

ソースコードの読み込みと生成 機能ガイド



---

## Generate and import source codes

*by SparxSystems Japan*

Enterprise Architect 日本語版

ソースコードの読み込みと生成 機能ガイド

(2016/10/07 最終更新)



## 1. はじめに

このドキュメントでは、ソースコードの読み込みと生成の操作方法について説明します。

なお、ソースコードの読み込みと生成は、Enterprise Architect デスクトップ版では利用できません。

## 2. ソースコードの読み込みと生成機能の概要

### 2.1. エディションの種類と利用できる機能

Enterprise Architect には、ソースコードと UML モデルに関連する機能が搭載されています。この機能は、Enterprise Architect のエディションによって以下のようになっています。

- Enterprise Architect プロフェッショナル版・コーポレート版

クラス図からソースコードを生成することと、既存のソースコードを読み込んでクラス図を生成することができます。

対象の言語は、Java, C#, C++, C, VB.Net, Delphi, PHP, Python, ActionScript, Visual Basic です。

- Enterprise Architect Suite ビジネスモデリング版

プロフェッショナル版の内容に加えて、Java, C++, C#, VisualBasic.NET, C 言語(オブジェクト指向モードのみ)の言語について、シーケンス図・アクティビティ図・ステートマシン図の3つの振る舞い図からのコード生成機能が利用できます。

また、「ビジネスルール」からのコード生成機能が利用できます。この機能については、PDF ドキュメント「ビジネスルールからのコード生成 機能ガイド」をご覧ください。

[http://www.sparxsystems.jp/products/EA/ea\\_documents.htm](http://www.sparxsystems.jp/products/EA/ea_documents.htm)

- Enterprise Architect Suite システムエンジニアリング版

プロフェッショナル版の内容に加えて、Ada, SystemC, Verilog, VHDL のソースコードを読み込んでクラス図を作成することと、クラス図から Ada, SystemC, Verilog, VHDL のソースコードを生成することができます。

また、SystemC, Verilog, VHDL, Java, C++, C#, VisualBasic.NET, C 言語(オブジェクト指向モードのみ)の言語について、シーケンス図・アクティビティ図・ステートマシン図の3つの振る舞い図からのコード生成機能が利用できます。

- ・ Enterprise Architect Suite アルティメット版

上記の全ての機能を利用することができます。

## 2.2. 機能の概要

### 2.2.1. ソースコードの読み込み

Enterprise Architect のソースコードの読み込み機能は、ソースコードからクラス図を生成することができます。

Enterprise Architect Suite シリーズで利用できる振る舞い図からのソースコード生成機能で生成したソースコードを読み込んで、シーケンス図やステートマシン図に反映することはできません。

(この振る舞い図からのソースコード生成機能は、UML モデルからソースコードへの一方のみとなります。)

ソースコードの読み込みでは、メソッド(操作)の内部(処理部分)は解析の対象になりません。なお、オプション設定を変更することで、メソッドの内部のコードをクラスの情報(メソッド)の情報の一つとして取り込むことは可能です。(設定は 6 章をご覧ください。)

読み込むソースコードの文字コードが SHIFT\_JIS 以外の場合には、事前に第 6 章の内容をご覧ください。設定を変更してください。

### 2.2.2. ソースコードの生成

Enterprise Architect のソースコード生成は、基本的にはクラス図から、ソースコードの「スケルトン」の出力となります(C 言語や C++言語における「ヘッダファイル」の情報の

みを出力します)。メソッドの内部については空になります。クラス図で定義されている、関連や汎化などの関係がソースコードに反映されます。

(各操作のプロパティ画面の「振る舞い」グループにある「既定のソースコード」の欄に入力された内容は、ソースコードの新規生成時に、操作(メソッド)の実装として出力されます。すでにソースコードがある場合の同期処理の場合には、この内容は反映されません。)

なお、ステートマシン図からのソースコード生成については、ドキュメント「ステートマシン図からのコード生成 機能ガイド」を、シーケンス図やアクティビティ図からのソースコード生成については、ドキュメント「アクティビティ図・シーケンス図からのコード生成 機能ガイド」をご覧ください。

[http://www.sparxsystems.jp/products/EA/ea\\_documents.htm](http://www.sparxsystems.jp/products/EA/ea_documents.htm)

生成するソースコードや読み込むソースコードの文字コードを `SHIFT_JIS` 以外の変更したい場合には、事前に第 6 章の内容をご覧ください、設定を変更してください。

### 3. クラス図からソースコードを生成する手順

まず、作成したクラス図からソースコードを生成する方法について説明します。最初に、クラスに対する設定を行う必要があります。今回の例では、以下のようなクラスを対象に説明します(図 1)。

