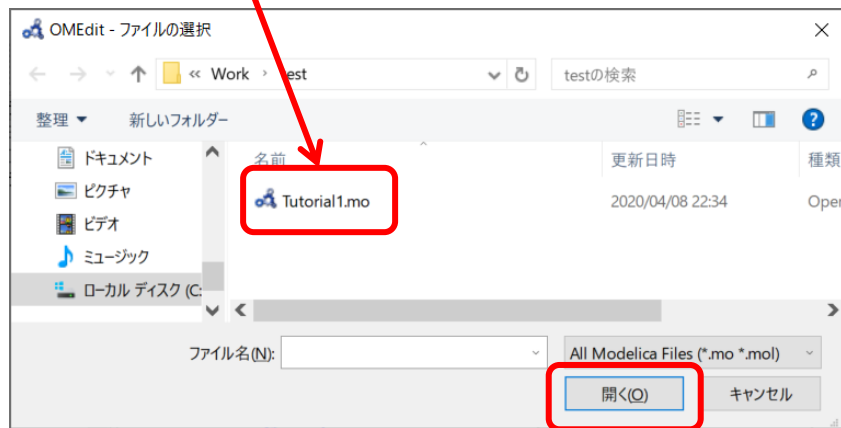
バネマスモデルの接続図

接続図に ρ と V のパラメータ変数を追加し
質量 m を入力するコマンドに $\rho \times V$ を追加します

ファイルを開く

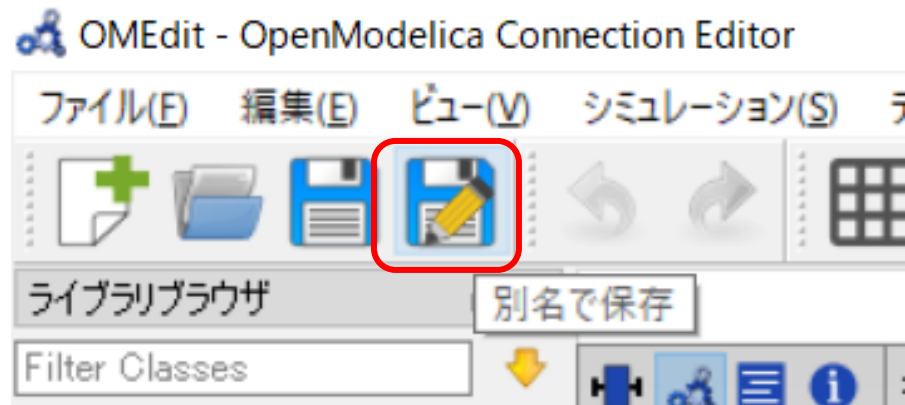
OpenModelica超初級チュートリアル1で作成したバネマスモデルを開きます。

- ① 「ファイル」 - 「モデル/ライブラリを開く」 から「Tutorial1.mo」を開く
- ② ライブラリブラウザに追加されたファイルをダブルクリック
- ③ モデルを確認してください

