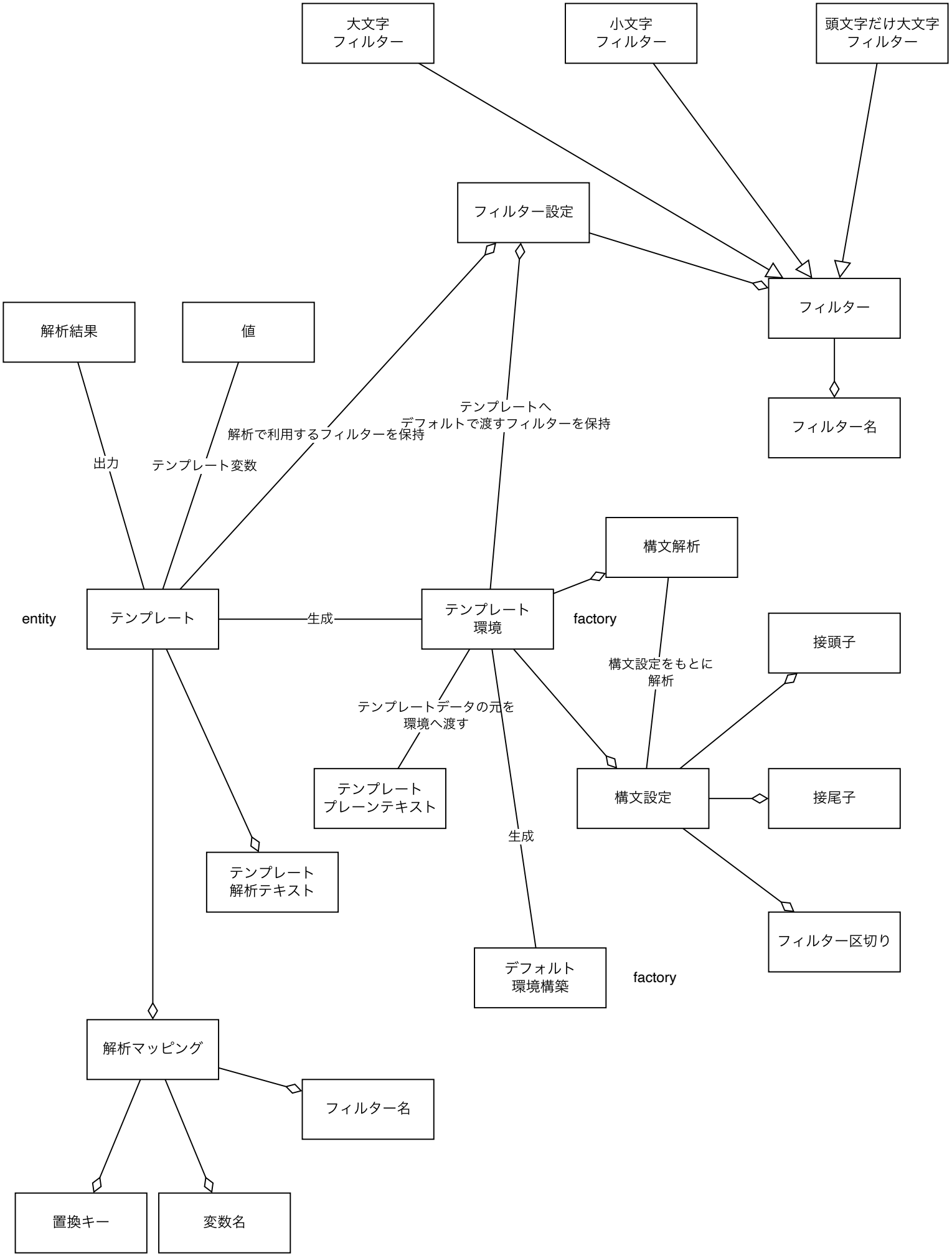
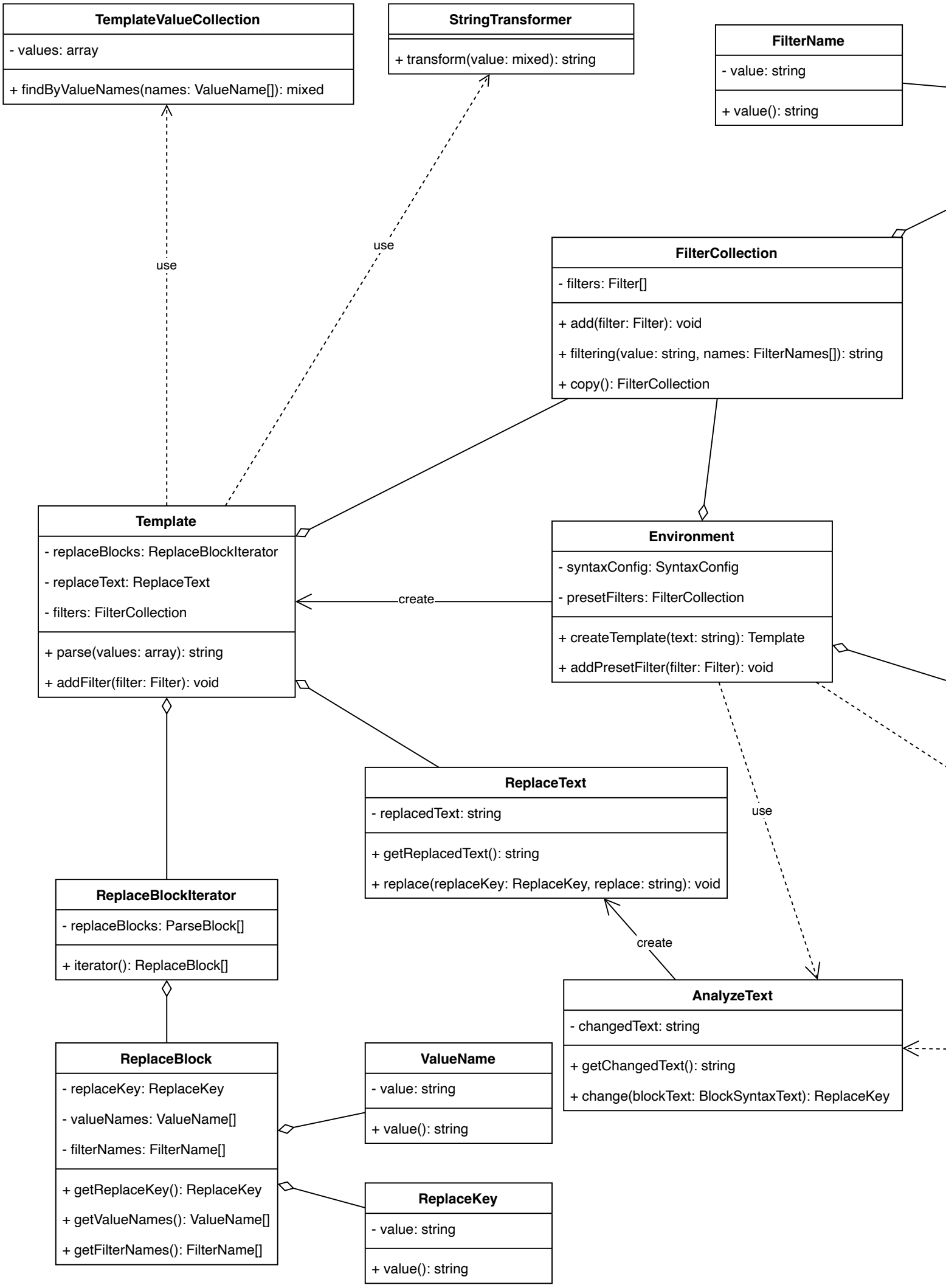
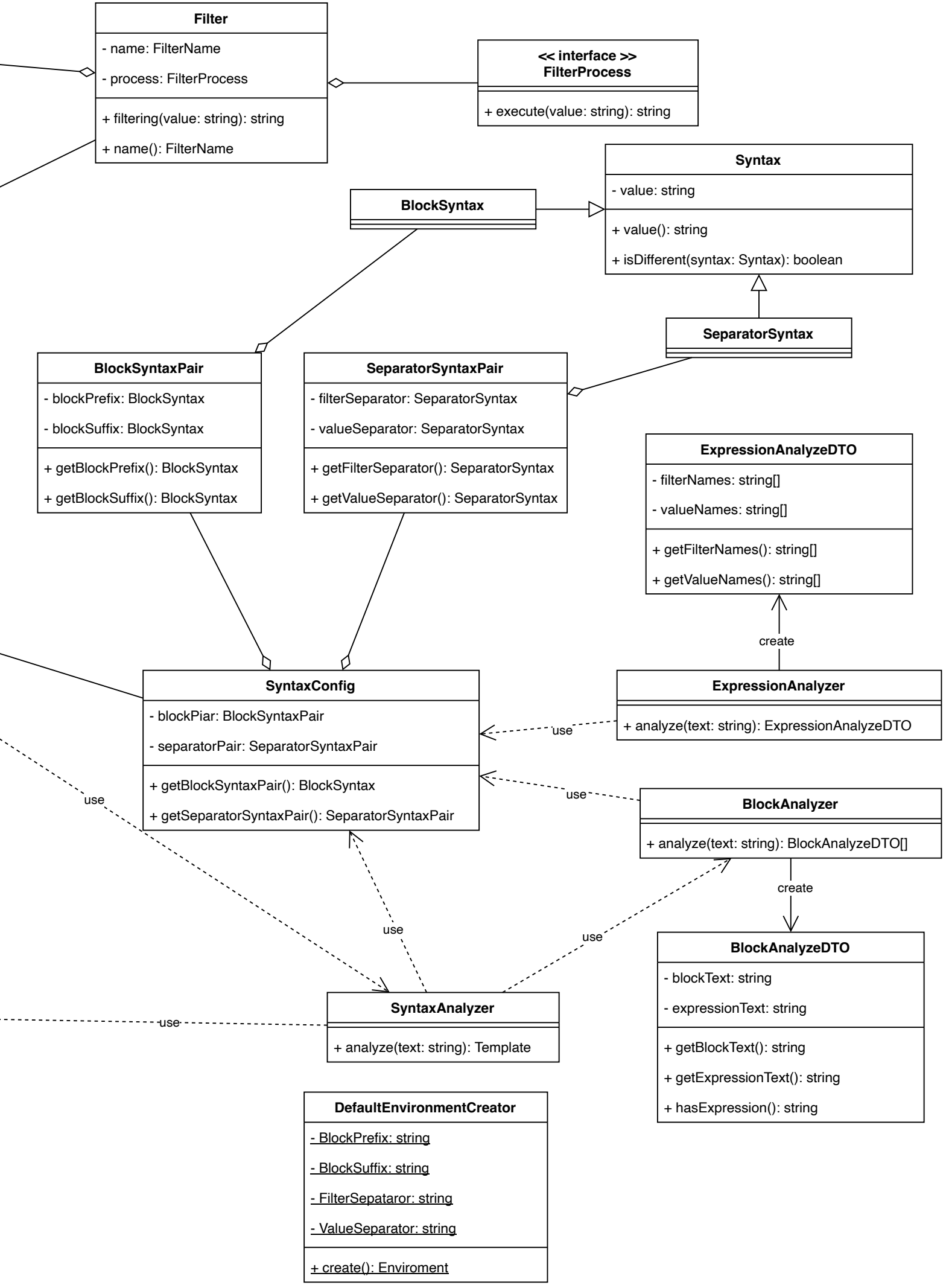

大文字
フィルター
小文字
フィルター
頭文字だけ大文字
フィルター
フィルター設定
フィルター
フィルター名
解析結果
値
出力
テンプレート変数
解析で利用するフィルターを保持
テンプレートへ
デフォルトで渡すフィルターを保持
構文解析
entity
テンプレート
生成
テンプレート
環境
factory
接頭子
構文設定をもとに
解析
テンプレートデータの元を
環境へ渡す
テンプレート
プレーンテキスト
構文設定
接尾子
生成
テンプレート
解析テキスト
フィルター区切り
デフォルト
環境構築
factory
解析マッピング
フィルター名
置換キー
変数名


TemplateValueCollection
- values: array
+ findByValueNames(names: ValueName[]): mixed
StringTransformer
+ transform(value: mixed): string
FilterName
- value: string
+ value(): string
use
use
FilterCollection
- filters: Filter[]
+ add(filter: Filter): void
