



## **Generate and import source codes**

---

*by SparxSystems Japan*

Enterprise Architect 日本語版

ソースコードの生成と読み込み 機能ガイド



## 目次

1. はじめに .....	3
2. ソースコードの生成と読み込み機能の概要 .....	3
2.1. エディションの種類と利用できる機能 .....	3
2.2. 機能の概要 .....	3
2.2.1. ソースコードの生成 .....	3
2.2.2. ソースコードの読み込み .....	4
3. クラス図からソースコードを生成する方法 .....	4
3.1. 単一のクラスに対してコードを生成する .....	5
3.2. まとめてソースコードを生成する .....	7
4. 言語ごとの個別設定 .....	9
5. 既存のソースコードを読み込む方法 .....	9
5.1. ソースディレクトリを指定する方法 .....	10
5.1.1. ソースコードの読み込み .....	10
5.1.2. 読み込み位置と方法の選択 .....	11
5.1.3. 注意点 .....	11
5.1.4. エラーが発生する場合 .....	12
5.1.5. マクロとして認識する文字列の追加 .....	12
5.1.6. クラス図作成の基本的なルール .....	13
5.2. プロジェクトを指定する方法 .....	13
5.2.1. プロジェクトの読み込み .....	14
5.2.2. 読み込む対象と方法の選択 .....	14
5.2.3. 動作解析の設定 .....	15
5.2.4. 注意点 .....	16
6. ラウンドトリップでの利用 .....	16
6.1. 概要 .....	16
6.2. モデルから削除した場合の動作について .....	17
7. ソースコードの読み込みのオプション .....	19
8. ソースコードエディタ .....	21

## 1. はじめに

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

## 2. ソースコードの生成と読み込み機能の概要

### 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 モデルを連携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。

- ・ 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の生成と、既存のソースコードを読み込んでクラス図の生成ができます。

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

- ・ ユニファイド版・アルティメット版

プロフェッショナル版・コーポレート版の内容に加えて、Java, C++, C#, Visual Basic.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。

### 2.2. 機能の概要

#### 2.2.1. 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」レベルの情報のみを出力する、とも言えます。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。

（各操作について「コードエディタ」で入力された内容は、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。既存のソースコードを表示するには、「コード」リボン内の「ソースコード」パネルにある「コードエディタ」ボタンを押し「ソースファイルを開く」を選択してください。)

生成する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 7 章の内容をご覧ください。

### 2.2.2. 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を利用すると、ソースコードからクラス図を生成できます。ただし、既存のソースコードの解析・分析のための機能としては位置づけておらず、一般的なソースコードの解析・分析ツールのようなさまざまな機能(例: 処理の流れをわかりやすく可視化・属性(変数)の利用位置の把握・利用されていないメソッド(関数)の利用位置・呼び出し関係の把握など)は一切ありません。作成したクラス図から生成したソースコードの同期を想定用途としています。

上位の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向のみとなります。)

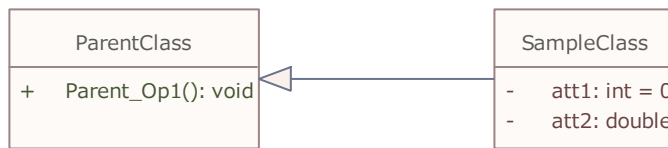
ソースコードの読み込みでは、メソッド(操作)の内部(処理部分)は読み込みの対象になりませんので、処理の流れの把握やメソッドの呼び出し位置の把握などにも利用できません。

