

# C++のコード例とAtScope解析例

2025年9月5日

ビースラッシュ株式会社



- アーキテクチャに従ってフォルダを作り
- EAでクラス図を作成し、コードを作成し、メソッドを追記
- そのコードをAtScopeで図面化

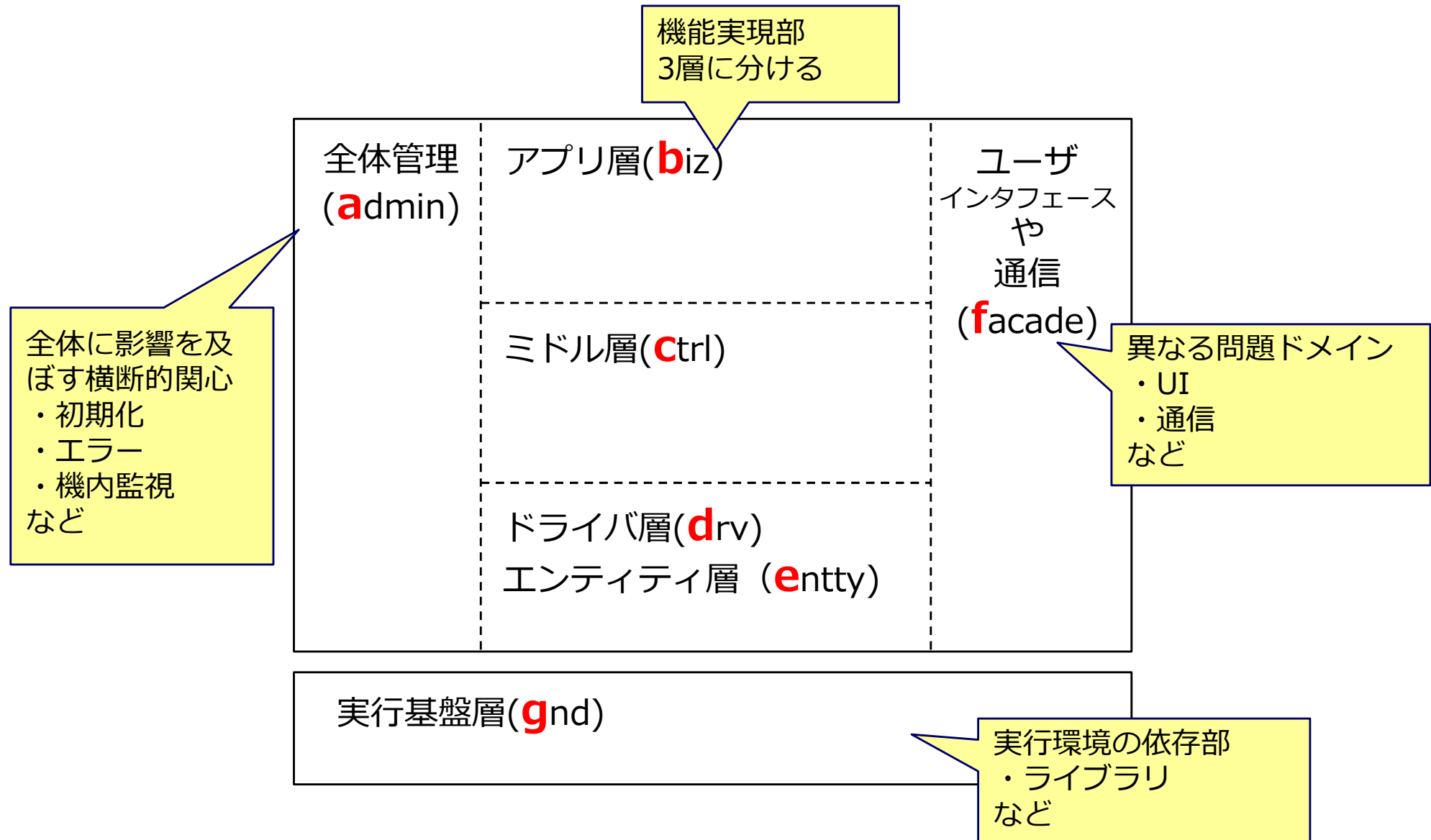