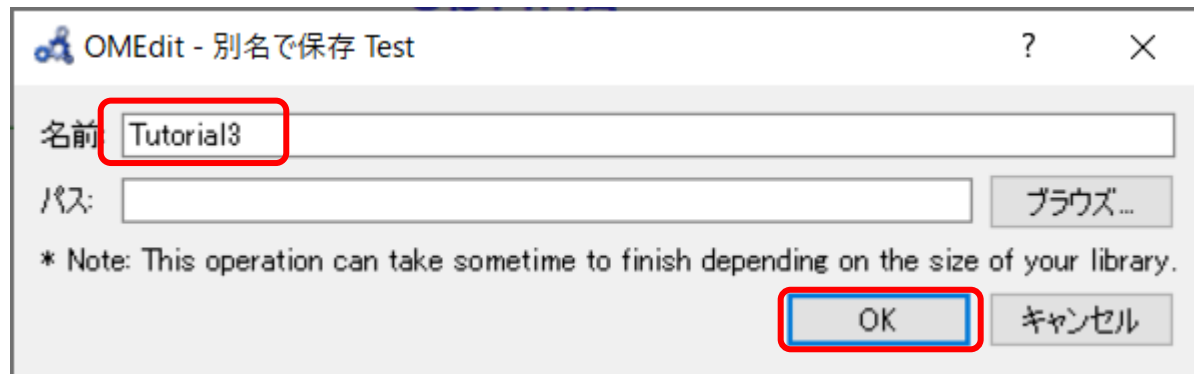


ファイルを別名保存

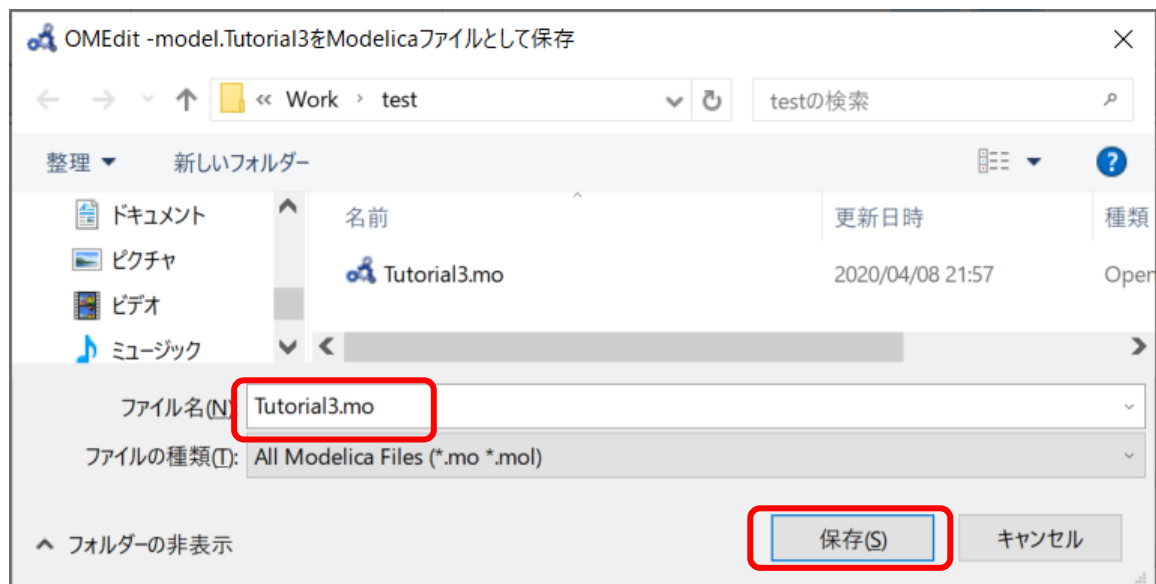
① 「別名で保存」をクリックします



② 任意の名前を入力し「OK」をクリックします

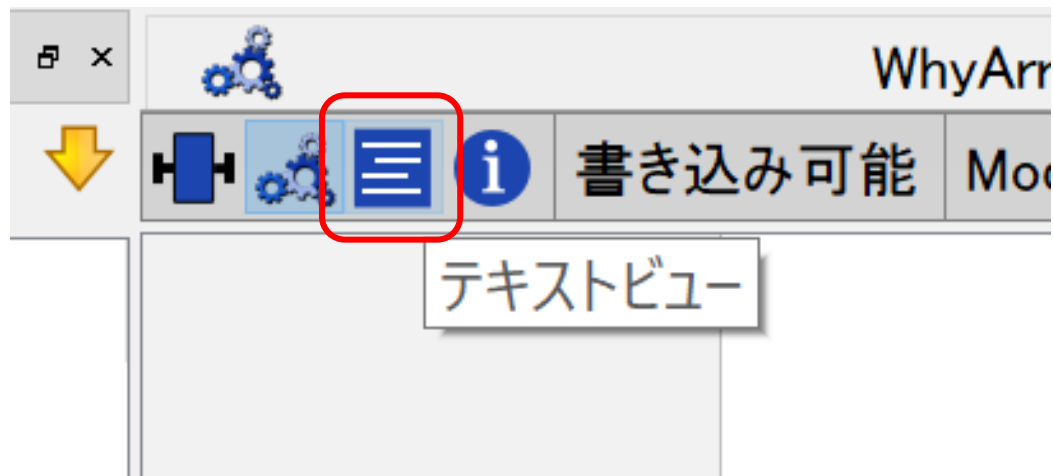


③ ファイル名を「保存」をクリックします



クラスのコード表示

ダイアグラムビュー左上にある
「テキストビュー」アイコンをクリックしてください



クラスに書かれたコードが表示されます

A screenshot of a code editor window titled 'Tutorial3*'. The editor shows the source code for a class named 'Tutorial3'. The code is as follows:

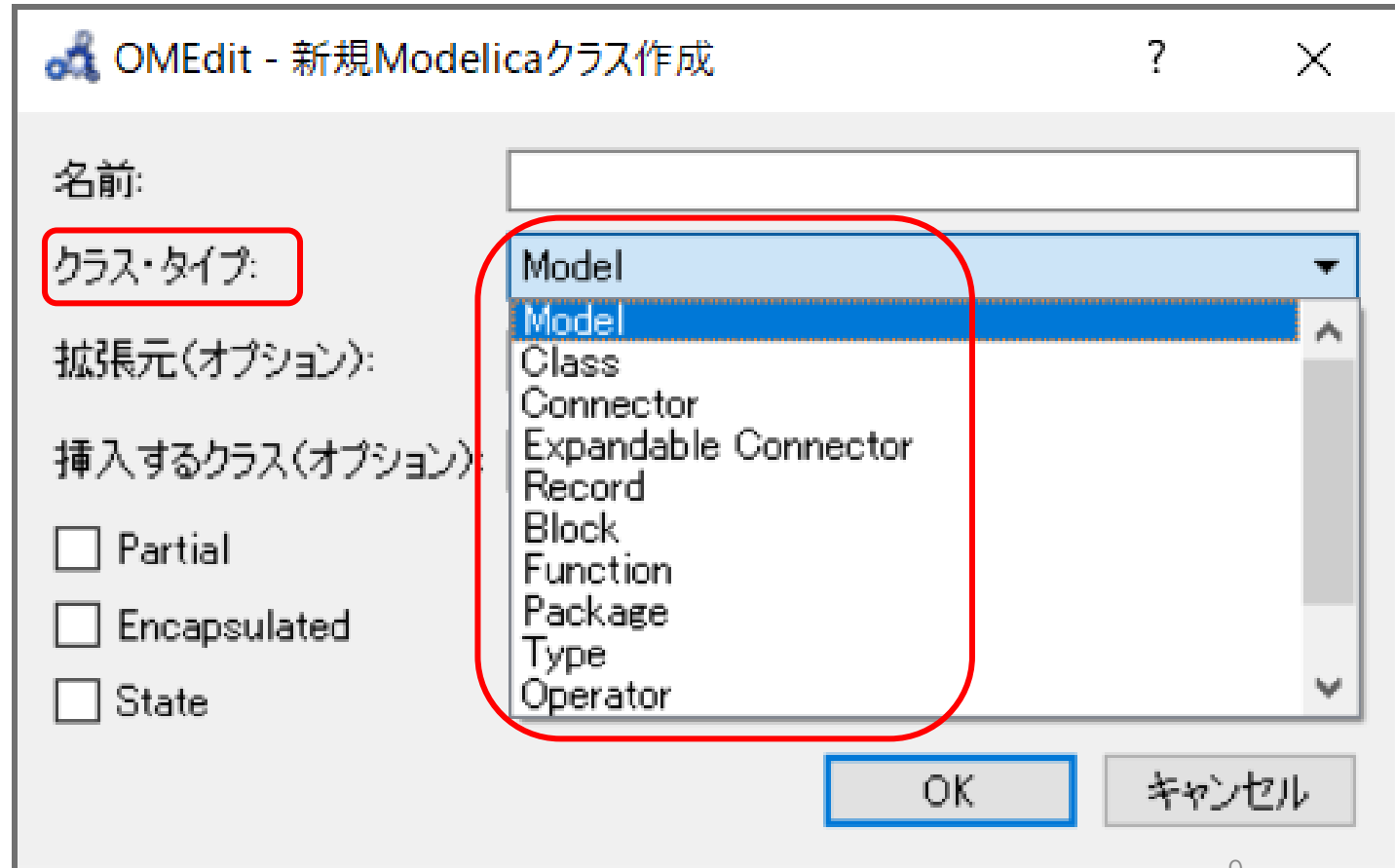
```
1 model Tutorial3
2   Modelica.Mechanics.Translational.Components.Fixed fixed annotation( ...);
4   Modelica.Mechanics.Translational.Components.Mass mass(m = 1, v(fixed = true, start = 0)) annotation( ...);
6   Modelica.Mechanics.Translational.Components.Spring spring(c = 1, s_rel(fixed = true, start = 0.1)) annotation( ..
8 equation
9   connect(fixed.flange, spring.flange_a) annotation( ...);
11  connect(spring.flange_b, mass.flange_a) annotation( ...);
13
14  annotation( ...);
16 end Tutorial3;
```