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 7 章の内容をご覧ください。

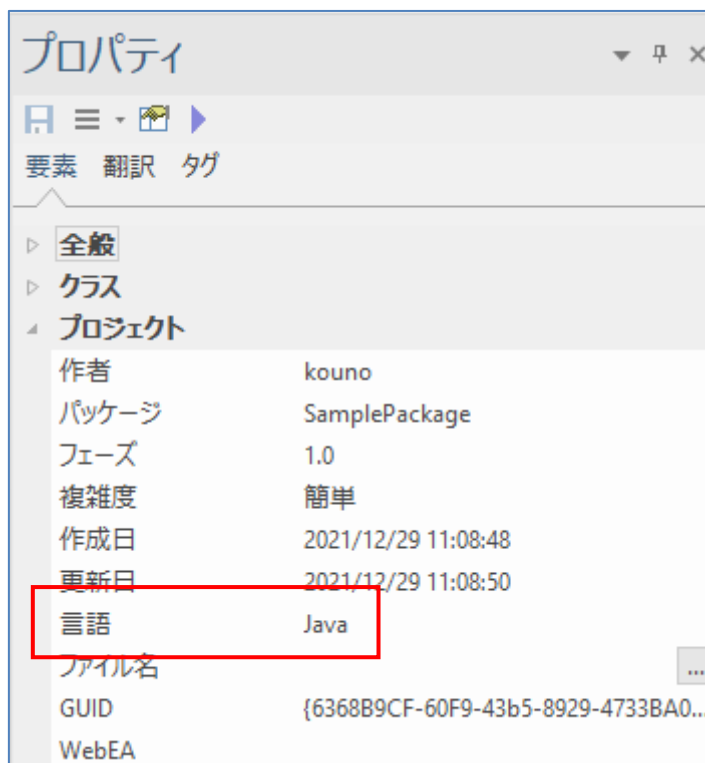
また、Visual Studio, Mono, Eclipse, NetBeans のプロジェクトを指定し、そのプロジェクトで利用されているソースコードを読み込みます。詳細については 5.2 章をご覧ください。

## 3. クラス図からソースコードを生成する方法

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。



クラスのソースコードに関連する設定を行うには、対象のクラスのプロパティサブウィンドウの「プロジェクト」グループにある「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。



### 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。対象のクラス要素をダイアグラム内で選択した状態で、「コード」リボン内の「ソースコード」パネルにある「生成」ボタンを押してください。メニューが表示されますので、「選択した要素」を選択します。

(あるいは、ショートカットキーCtrl+G が利用できます。)

すると、「ソースコードの生成」画面が表示されます。「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。

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、モデルの内容とは別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 などを入力します。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。対象のクラス要素をダイアグラム内で選択した状態で、「コード」リボン内の「ソースコード」パネルにある「コードエディタ」ボタンを押し「ソースファイルを開く」を選択してください。タブとしてソースコードが表示されます。(ショートカットキーCtrl+E や F12 が便利です。)

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「設定」パネルにある「オプション」ボタンを押すと表示されるメニューから「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。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。



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。モデルブラウザから対象となるパッケージを選択し、「コード」リボン内の「ソースコード」パネルにある「生成」ボタンを押してください。メニューが表示されますので、「パッケージ」を選択します。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。

ソースコードの生成(パッケージ)

パッケージ: SamplePackage

同期: 既存のコードと同期

生成:

☐ ファイルパスの自動生成    基準ディレクトリ:    参照

☐ 設定済みのパスを優先

生成対象の要素    ☐ 子パッケージをすべて含む

要素	種類	ターゲットファイル
ParentClass	クラス	
SampleClass	クラス	C:\Users\kouno\Desktop\SampleCl...

すべて選択    すべて解除

生成    キャンセル    ヘルプ

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。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、生成実行中に出力する位置とファイル名を入力できます。

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。

生成:

☒ ファイルパスの自動生成 基準ディレクトリ: C:\src ...

☐ 設定済みのパスを優先

生成対象の要素 ☐ 子パッケージをすべて含む

要素	種類	ターゲットファイル
ParentClass	クラス	C:\src\SamplePackage\ParentClass.java
SampleClass	クラス	C:\src\SamplePackage\SampleClass.java

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 はさまざまな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方法について紹介します。読み込む方法はソースディレクトリを指定する方法と Visual Studio, Mono, Eclipse, NetBeans のプロジェクトを指定する方法の 2 つがあります。それぞれの方法について説明します。

## 5.1. ソースディレクトリを指定する方法

### 5.1.1. 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 章の内容をご覧になり、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モデルブラウザ内で選択後、「コード」リボン内の「ソースコード」パネルにある「読み込み」ボタンを押してください。メニューが表示されますので、「ソースディレクトリの読み込み」を選択します。すると、次のような画面が表示されます。

ソースディレクトリの読み込み

ルートディレクトリ  参照

ソースの種類  拡張子

オプション

- ☐ テスト実行
- ☒ サブディレクトリを再帰的に処理
- ☐ コンポーネントの読み込み 対象: なし
- ☐ privateの属性・操作は対象外
- ☒ マクロの定義がない場合に確認