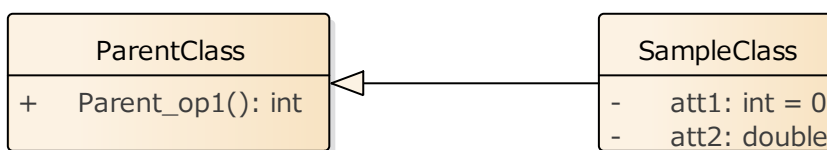


図 1

クラスの設定を行うには、対象のクラスのプロパティを開きます。「言語」のドロップダウンリストから希望する言語を選択します(図 2)。

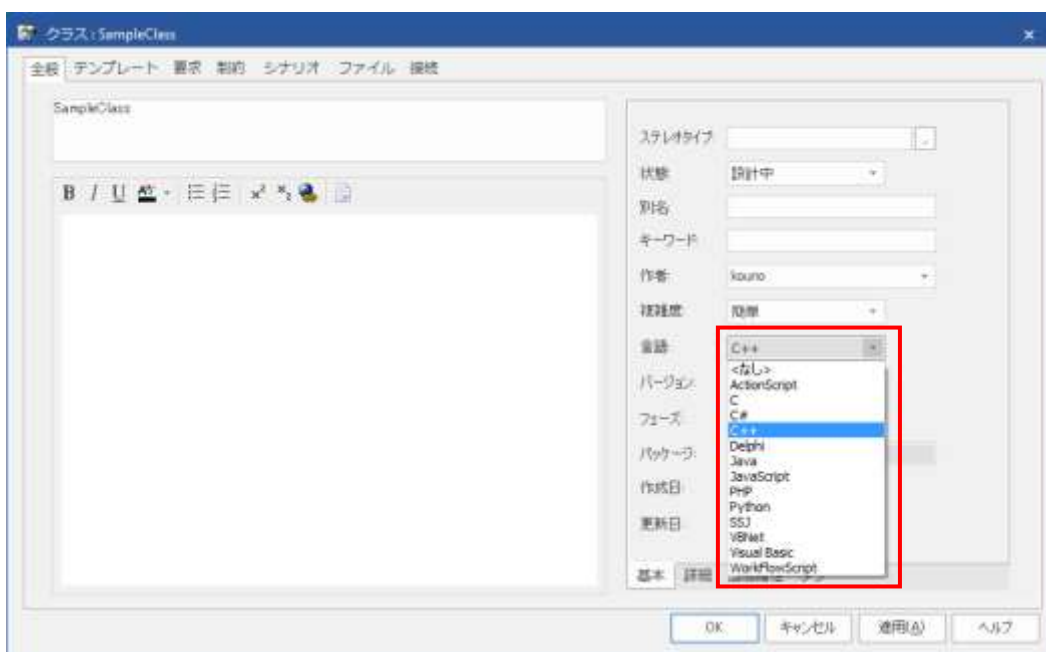


図 2

以上で、ここで行う設定は完了です。

### 3.1. 単一のクラスに対してコードを生成する

次に、このクラスのソースコードを出力します。対象のクラスを右クリックして、「ソースコードの生成と読み込み」→「ソースコードの生成」を選択します。

すると、「ソースコードの生成」画面が表示されます(図 4)。「対象の言語」に先ほど設定した言語が表示されていることを確認してください。

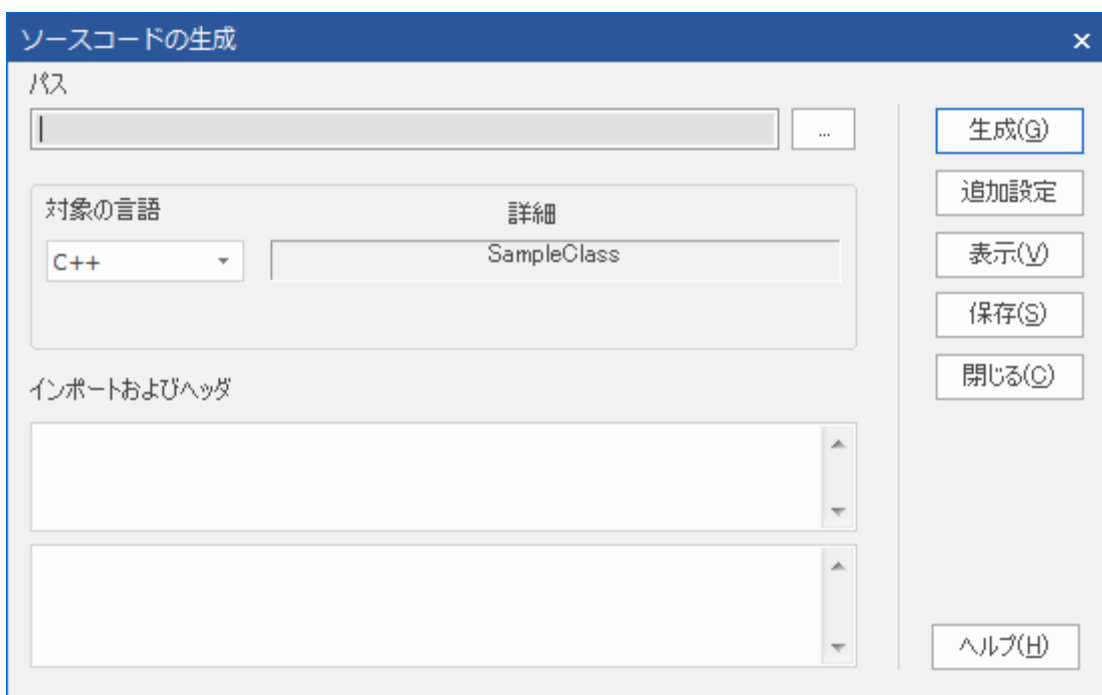


図 4

この画面の上部の「パス」は、実際に出力されるファイル名です。右端の「...」ボタンを押して出力する位置とファイル名を指定します。また、下にある「インポートおよびヘッダ」は、無条件にファイルの先頭に付加する文字列を指定します。例えば、**Java** の場合には **import** 文、**C++** の場合には共通ライブラリのヘッダの **include** などを記入します。なお、**Java** のように生成するファイルが1つのみの場合には、上段のエディットボックスを利用してください。**C++** のように、ヘッダファイル(.h)とプログラムソースファイル(.cpp)が分かれている場合には、上段がヘッダファイルで下段がソースファイルに対応します。

基本的には、この画面ではパスのみを設定します。なお、パス情報は保存されますので、以前に生成したことがある場合には、その情報が設定されています。

設定が完了したら、「生成」ボタンを押します。

ソースコードを確認する簡単な方法として、**Enterprise Architect** の内部で表示する方法があります。対象の要素を右クリックして「ソースコードの生成と読み込み」→「ソースコードの表示」を選択して下さい。タブとしてソースコードが表示されます。

このタブはエディタになっていますので、簡易な編集も可能です。

### 3.2. まとめてソースコードを生成する

以上の方法で、ひとつのクラスに対してソースコードを生成することができますが、クラスの数が多い場合には、この方法でソースコードを生成するのは手間がかかります。そこで、パッケージ単位でソースコードを生成する方法について説明します。

なお、この方法を利用する場合でも、それぞれのクラスごとに言語を設定する必要があります。クラスが生成される際の既定の言語を設定することができますので、利用する言語が決まっている場合には最初に設定してください。

「ホーム」リボン内の「オプション」パネルにある「プロジェクト」を選択し、プロジェクトのオプション画面を表示させます。左側のツリーから「ソースコードの生成と読み込み」カテゴリを選択すると、右側に「コード生成の既定言語」という選択肢があります。このドロップダウンリストから、希望する言語を設定した後にクラスを作成すれば、クラスの言語が自動的に設定されます(図 5)。なお、既存のクラス要素の設定には影響しません。既に作成済みのクラスの言語の設定を一括変更する方法は、ヘルプの「パッケージ内のクラスの設定を一括変更」のページをご覧ください。