クラスとは？

Modelica内で作成するオブジェクト（プログラム）はすべて”クラス”と呼ばれます。
クラスは変数と方程式を規定したもので用途に応じて様々な種類があります。
主に使うのは以下の表のクラスで、まずは上位3つを覚えれば十分です。

クラス名	用途
Model	変数、方程式を有しシミュレーションが実行できる
Connector	クラス同士を接続する。 方程式は記述できない
Package	クラスを一つにまとめる
Block	変数、方程式を有しシミュレーションが実行できる。 入出力が決まっており因果的モデルが記述できる

クラスの選択は以下



復習 コードの概要

Modelica言語のコードは大きく以下の二つのセクションに分けて考えると学習しやすいです。

model HelloWorld

declarationセクション

parameter . . .
Real . . .

変数を宣言したり、継承関係を記述します。

この範囲を便宜的に「**declaration(宣言)セクション**」と呼びます。

equation

equationセクション

$\text{der}(x) = a * x;$

“equation”以下に方程式やモデル間の接続関係を記述します。

ここで記述された方程式群は計算実行時に自動的に未知数が選別され連立方程式が立てられ解が得られます。これを**非因果的(acausal)モデリング**と呼びます。

この範囲を便宜的に「**equation セクション**」と呼びます。

end HelloWorld;

コードの概要 – インスタンス

declarationセクション内には変数のほかにも使用するモデルも宣言できます。

宣言されたモデルをインスタンスと呼びます。

インスタンスはいくら変更しようとも、元のモデルには影響しないので安全に書き換えられます。

インスタンスの文法

インスタンスの元となるクラス名 インスタンス名（修飾） ダイアグラムビューの表示設定 ;

例. おもりを表すmassモデルのインスタンス

```
Modelica.Mechanics.Translational.Components.Mass mass(m = 1, v(fixed = true, start = 0)) annotation(
```

インスタンスの元となるクラス

インスタンス名
大抵、元のモデルの名前の頭文字を
小文字にします。

ダイアグラムビューの表示設定

クラス名の後ろの()内に"変数名=〇"と
することで値を与えることができます。
これを**modification(修飾)**と呼びます。
modificationには数式も使用できます。

コードの概要 – connect

モデル同士の接続関係をconnectを使って記述します。

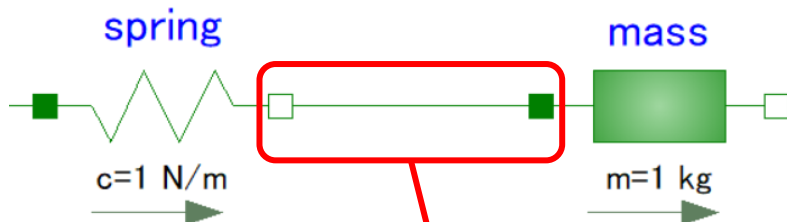
接続すると、コネクタ(connector)内で定義された変数をモデル間で参照(受け渡し)できます。

一般的に、Modelica言語ではモデル間の変数の受け渡しはコネクタを使用します。

connectの文法

connect (インスタンス名1.コネクタ名, インスタンス名2.コネクタ名) ダイアグラムビューの表示設定 ;

例. バネ(spring)と質量(mass)の接続



```
connect (spring.flange_b, mass.flange_a) annotation ( ... );
```

connect(○、×) で
○と×を接続しています。

インスタンス名.コネクタ名

ダイアグラムビューの表示設定

コードの概要 – connector

connectorとは、クラス同士の接続を行うためのクラスのことです。
connectorクラス内に宣言された変数をモデル同士で参照(受け渡し)できます。
(connectorは慣例的にPortと呼ばれることもあります)

connectorの構成

```
connect コネクター名  
  
    型やクラス名 変数名;  
    ⋮  
end      コネクター名;
```

例. connectorクラスFlangeのコード

```
connector Flange "One-c  
  
    SI.Position s "Absolute  
    flow SI.Force f "Cut  
    annotation ( ... );  
end Flange;
```

