



**Generate source codes**

---

*by SparxSystems Japan*

Enterprise Architect 日本語版

ソースコードの生成 機能ガイド



## 目次

1. はじめに .....	3
2. ソースコードの生成機能の概要 .....	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
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 .....	6
4. 言語ごとの個別設定 .....	9
5. ラウンドトリップでの利用 .....	9
5.1. 概要 .....	9
5.2. モデルから削除した場合の動作について .....	10
6. ソースコードエディタ .....	12

## 1. はじめに

このドキュメントでは、ソースコードの生成の操作方法について説明します。

## 2. ソースコードの生成機能の概要

### 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

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。

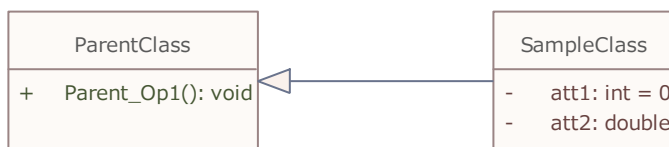
(各操作について「コードエディタ」で入力された内容は、ソースコードの新規生成時に、操作(メソッド)の実装として出力されます。すでにソースコードがある場合の同期処理の場合には、この内容は反映されません。既存のソースコードを表示するには、「コード」リボン内の「ソースコード」パネルにある「コードエディタ」

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。)

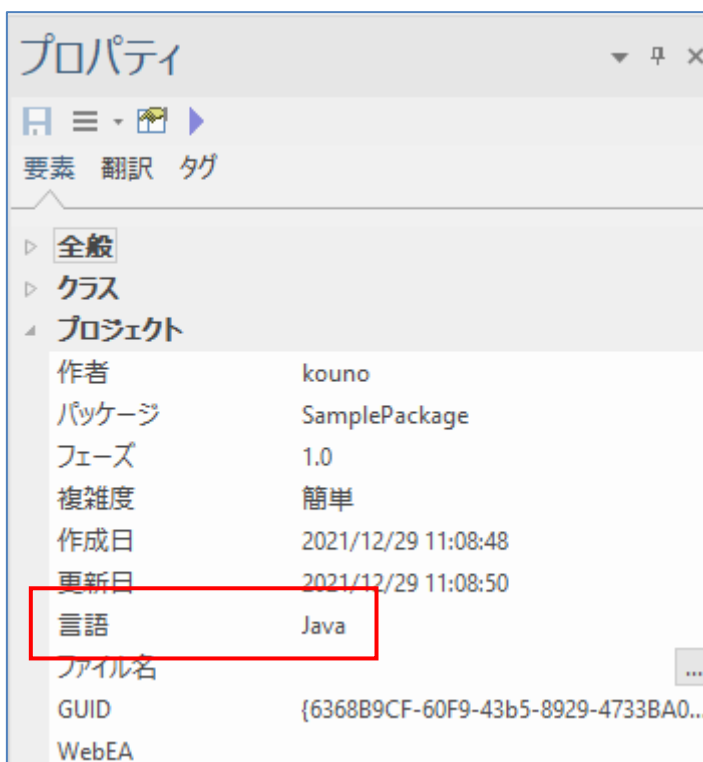
生成するソースコードの文字コードを SHIFT\_JIS 以外の変更したい場合には、事前に 7 章の内容をご覧になり、設定を変更してください。