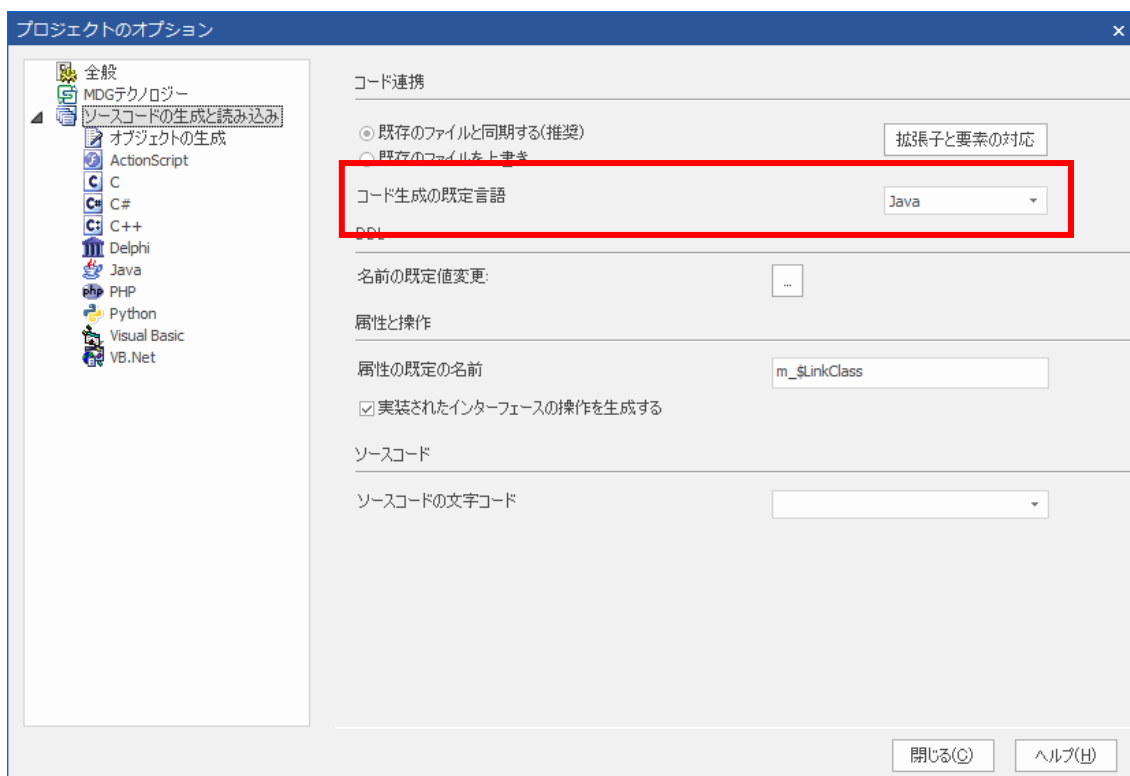


図 5

それでは、パッケージを対象に一括コード生成を行う方法を説明します。プロジェクトブラウザから対象となるパッケージを選択し、右クリックしてコンテキストメニューを表示させます。このメニューから「ソースコードの生成と読み込み」→「ソースコードの生成」を選択してください。すると、「ソースコードの生成(パッケージ)」画面が表示されます(図 6)。

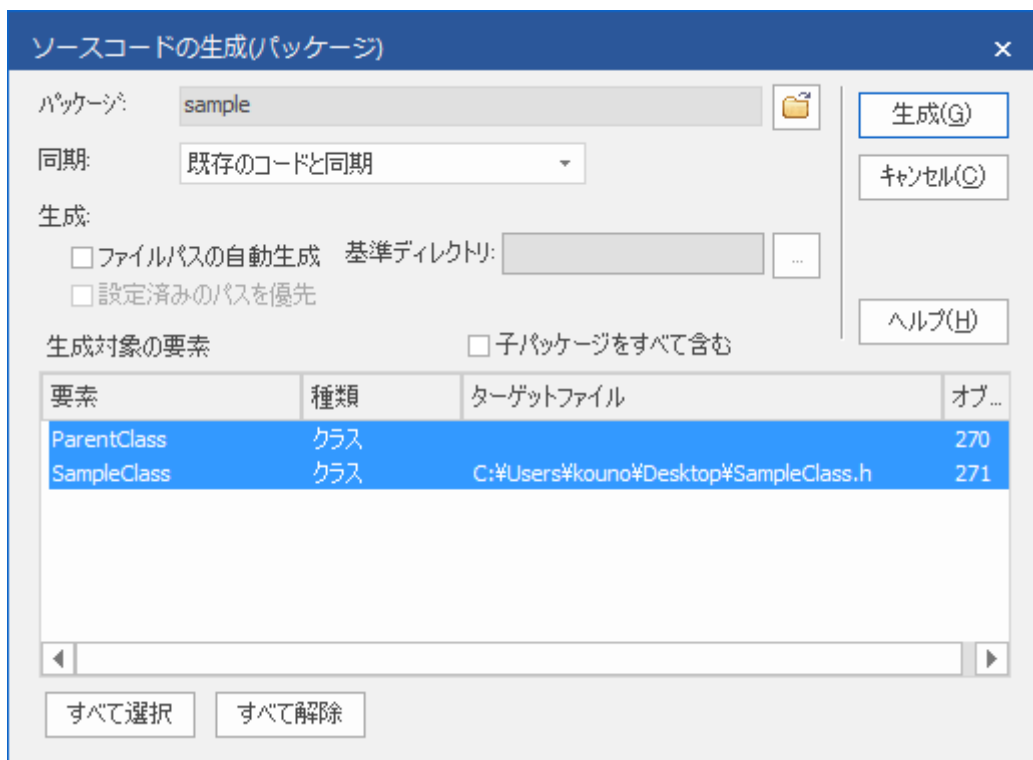


図 6

この画面で、設定する必要がある主な項目について説明します。まず、「生成対象の要素」の一覧は、対象となるクラスやインターフェースの名前・種類・出力ファイルのパスが表示されています。この一覧は既定値ですべての項目が選択状態になっています。この選択を解除すると、生成の対象外となります。必要に応じて、この一覧から出力したいクラスを選択してください。このときに、「すべて選択」「すべて解除」のボタンが便利です。また、指定したパッケージの子パッケージに含まれるクラスをまとめて出力したい場合には「子パッケージをすべて含む」にチェックを入れてください。自動的にクラスやインターフェースを検索して、一覧を更新します。

なお、上記図 6 の例ですと、ParentClass はターゲットファイルが指定されていません。このような場合に生成を実行すると、実行時に出力する位置とファイル名を入力する必要があります。



一括で出力する場合に、ファイルパスをまとめて指定したい場合もあると思います。このような場合には、「ファイルパスの自動生成」にチェックを入れてください。ディレクトリの位置を指定する画面が表示されますので、希望する位置を指定してください。すると、現在のパッケージの階層構造と同じディレクトリ構成のパスをすべてのクラスに対して自動的に生成します。次の例は、生成パスを自動生成した例です(図 7)。

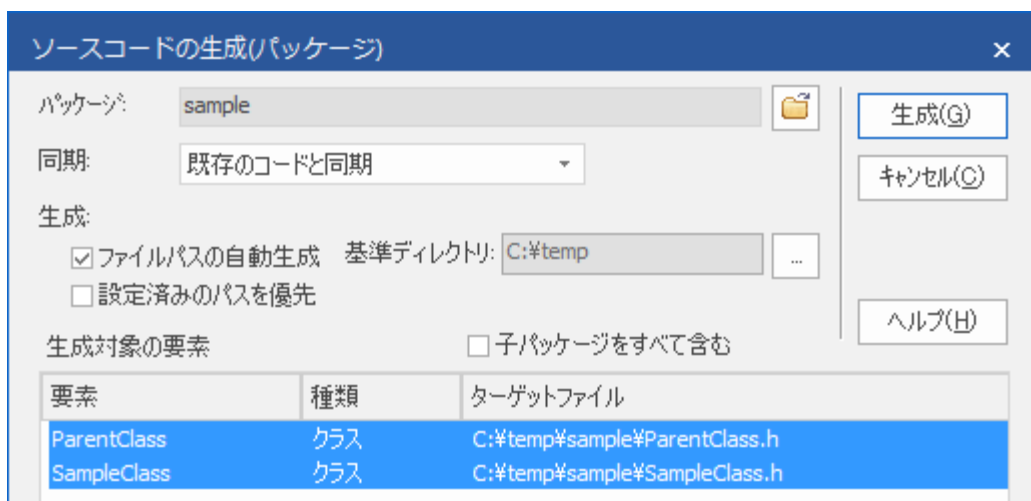


図 7

なお、既にパスが指定してあるクラスに対してはそのパスを利用し、それ以外のクラスについては自動的にパスを生成したいという場合には、「設定済みのパスを優先」にチェックを入れてください。