connector コネクター名

以下の変数の接続が可能なことを表しています。

位置座標	s
力	f

fには接頭辞にflowがついています。
これはチュートリアル7で説明します。

質量変数のカスタマイズ

質量変数 m を体積 V と密度 ρ の積で定義したい場合があります。
そのためにソースコードへ以下の追加を行います。

1st. パラメータ変数 V , ρ の追加

2nd. 計算式の追加

質量変数のカスタマイズー 1st. パラメータ変数 V, ρ の追加

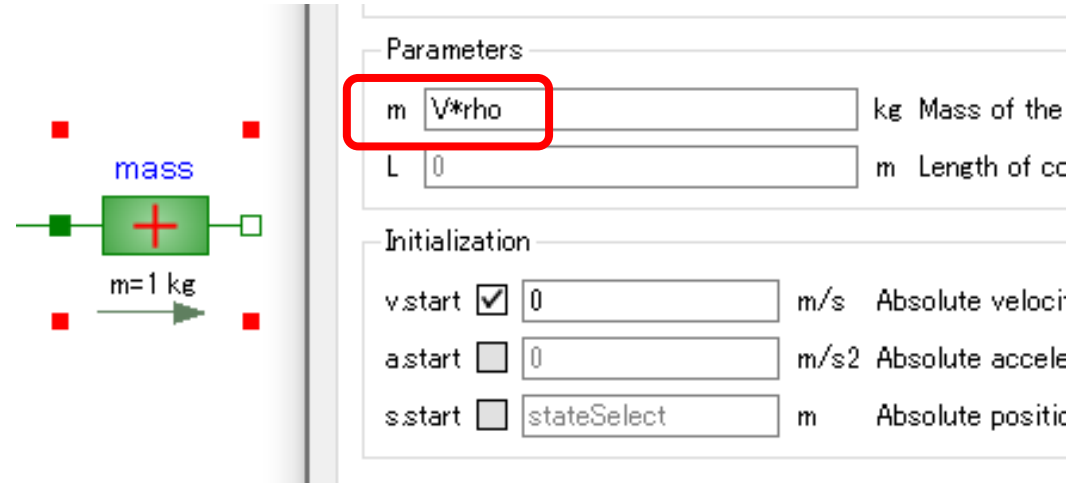
バネマスモデルの接続図(Tutorial3.mo)のテキストビューから
declarationセクションへ以下のパラメータ変数(体積 V , 密度 ρ)を追加してください。

```
model Tutorial3
  parameter Real V=3;
  parameter Real rho=5;
  Modelica.Mechanics.Tra
  Modelica.Mechanics.Tra
  Modelica.Mechanics.Tra
equation
  connect(fixed.flange,
  connect(spring.flange_
annotation( ... );
end Tutorial3;
```

追加

質量変数のカスタマイズー 2nd. 計算式の追加

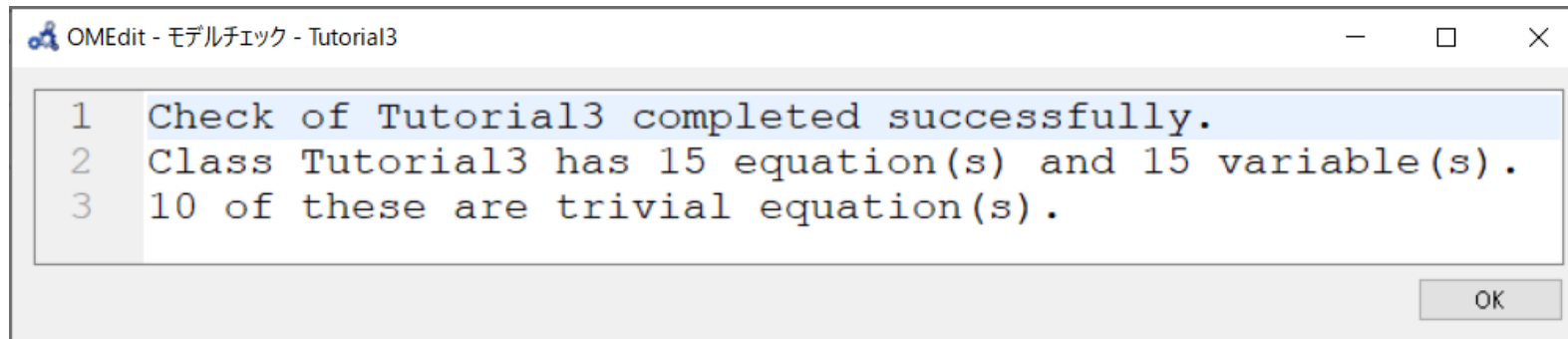
① massのパラメータmへ $V \cdot \rho$ を入力してください。



② テキストビューからmassのモディフィケーションが「 $m = V \cdot \rho$ 」となっていることを確認してください。

```
.Components.Mass mass (m = V * rho,
```