ログの表示

パッケージ構成

- ☐ ディレクトリごとにパッケージを作成
- ☒ 名前空間ごとにパッケージを作成
- ☐ ファイルごとにパッケージを作成

ダイアグラム

- ☒ パッケージごとにダイアグラムを作成

オプション

同期

- ☒ 既存のクラスを同期
- ☐ 既存のクラスを上書き
- ☐ タイムスタンプを利用

ソースコードに存在しないクラスの処理

- ☒ 削除しない
- ☐ 処理を確認
- ☐ 常に削除

OK キャンセル ヘルプ

### 5.1.2. 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。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、「パッケージごとにダイアグラムを作成」にチェックを入れて下さい。パッケージ(名前空間あるいはディレクトリ)ごとに、その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」ボタンを押すと、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。

### 5.1.3. 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 1,000 個ある場合には、クラス間の関係の最大値は、およそ  $1,000^2 = 1,000,000$  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。

そのため、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」グループにある「区画の表示」パネル内の「属性」「操作」のチェックを外します。あるいは、「属性・操作」グループにある「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「編集」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。

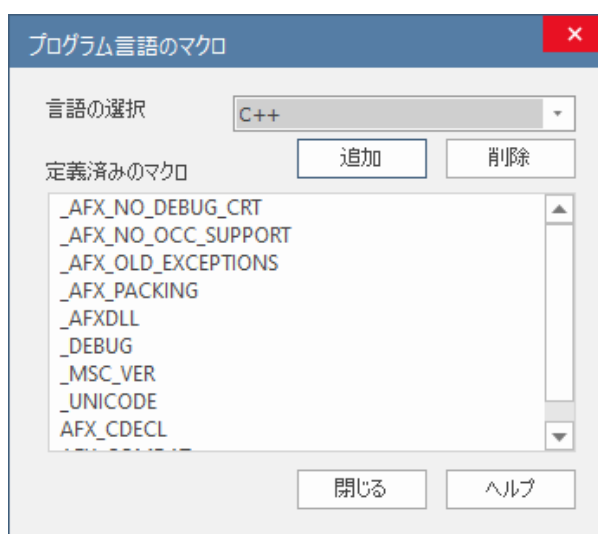
#### 5.1.4. エラーが発生する場合

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。このような場合の多くは、環境に応じた独自のマクロやキーワードなどが原因でソースコードの内容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。)

#### 5.1.5. マクロとして認識する文字列の追加

このようなマクロやキーワードなどがある場合には、「コード」リボン内の「ソースコード」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。次の画面が表示されます。ここで追加した値は、マクロと認識し無視します。



「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やキーワード名を入力してください。

#### 5.1.6. 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 つのソースファイルに対して 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。
- ・ 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。

## 5.2. プロジェクトを指定する方法

Visual Studio, Mono, Eclipse, NetBeans のプロジェクトを指定して既存のソースコードを読み込みます。5.1 章のソースディレクトリを指定する方法と比較すると以下の利点があります。

- ・ソースコードの配置場所を意識せず簡単に読み込める  
プロジェクトやソリューションファイルを指定するだけでソースコードを読み込めるため、ソースコードの実際の配置場所を意識せずに設定できます。
- ・動作解析の設定を自動で追加  
読み込みと同時に動作解析の設定を自動で追加できるため、たとえば読み込み後に Enterprise Architect のデバッグ機能を利用してシーケンス図の自動生成を手軽に行なえます。(シーケンス図の自動生成機能についてはヘルプの「シーケンス図生成機能の概要」のページをご参照ください。)

以下、プロジェクトを指定する方法について Visual Studio 2016 で作成したプロジェクトを読み込む例で説明します。

### 5.2.1. プロジェクトの読み込み

プロジェクトの読み込みを実行する前に、7 章の内容をご覧になり、必要に応じて設定を変更して下さい。

最初に、プロジェクト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モデルブラウザ内で選択後、「コード」リボン内の「ソースコード」パネルにある「プロジェクト」ボタンを押してください。メニューが表示されますので、「Visual Studio のソリューションの読み込み」を選択します。すると、次のような画面が表示されます。



### 5.2.2. 読み込む対象と方法の選択

ソリューションファイルには、読み込む対象のソリューションファイルを選択します。ソリューションに含まれるプロジェクトが一覧表示されますので読み込む対象のプロジェクトにチェックを入れて下さい。