以上の設定を行った後で、「生成」ボタンを押せば、ソースコードが生成されます。

## 4. 言語ごとの個別設定

Enterprise Architect は 10 種類(Enterprise Architect Suite システムエンジニアリング版・アルティメット版は 14 種類)のプログラム言語の入出力に対応しています。それぞれの言語には、言語固有の表現やキーワードがあります。Enterprise Architect では、こうした言語固有の情報はタグ付き値やステレオタイプで管理しています。

例えば、C++言語の場合に、クラス要素にステレオタイプ<<struct>>を設定してコード生成を行うと、構造体として出力されます。

このように、言語ごとの個別の関連付けについては、ヘルプの「ソースコードとモデルの対応付け」のページをご覧ください。

## 5. 既存のソースコードを読み込む方法

次に、既存のソースコードを読み込む手順について紹介します。

この章では具体的な例として、オープンソースソフトウェアである **Tomcat** のソースコードの一部を例にして説明します。

### 5.1. ソースコードの読み込み

ソースコードの読み込みの実行前に、「7. ソースコードの読み込みのオプション」の内容をご覧ください。必要に応じて設定を変更して下さい。

最初に、ソースコードを読み込んでクラス要素を生成する対象のパッケージを指定します。作成する位置のパッケージをプロジェクトブラウザで選択し、右クリックしてください。コンテキストメニューが表示されますので、「ソースコードの生成と読み込み」→「ソースディレクトリの読み込み」を選択します。すると、図 8 のような画面が表示されます。

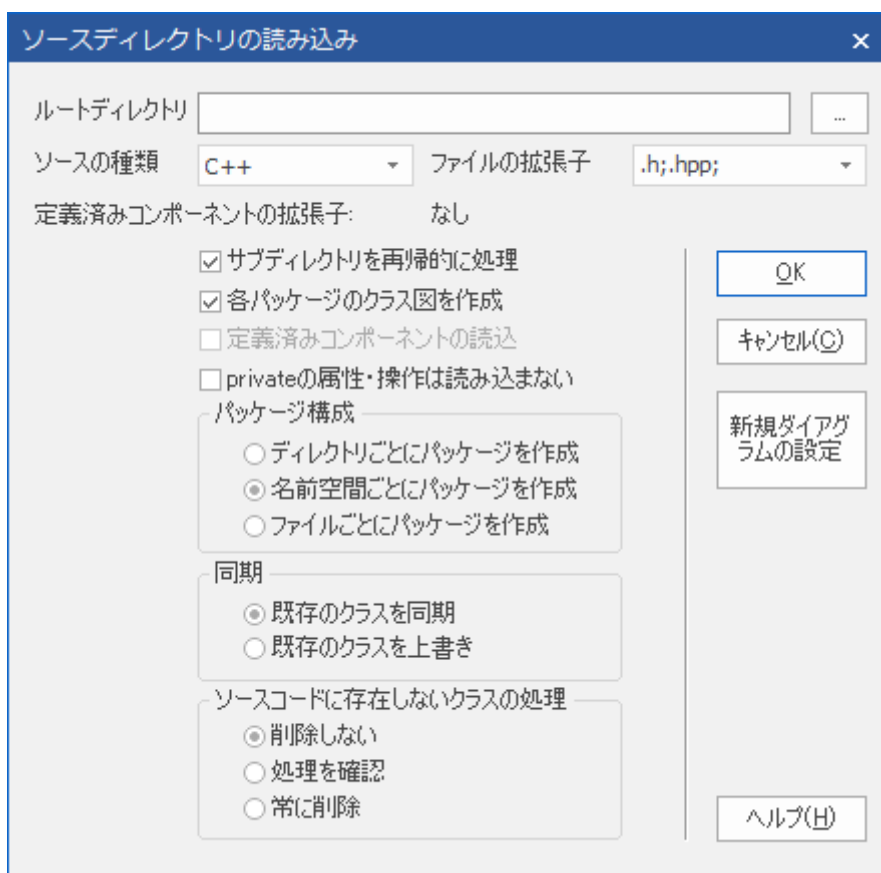


図 8

## 5.2. 読み込み位置と方法の選択

ルートディレクトリには、読み込む対象のファイルが含まれるディレクトリを選択します。また、ソースの種類を設定します。今回の場合には、**Java** に設定します。ソースの種類を指定すると、拡張子の内容は自動的に変わります。この設定が適切でない場合には、ソースコードの読み込みを実行しても対象のソースファイルを発見できず、そのまま終了します。何も読み込まれない場合には、この部分を確認してください。

ソースコードの読み込み時にクラス図を作成する場合には、「各パッケージのクラス図を作成」にチェックを入れて下さい。各パッケージ(名前空間あるいはディレクトリ)ごとに、園パッケージに含まれるクラス要素が配置されたクラス図も生成されます。

その他の設定項目については、必要に応じて変更して下さい。「OK」ボタンを押せば、読み込みが開始されます。画面にクラスが追加され、しばらくすると読み込まれた結果が表示されます。

### 5.3. 注意点

ソースコードの読み込みを行う場合に注意点があります。

既存のソースコードの解析を行う場合に、対象のソースコードが多い場合にまとめて読み込むと、解析に非常に時間がかかります。例えば、クラスが 1000 個ある場合には、クラス間の関係は  $1,000^2=1,000,000$  個になります。これだけの関係を同時に解析することになるため、多くのメモリ量および高いマシンの性能が必要になります。

ソースコードの量が多い場合には、分割して読み込むことをお勧めします。例えば、対象のルートディレクトリのなかにサブディレクトリが複数個ある場合には、それぞれのサブディレクトリ単位で読み込み機能を実行します。これにより、同時に解析する必要のあるクラス間の個数が減るため、結果的には短時間で作業が完了します。

また、1つのディレクトリ(言語によっては名前空間)に多くのクラスが含まれる場合、自動生成されるクラス図には多くのクラスが含まれ、表示する場合に時間がかかります。この場合には、属性や操作を非表示にすると改善できます。