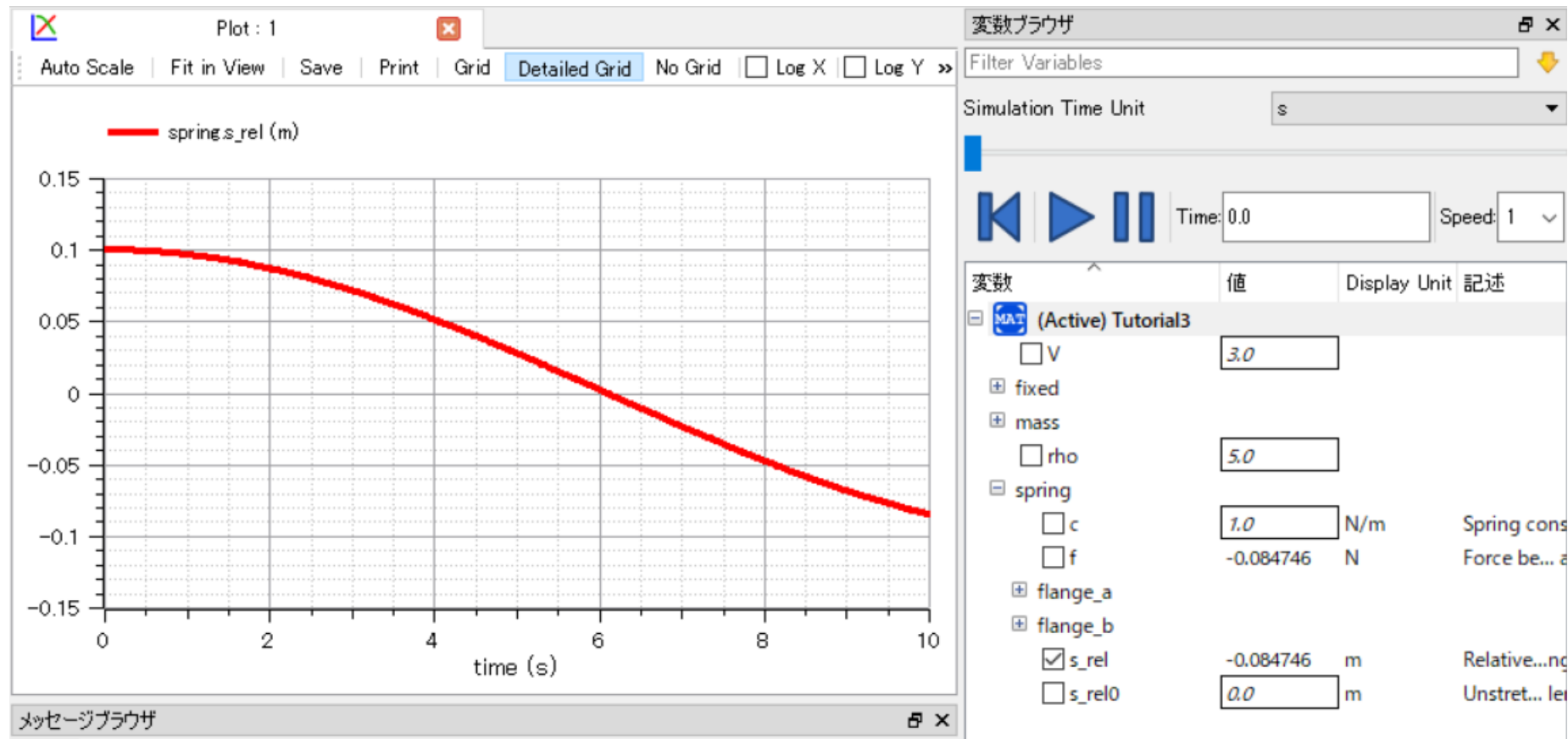
③ チェックして、変数と方程式が等しいことを確認してください。



質量変数のカスタマイズー 解析実行と結果

「シミュレーションのセットアップ」から「終了時刻」を10秒にして解析を実行してください。

解析結果より、チュートリアル1と比較して質量が大きくなっているためバネの変形がゆっくりになっている事を確認してください。



解析結果

Exercise

1. ρ 、 V を様々な値に変更し波形の変化を確認してください。
2. その他のパラメータ変数を関数にしてみてください。
例. 初期位置、バネ定数、体積 etc.
3. 質量 m を密度と体積の関数とする方法は沢山あります。
これまでのチュートリアルを超えていますが
興味ある人は、massモデルをコピーして改造したり
 m を別モデルで定義するなどして遊んでみてください。