# 1. アーキテクチャに従ってフォルダ作成



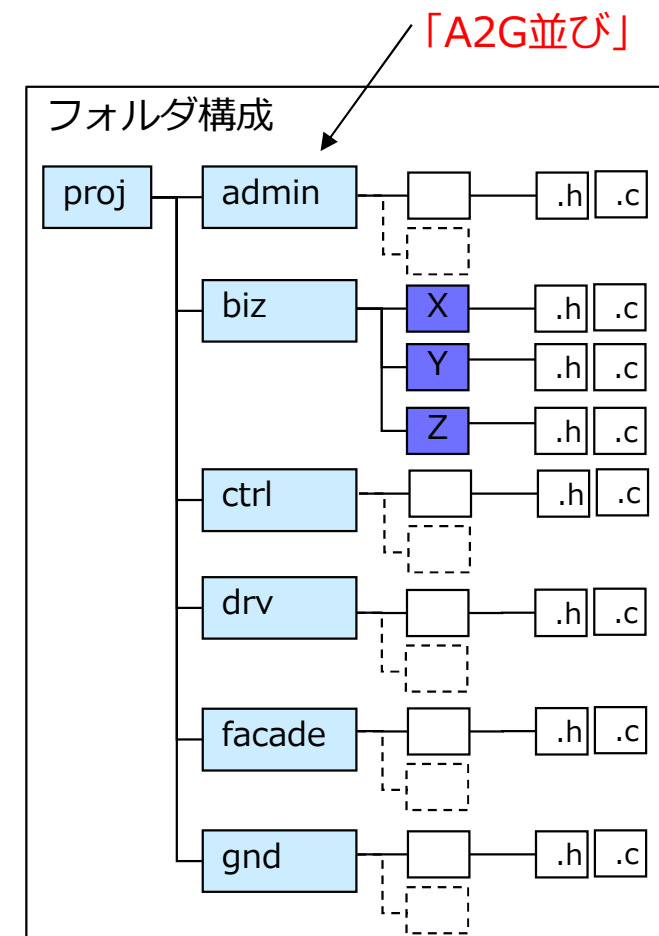
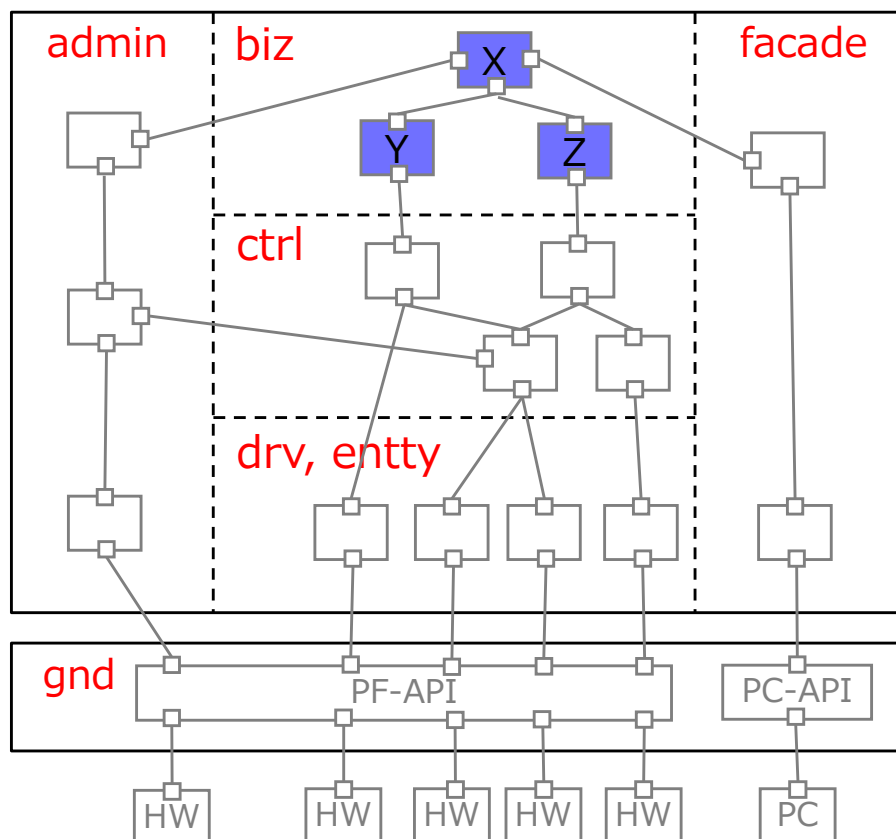
# 静的構造の型：A2Gアーキテクチャ