その手順は次の通りです。

1. ダイアグラムを開きます。
2. ダイアグラムの背景でダブルクリックし、プロパティ画面を表示します。
3. 「要素」タブにある「区画の表示」グループ内の「属性」「操作」のチェックを外します。あるいは、「属性・操作」タブにある、「表示対象の指定」のうち、不要な属性や操作の可視性のチェックを外します。
4. ダイアグラムに戻ると、不要な情報が表示されなくなり、描画速度を改善できます。
5. 要素に余白が表示されているような場合で、要素の大きさを調整する場合には、**Ctrl+A** でクラス図内の要素を全て選択した状態で、**ALT+Z**(「ダイアグラム」リボン内の「ダイアグラム」パネルにある「要素」ボタンを押すと表示されるメニューから「選択した要素の自動サイズ調整」)を押すと、最適なサイズに縮小されます。
6. さらに、再度自動レイアウトを実行する場合には、レイアウトツールサブウィンドウを利用してください。「基本」レイアウトが便利です。

### 5.4. エラーが発生する場合

読み込みの際に、エラーが発生する場合があります。このエラーのうち、C++言語の場合には出力されるログに「プログラム言語のマクロを定義する必要があるかもしれません。」と表示される場合があります。このような場合には、多くの場合には独自のマクロが解釈できずにエラーとなっています。

(Enterprise Architect では、マクロの解釈・展開などのプリプロセス処理は行いません。)

## 5.5. マクロとして認識する文字列の追加

このようなマクロがある場合には、「プロジェクト」リボン内の「ツール」パネルにある「設定」ボタンを押すと表示されるメニューから「プログラム言語のマクロ」を実行してください。図 10 の画面が表示されます。ここで追加した値は、マクロと認識して無視することができます。

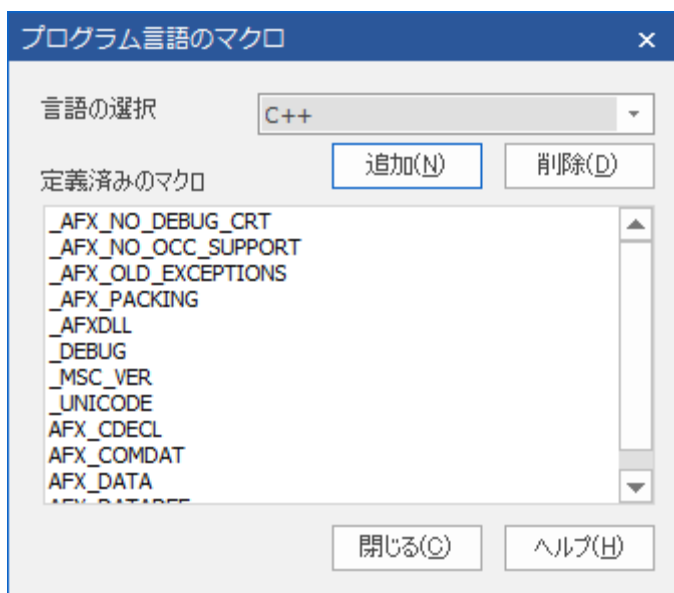


図 10

「追加」ボタンを押し、表示される画面でエラーになっているマクロ名を入力してください。

## 5.6. 独自のクラス図の作成

ソースコードの読み込み機能で自動的に作成可能なクラス図は、同じパッケージに含まれるすべてのクラス要素を含んだものになります。パッケージ内のクラス要素が多い場合には、非常に大きなクラス図が生成されます。

特定のクラスに着目したクラス図を作成するには、次の方法で行います。

1. 新規にクラス図を作成し、注目したいクラス要素を1つ、クラス図の中に配置します。  
(プロジェクトブラウザからドラッグします)
2. その要素を右クリックし、「関係する要素の追加」を実行します。「関係する要素の追加」画面が表示されますので、最初は何も設定を変更せず、「全て」ボタンを押して条件を満たす要素を全て選択した状態でOKボタンを押してください。関連や依存などの関係がある要素が、自動的に配置されます。
3. この作業を他のクラスに対して繰り返したり、設定内容を変更して必要な要素のみを追加したりするなどの方法で、注目したい情報のみが配置されたクラス図を作成できます。

なお、特定のパッケージについて、子パッケージも含めて全てのパッケージ内に含まれるクラスのクラス図を作成したい場合には、以下のページからダウンロードできるアドイン「クラス図自動作成アドイン」が便利です。

(製品のサポートユーザーであれば無料でダウンロード・利用できます。)

<http://www.sparxsystems.jp/products/EA/tech/Addins.htm>

## 5.7. クラス図作成の基本的なルール

ソースコードを読み込んでクラス図を作成する場合の、基本的なルールは次の通りです。

- ・ 原則として、1つのクラスに対して1つのクラス要素を生成します。
  - C#の **Partial** クラスの場合には、複数のクラス要素を生成します。
  - C言語の場合には、1つのファイル(.h)に対して1つのクラス要素を生成します。
- ・ 継承関係は、「汎化」の関係を作成します。
- ・ 属性として他のクラスを保持している場合には、「関連」の関係を作成します。
  - 「集約」は作成しません。
- ・ 第7章で説明するオプション「戻り値や引数にクラスが利用されている場合には、読み込み時に依存関係を作成」が有効な場合には、属性の型・操作の引数の型・操作の戻り値の型に他のクラスを利用している場合には、「依存」の関係を作成します。
- ・ コレクションクラス(例えば `list<SomeClass>` のような形式での利用)に対して適切に「関連」を作成するためには、ユーザーオプションの「追加のコレクションクラス」の設定が必要になる場合があります。この設定についてはヘルプファイルをご覧ください。

## 6. ラウンドトリップでの利用

### 6.1. 概要

Enterprise Architect では、クラス要素からソースコード生成を実行したり、ソースコードを読み込んでクラス図を作成したりした後に、以下のような操作が可能です。

- ・ ソースコードを編集し、クラス要素に同期反映する
- ・ クラス要素の内容を変更し、ソースコードに同期反映する

これらの操作方法は、新規にソースコードを生成したり、クラス図を生成したりする手順と変わりません。例えば、ディレクトリを指定してソースコードを読み込む場合、既に読み込み済みのクラス要素がモデル内にある場合には、そのクラス要素との同期処理が行われます。モデル内に対応するクラス要素が存在しない場合には、新規にクラス要素が生成されます。

このラウンドトリップにおいては、以下の注意事項があります。

- ・ 同期するかどうかの判定は、それぞれのクラス要素内で保持しているソースファイルのパスが一致するかどうかが基準です。同じソースコードであっても、配置されているディレクトリが異なる場合には、新規のクラス要素として読み込まれます。複数人で作業をしていて、クラス要素に対応するソースコードの位置が変わる場合には、「ローカルパス」の機能をご利用ください。
- ・ Enterprise Architect Suite の各エディションで利用できる、振る舞い図からのソースコード生成は、読み込みやラウンドトリップには対応しません。
- ・ 同期の対象は、クラスが持つ属性や操作(メソッド)の定義のみです。操作(メソッド)の中身(実装)は対象外です。また、クラス名が変わる場合やパッケージ名(ディレクトリ・名前空間)が変わる場合も、同期の対象外です。  
(Enterprise Architect では、ソースファイルに、ツールのための専用のコメントを埋め込みません。そのため、位置や名前が変わる場合には、同じかどうかの判別ができません。)
- ・ クラス要素の操作を編集・削除した場合や、ソースコードのメソッドを編集・削除した場合に同期する場合の結果は、下記の「モデルから削除した場合の動作について」をご覧ください。