プロジェクトの読み込み時にクラス図を作成する場合には、「パッケージごとにダイアグラムを作成」にチェックを入れて下さい。パッケージ(名前空間あるいはディレクトリ)ごとに、その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。

また、動作解析の設定を自動で追加する場合には、「動作解析の設定を追加」で対象プラットフォームにチェックを入れてください。この設定により読み込み後、手軽に動作解析の機能を利用できます。

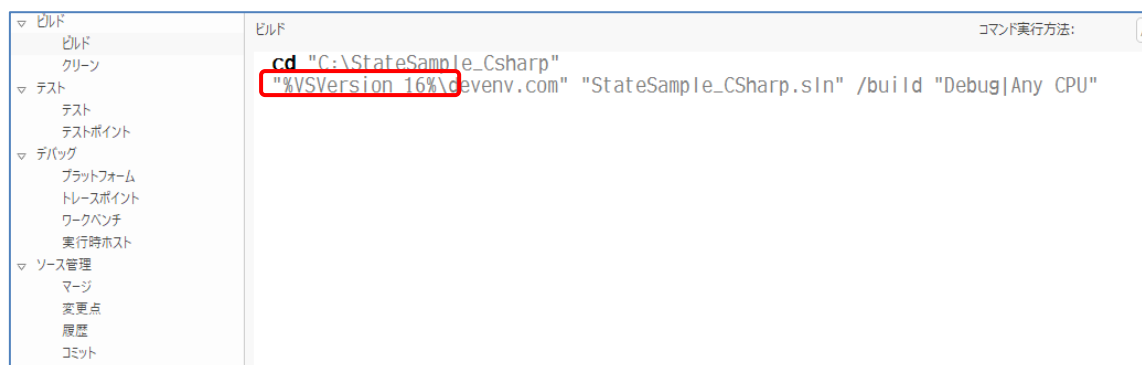
その他の設定項目については、必要に応じて変更して下さい。「読み込み」ボタンを押すと、読み込みが開始されます。

### 5.2.3. 動作解析の設定

5.2.2 章で「動作解析の設定を追加」を行なっている場合、動作解析の設定が自動で追加されます。ただし、下記環境においては設定変更が必要となりますのでご注意ください。

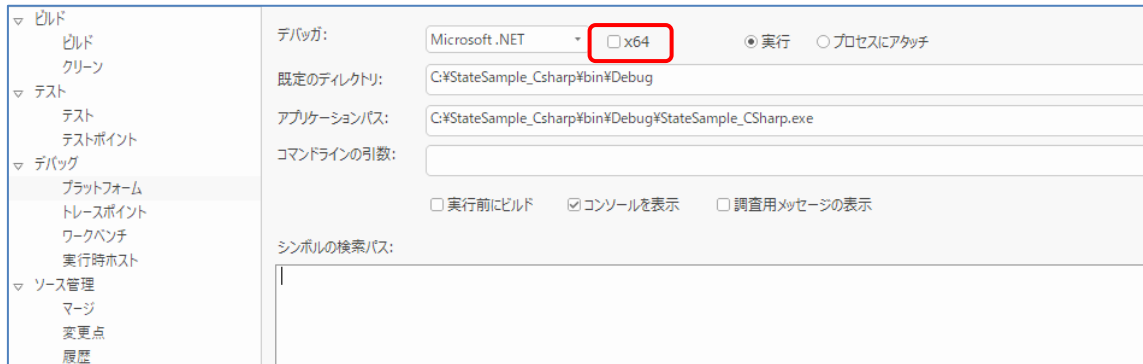
#### ・環境変数の編集

下記画面で IDE のパスに使われている「%VSVersion 16%」のような、この機能で出力される内容に含まれる環境変数については、「C:¥Program Files (x86)¥Microsoft Visual Studio¥2016¥Professional¥Common7¥IDE」のように絶対パスにするか、もしくは Enterprise Architect のローカルパスに変更してください。ローカルパスについてはヘルプの、プログラム  
の動作解析 > ビルドとデバッグ > 動作解析の設定 > ビルド をご覧下さい。