## ■ 水平垂直分割する



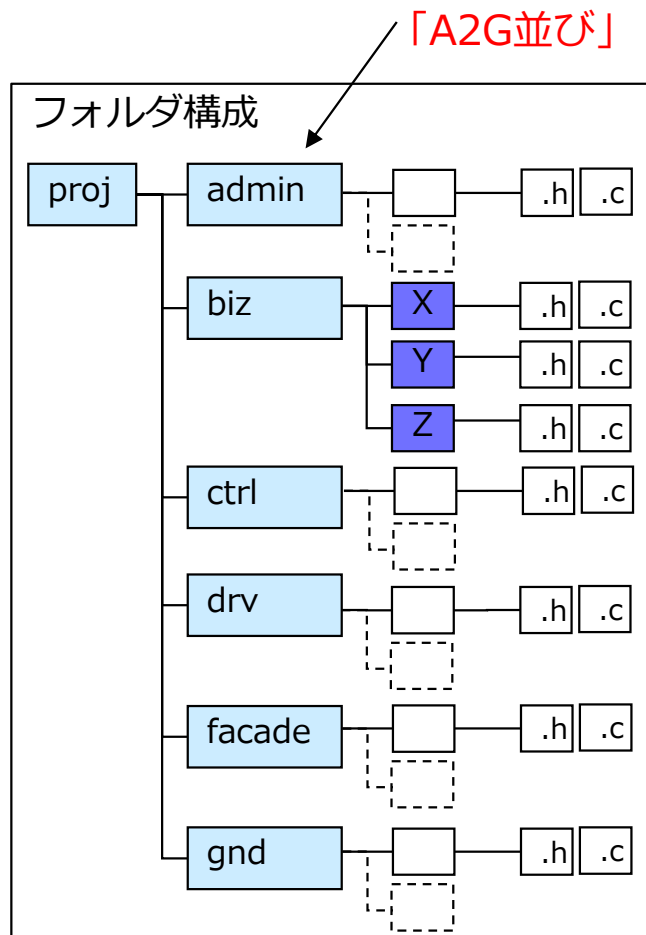
# A2Gアーキテクチャとフォルダ構成

- システム全体を水平垂直分割して、コンポーネントを置く
  - 中央部をアプリ層(biz) – ミドル層(ctrl) – ドライバ層(drv, entty) ----- drvはハードウェア(HW)呼出し  
enttyはデータベース(DB)呼出し
  - 左側に横断的関心(admin)、右側に異なるドメイン(facade)
  - 下部にライブラリ利用(gnd)