- ・ クラス要素の属性・操作の名前を変更してソースコード生成を実行する場合には、オプション画面の「属性と操作」グループにあるオプション「コード生成時にモデルにない属性と操作の削除を確認する」が有効になっていない場合には、変更後の名前の属性や操作がソースコードに追加されます。名前を変更して置換したい場合には、このオプションを有効にしてください。  
(下記の「モデルから削除した場合の動作について」をご覧ください。)

ラウンドトリップで有用な情報は、無料セミナー「Enterprise Architect ソースコード生成ルール カスタマイズ 実践セミナー」でも説明しています。このセミナーは下記 Web サイトから動画をいつでもご覧いただけます。「1. ソースコード生成の概要」の内容をぜひご覧ください。

Enterprise Architect ソースコード生成ルール カスタマイズ 実践セミナー

<http://www.sparxsystems.jp/seminar/EACodeGenerate.htm>

## 6.2. モデルから削除した場合の動作について

Enterprise Architect では、既定の設定では、以下のような動作になります。

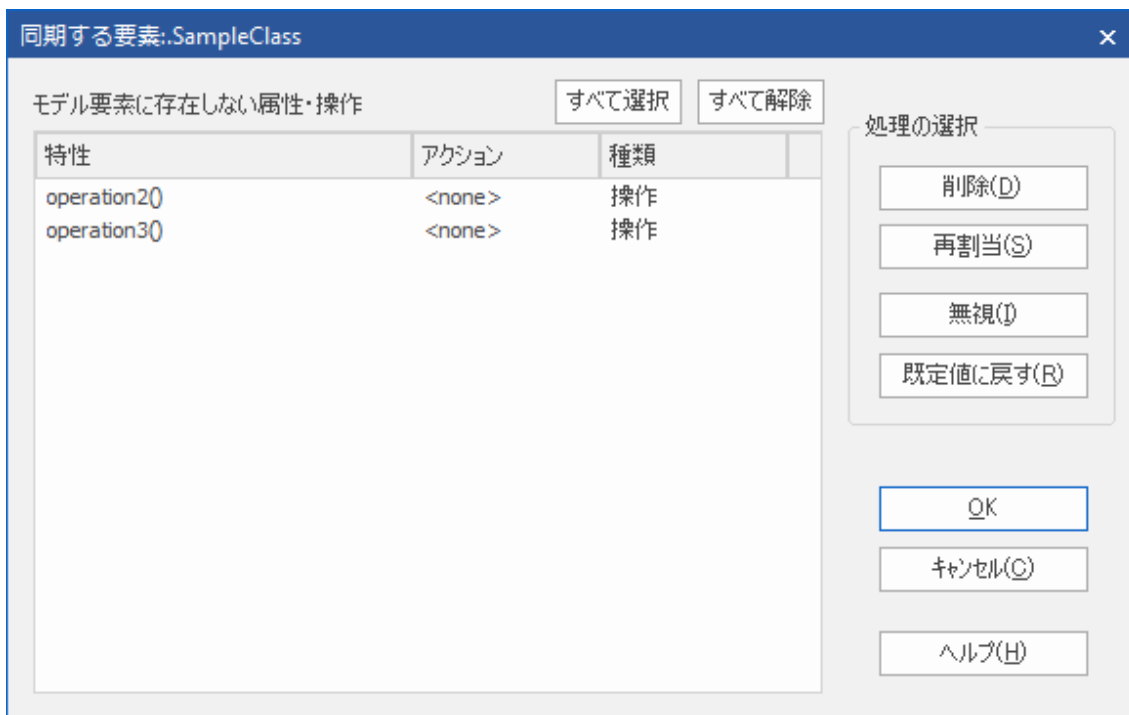
- ・ ソースコードで属性やメソッドの追加・編集・削除を行った後にモデルに反映する場合には、すべての変更内容が反映されます。
- ・ モデルで属性やメソッドの追加を行った後にモデルに反映する場合には、すべての変更内容がソースコードに反映されますが、編集した場合には、正しく反映できない場合があります。また、削除した場合には、変更内容は反映しません。

このような動作になっている理由は、次の 2 点です。

- ・ モデルからのソースコード生成では、操作(メソッド)の実装は出力されず、ソースコード側に追記します。つまり、情報量としてはモデルよりもソースコード側が多いということになります。この情報量の違いがあるため、モデルから属性や操作を削除した場合に、情報量が多いソースコード側を自動的に削除することは危険です。  
(ソースコード側のみに存在する情報が存在するため)
- ・ Enterprise Architect では、ソースコード内にツールのための情報(タグ)を埋め込みません。そのため、属性や操作の編集や削除を正確に判断することができません。上記のように、情報量の違いがあるため、不正確な判断で編集・削除することが問題となる場合があります。



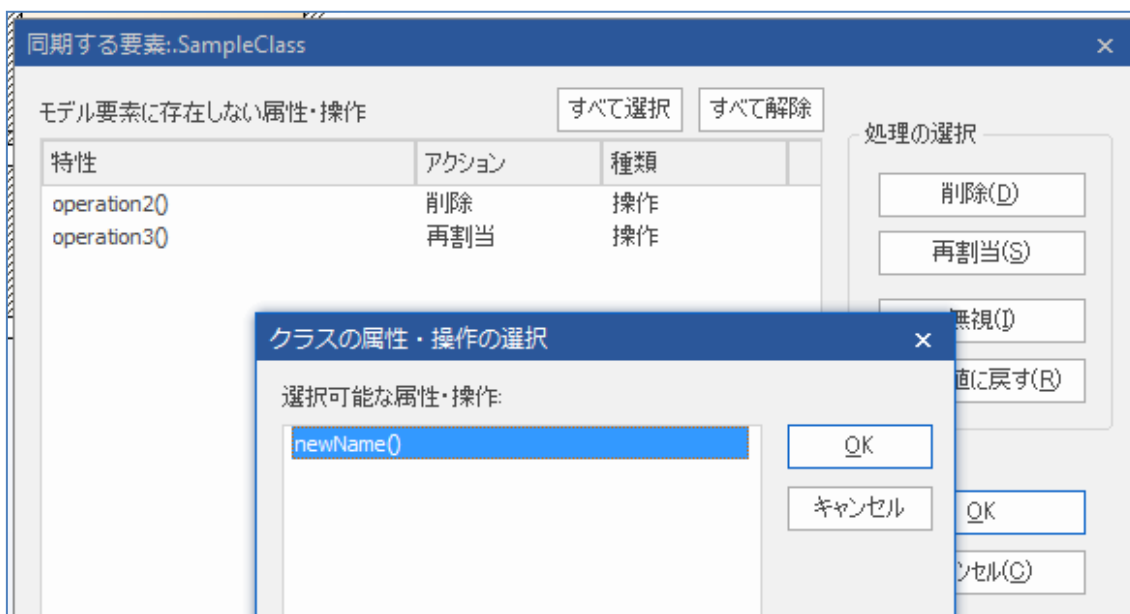
そのため、属性や操作の編集や削除を実行するためには、「コード生成時にモデルにない属性と操作の削除を確認する」のオプションを有効にする必要があります。このオプションが有効になっている場合に、モデルからソースコードを同期生成すると、以下のような画面が表示される場合があります。