#### ・64bitOS で 32bit アプリケーションをデバッグ対象とする場合

デバッグを開始したときにデバッグサブウィンドウに「この要求はサポートされていません。」と表示されデバッグを開始できないことがあります。その場合には下記画面で「x64」のチェックを有効にしてデバッグを開始してください。



#### 5.2.4. 注意点

- ・プロジェクトの読み込みを実行後、「プロジェクトの再読み込み」を実行した場合は「既存クラスを同期」の動作となります。

「プロジェクトの再読み込み」の方法は、モデルブラウザ内でプロジェクトの読み込み時に指定したパッケージ配下にある Visual Studio アイコンのあるパッケージを選択し、「コード」リボン内の「読み込み」パネルにある「プロジェクト」ボタンを押して表示される「プロジェクトの再読み込み」を実行してください。

- ・5.1.3～5.1.6 章の関連情報も合わせてご覧下さい。

## 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。
- ・ 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」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、「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 サイトから動画をご覧ください。「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

<https://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」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。



表示される内容は、ソースコードを直接変更した場合や、モデルから操作を削除したり操作の名前を変更したりして、ソースコードの内容と整合ができなかったものとなります。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」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が、モデルのみに存在する内容は自動追加となります。)

今回の例では、以下のように、methodInSourceOnly1 については「削除」を、methodInSourceOnly2 については newName への「再割当」(名前を変更した場合で、その変更関係を指示)を指定しました。



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「設定」パネルにある「オプション」ボタンを押すと表示されるメニューから「ユーザー」を選択します。その後、左側の一覧で「属性と操作」を選択して下さい。以下のような画面になります。



- 読み込み時に操作の実装もモデルに読み込み

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、操作のプロパティサブウィンドウにある「コード」タブで参照・編集できます。

なお、もし「コード」タブ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

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「設定」パネルにある「オプション」ボタンを押すと表示されるメニューから「プロジェクト」を選択します。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。

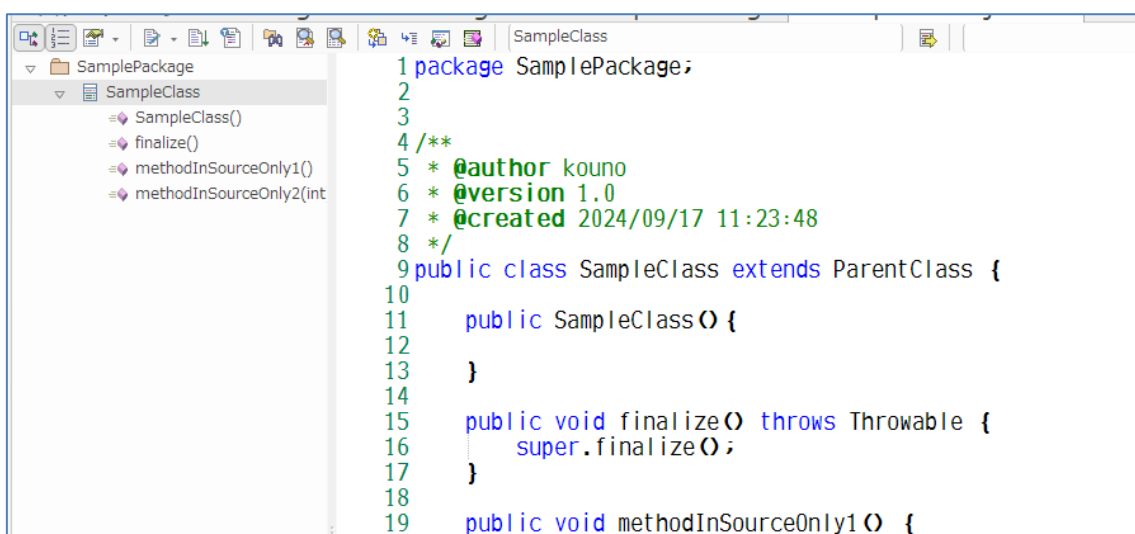
Enterprise Architect の内部でソースコードを表示するには、対象のクラス要素を選択した状態で、「コード」リボン内の「ソースコード」パネルにある「コードエディタ」ボタンを押し「ソースファイルを開く」を選択してください。タブとしてソースコードが表示されます。

(ショートカットキー Ctrl+E や F12 が便利です。)



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実行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。



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。

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できます。 ボタンを押すと、モデル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、モデル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できます。