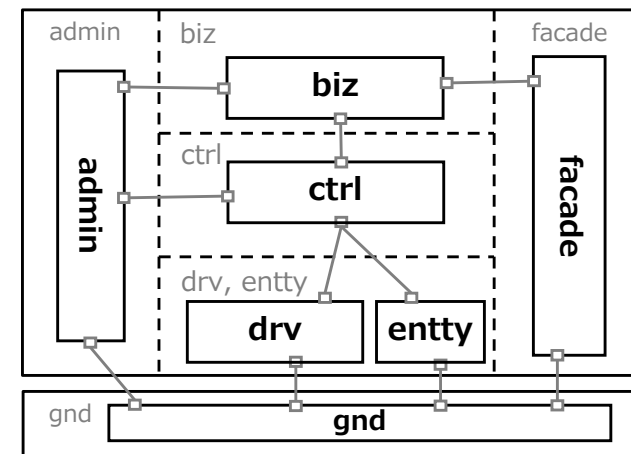


# フォルダ単位の静的構造図

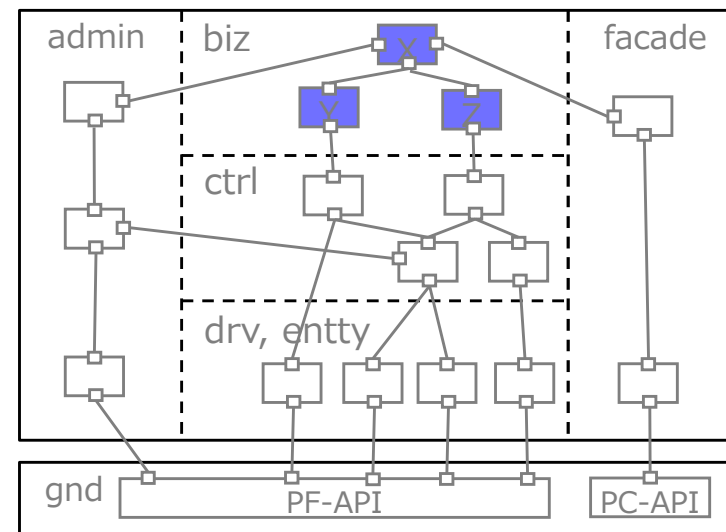
- フォルダ指定場所で**粒度**を変えることができる
  - ルート直下フォルダでは、全体を7個のコンポーネントで図面化
  - 2層目フォルダでは、りより粒度の小さい単位で図面化