### 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。

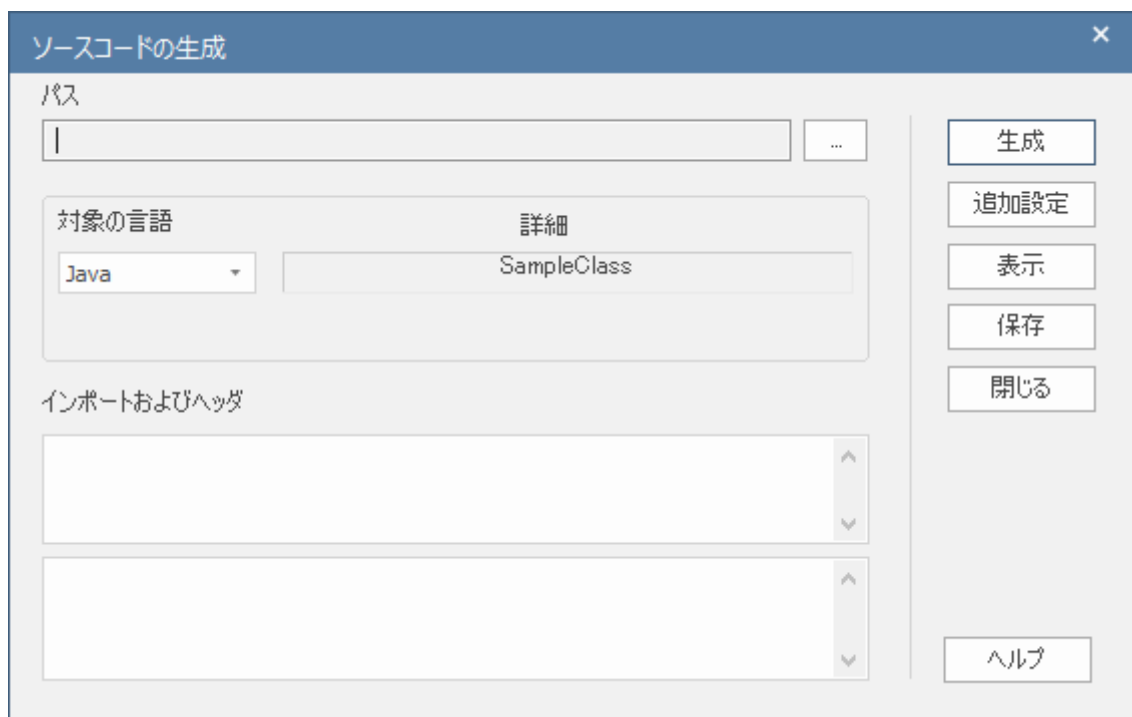


### 3.1. 単一のクラスに対してコードを生成する

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。

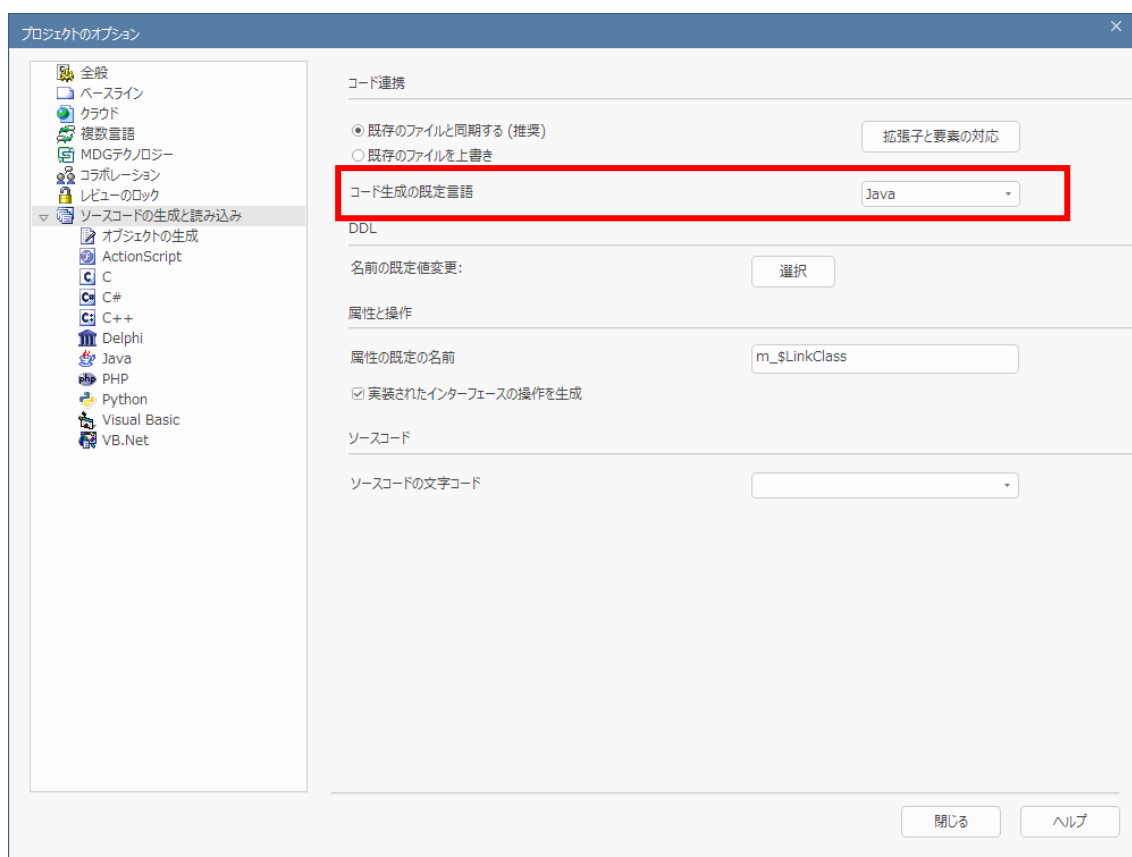
### 3.2. まとめてソースコードを生成する

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。

## ソースコードの生成 機能ガイド



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:  

同期:

生成:

ファイルパスの自動生成 基準ディレクトリ:

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...

この画面で、設定する必要の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。

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. ラウンドトリップでの利用

### 5.1. 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を基準です。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。
- ・ 振り舞い図からのソースコード生成は、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>