この例では、モデルから **operation2** を削除し、**operation3** の名前を **newName** に変更しています。これらの変更が確実に判断できなかったため、どのようにすればよいのかをツールの利用者に確認しています。このように、不確実な点がある場合には、利用者が判断するようになっています。

(オプション「コード生成時にモデルにない属性と操作の削除を確認する」が有効になっていない場合にはこの画面は表示されず、属性や操作が削除されることはありません。)

今回の例では、以下のように、**operation2** については「削除」を、**operation3** については **newName** への「再割当」を指定する必要があります。



こうした操作により、同期生成時に適切にソースコードに反映することができます。

## 7. ソースコードの読み込みのオプション

ソースコードを読み込む際に、Enterprise Architect のオプションを変更することで、結果が変わる部分があります。このようなオプション項目のうち、知っておくと便利な項目をご紹介します。

オプション画面のうち、ユーザーごとに設定が保存されるオプションを表示するには、「ホーム」リボン内の「オプション」パネルにある「ユーザー」ボタンを押します。その後、左側の一覧で「ソースコードの生成と読み込み」を選択して下さい。以下のような画面になります。

## ソースコードの読み込みと生成 機能ガイド

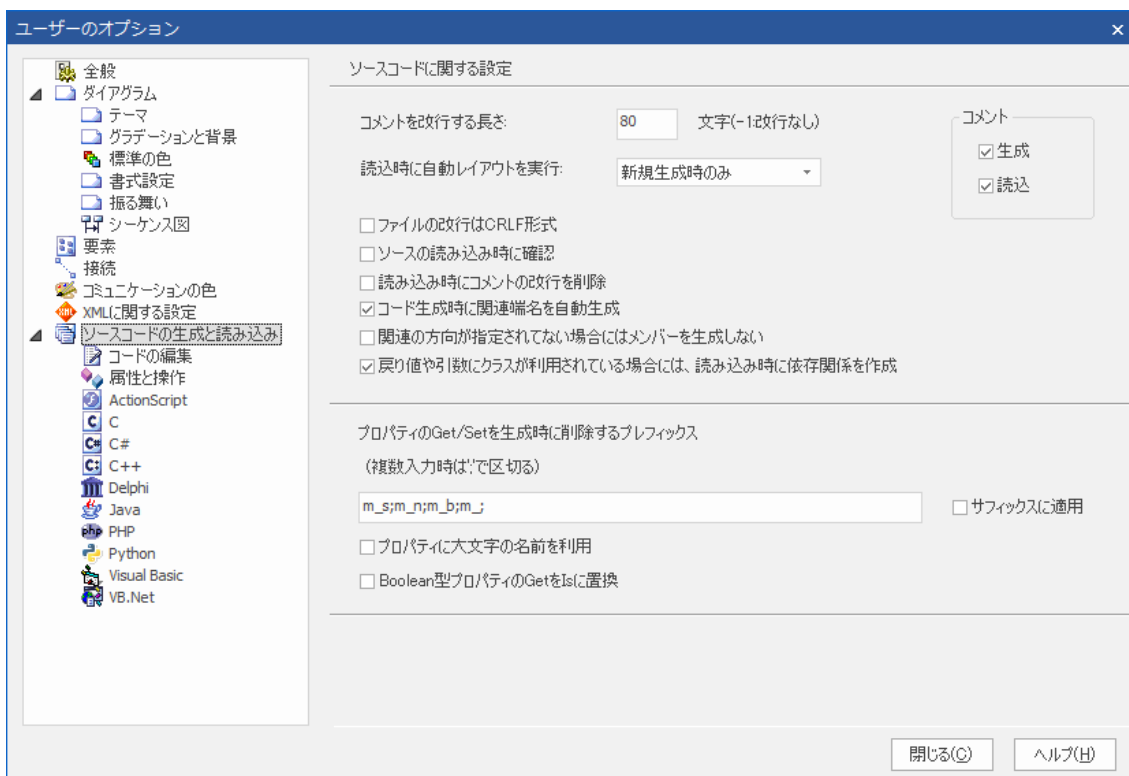


図 12

ここで、よく利用する設定は次の項目です。

- ・ 戻り値や引数にクラスが利用されている場合には、読み込み時に依存関係を作成

このチェックボックスを ON にすると、ソースコードの読み込み時に、「関連」の関係だけでなく、「依存」の関係も追加されます。そのため、既存のソースコードの解析に役立ちます。

また、「属性と操作」のページでは、以下の設定が役に立ちます。



- 読み込み時に操作の実行もモデルに読み込む

この項目にチェックを入れると、メソッドの実装(内部のコード)を、**Enterprise Architect** のプロジェクトファイル内に取り込みます。この情報は、操作のプロパティ画面にある「振る舞い」グループの「既定のソースコード」欄に格納されます。なお、もし「既定のソースコード」欄に情報がある場合には、ソースコードの生成時にこの内容をソースコードに出力します。(新規生成時のみ。同期の場合は除きます。)

- コード生成時にモデルにない属性と操作の削除を確認する

この項目については、**6.2** 章をご覧ください。

プロジェクトごとに設定が保存されるオプションを表示するには、「ホーム」リボン内の「オプション」パネルにある「プロジェクト」ボタンを押します。その後、左側の一覧で「ソースコードの生成と読み込み」を選択して下さい。

このページでよく利用する設定は次の項目です。

- ソースコードの文字コード

ソースコードが SHIFT\_JIS あるいは US\_ASCII 以外の場合には、変更が必要です。なお、日本語 EUC の場合には、「20932 (JIS X 0208-1990 & 0212-1990)」を指定してください。