ルート直下  
フォルダ

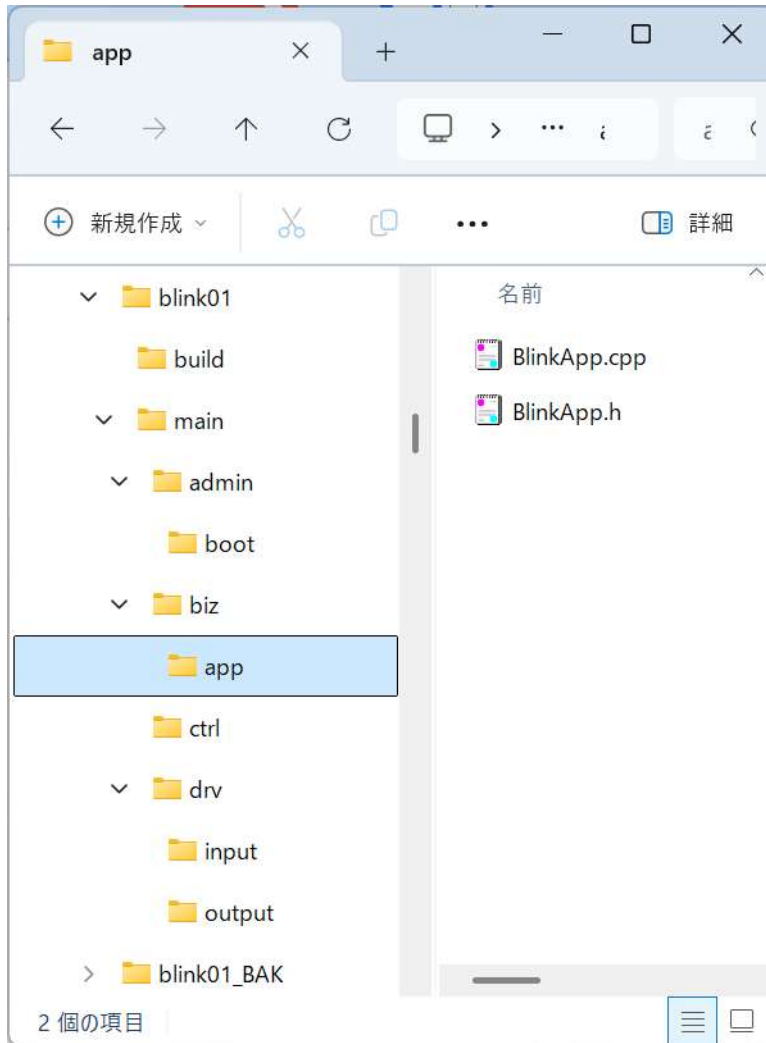


2層目フォルダ



# コード例のフォルダ構成

- 水平垂直分割アーキテクチャを想定してフォルダ作成



## 2. EAでクラス図を作成



class blink01

## ■動作仕様

2つのSWを読んで、2つのLEDを点滅させる



100msec周期起動

staticをtrue  
メソッドにアンダースコア

※データ型はenum定義せずに  
intにしている