## 5.2. 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。

## 6. ソースコードエディタ

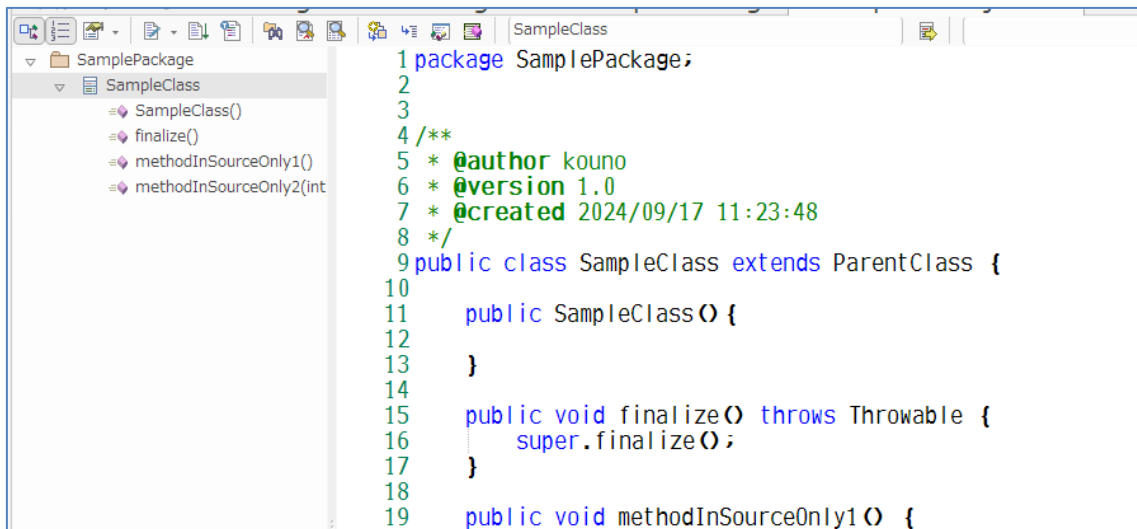
Enterprise Architect では、Enterprise Architect の内部でタブとしてソースコードエディタを表示し、ソースコードの参照・編集を行うことができます。この章では、この機能について説明します。

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。  
(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。



```
1 package SamplePackage;
2
3
4 /**
5  * @author kouno
6  * @version 1.0
7  * @created 2024/09/17 11:23:48
8  */
9 public class SampleClass extends ParentClass {
10
11     public SampleClass() {
12
13     }
14
15     public void finalize() throws Throwable {
16         super.finalize();
17     }
18
19     public void methodInSourceOnly1() {
```

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。