なお、この設定は、CSV ファイルの入出力時の文字コード設定としても利用されます。

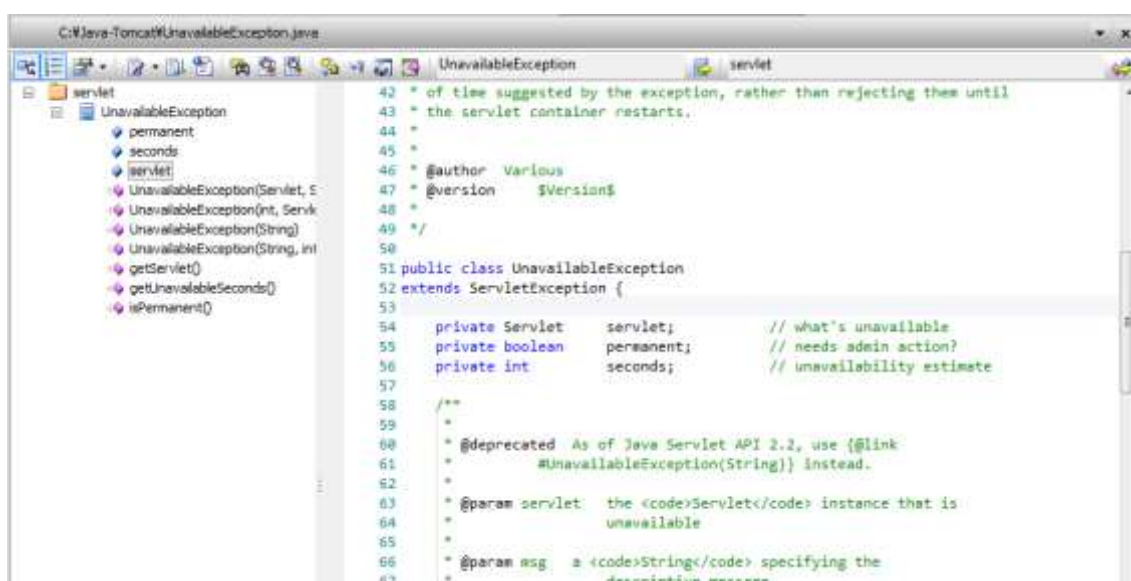
## 8. ソースコードエディタ

Enterprise Architect では、Enterprise Architect の内部でタブとしてソースコードエディタを表示し、ソースコードの参照・編集を行うことができます。この章では、この機能について説明します。

Enterprise Architect の内部でソースコードを表示するには、対象のクラスを右クリックして「ソースコードの表示」を実行して下さい。なお、ソースファイルと関連づけられているクラスのみが対象です。

(ソースファイルから生成したクラス図か、クラス要素からソースファイルを生成したことがある場合に、「ソースコードの表示」の項目を選択することができます。)



ソースコードを表示するタブでは、上部にツールバー・左側に構造ツリー・右側にソースコードが表示されます。



## ソースコードの読み込みと生成 機能ガイド

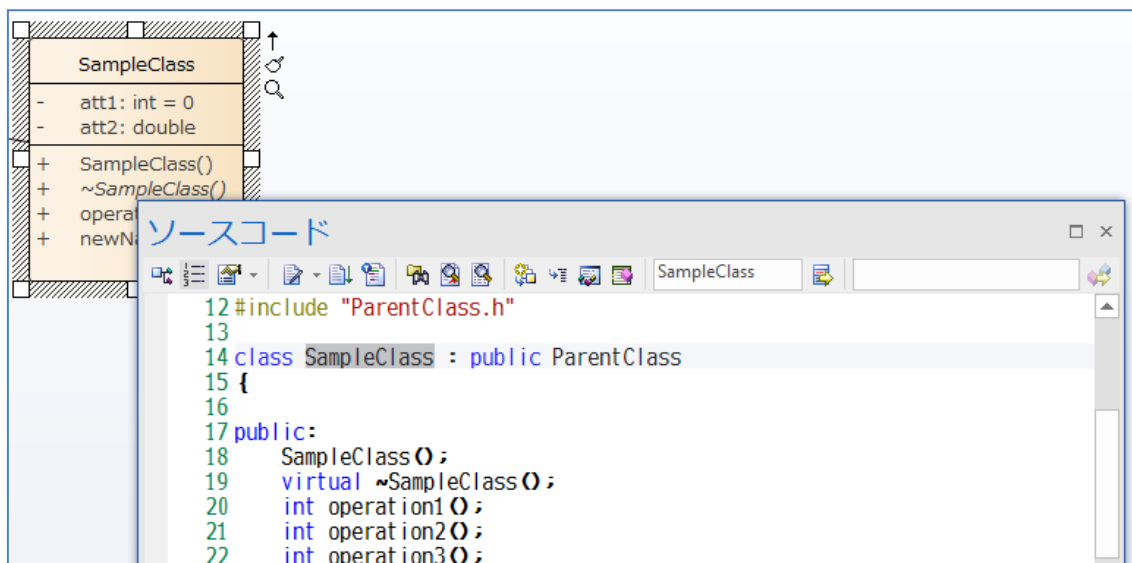
このエディタは、一般的なエディタと同じ操作が可能です。自動的にインデントを制御する機能などが提供されています。また、**Ctrl+Space** キーで、モデル内で定義されている操作名などを挿入することができたり、操作(メソッド)の定義の上にカーソルを載せると引数(パラメータ)の説明が表示されたりするような、コードの編集を効率化するための仕組みも搭載されています。

```
82     */
83
84     public ServletContext getContext(String uripath);
85     ServletContext ServletContext.getContext(String uripath)
86
87
```

また、画面上部のツールバーにあるボタンを利用すると、ソースコードからモデル内の要素の位置へ移動することもできます。 ボタンを押すと、プロジェクトブラウザ内の対象のクラス要素にカーソルが移動します。 ボタンを押すと、プロジェクトブラウザ内の対象の属性や操作にカーソルが移動します。この状態で「利用されているダイアグラム」(**Ctrl+U**)の機能を利用することで、ソースコードから、ダイアグラム内のモデル要素に移動することができます。

なお、ソースコードとモデルの両方を参照しながら作業を行いたい場合には、上記のようなソースコードタブではなく、ソースコードサブウィンドウを利用すると、モデルとソースコードを同時に両方見ることができますので、便利です。

(ソースコードサブウィンドウが表示されている状態で、クラス要素・属性・操作をプロジェクトブラウザやダイアグラム内で選択すると、自動的に関連するソースコードが表示されます。属性や操作を選択した場合には、定義位置にカーソルが移動します。)



## 改版履歴

- 2009/09/01 初版(過去のチュートリアル・スタートアップマニュアルを再構成し、加筆)
- 2009/10/09 コンポーネント図に関する情報が末尾に誤って掲載されていたものを修正。機能についての説明(2章)と読み込みの際のオプション(6章)を追加。
- 2010/03/29 ソースコードの読み込みについて、いくつかの説明を追加。
- 2010/04/16 Enterprise Architect 8.0 のリリースに伴い、内容を更新。7章を追加。
- 2011/05/18 Enterprise Architect 9.0 のリリースに伴い、内容を更新。
- 2011/09/12 「既定のソースコード」欄についての説明を追加。
- 2011/12/06 Enterprise Architect 9.2 のリリースに伴い、内容を更新。
- 2012/03/07 Enterprise Architect 9.3 のリリースに伴い、内容を更新。
- 2012/12/14 Enterprise Architect 10.0 のリリースに伴い、内容を更新。
- 2014/04/22 Enterprise Architect 11.0 のリリースに伴い、内容を更新。
- 2014/06/09 ラウンドトリップに関する情報を6章として追加。
- 2015/02/12 Enterprise Architect バージョン 12.0 リリースに伴い、内容を更新。
- 2015/03/02 5.7章を追加。5章の内容を改善。
- 2015/12/01 Enterprise Architect バージョン 12.1 リリースに伴い、内容を更新。6.2章を追加。
- 2016/10/07 Enterprise Architect バージョン 13.0 リリースに伴い、内容を更新。