## app\_layer::BlinkApp

- alt\_counter: int  
+ run(): void

「コード」→「設定」→「名前空間の基準に設定」  
※但し名前空間は生成コードに反映されない

## hw\_handler::SwDrv

+ read(): int

## hw\_handler::LedDrv

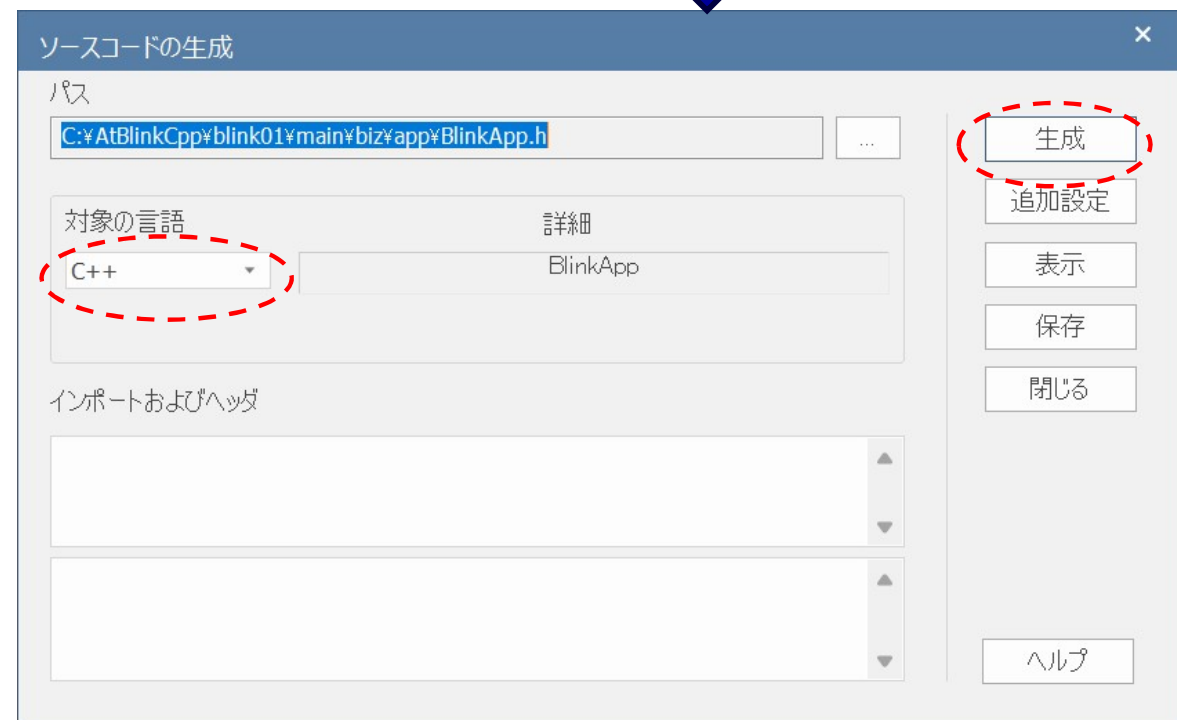
+ on(int): void  
+ off(): void

# 3. EAでスケルトンコードを生成



# EAでのスケルトンコード生成

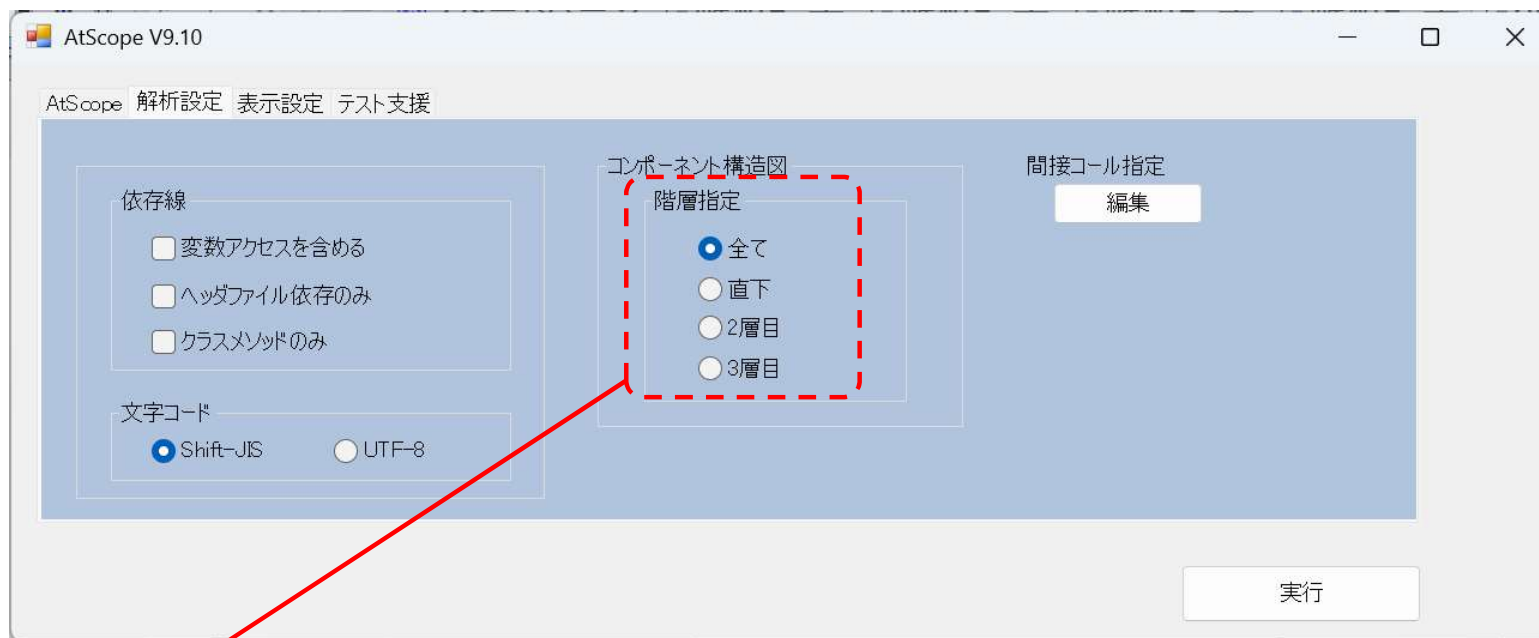
- 「コード」 → 「生成」 → 「選択した要素」 でコード生成



# 5. AtScopeで図面化

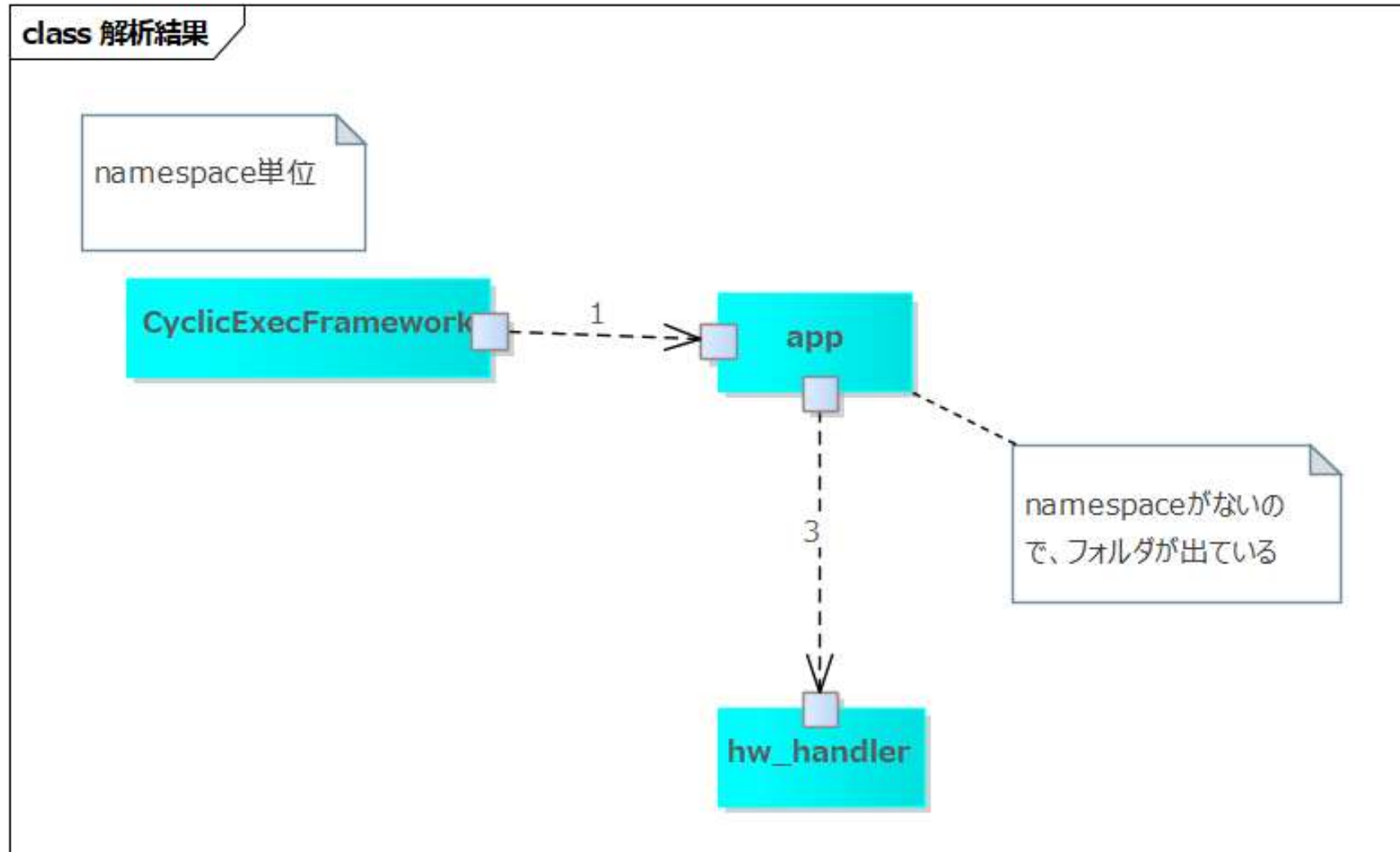


- 「解析設定」→「コンポーネント構造図－階層設定」でコンポーネントの単位を指定します

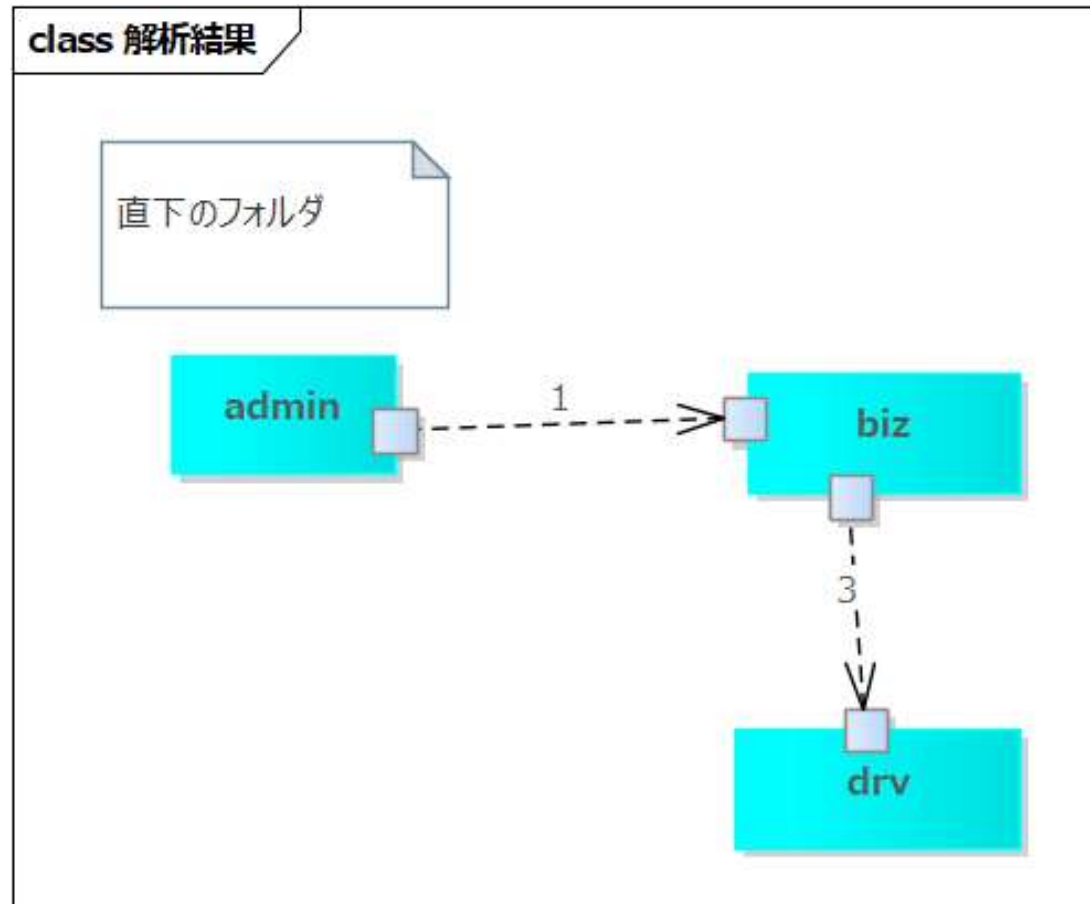


階層指定	コンポーネントの単位	備考
全て	namespace単位	namespaceがない場合はフォルダ
直下	指定したフォルダの直下のフォルダ	フォルダがない場合は何も出ない
2層目	指定したフォルダの2層目のフォルダ	2層目がない場合は直下のフォルダ
3層目	指定したフォルダの3層目のフォルダ	3層目がない時は2層目のフォルダ

# コンポーネント構造図 (全て)



# コンポーネント構造図（直下）



# コンポーネント構造図 (2層目)