+ filtering(value: string, names: FilterNames[]): string
+ copy(): FilterCollection
Template
- replaceBlocks: ReplaceBlockIterator
- replaceText: ReplaceText
- filters: FilterCollection
+ parse(values: array): string
+ addFilter(filter: Filter): void
create
Environment
- syntaxConfig: SyntaxConfig
- presetFilters: FilterCollection
+ createTemplate(text: string): Template
+ addPresetFilter(filter: Filter): void
ReplaceText
- replacedText: string
+ getReplacedText(): string
+ replace(replaceKey: ReplaceKey, replace: string): void
use
ReplaceBlockIterator
- replaceBlocks: ParseBlock[]
+ iterator(): ReplaceBlock[]
create
AnalyzeText
- changedText: string
+ getChangedText(): string
+ change(blockText: BlockSyntaxText): ReplaceKey
ReplaceBlock
- replaceKey: ReplaceKey
- valueNames: ValueName[]
- filterNames: FilterName[]
+ getReplaceKey(): ReplaceKey
+ getValueNames(): ValueName[]
+ getFilterNames(): FilterName[]
ValueName
- value: string
+ value(): string
ReplaceKey
- value: string
+ value(): string


Filter
- name: FilterName
- process: FilterProcess
+ filtering(value: string): string
+ name(): FilterName
<< interface >>
FilterProcess
+ execute(value: string): string
Syntax
- value: string
+ value(): string
+ isDifferent(syntax: Syntax): boolean
BlockSyntax
SeparatorSyntax
BlockSyntaxPair
- blockPrefix: BlockSyntax
- blockSuffix: BlockSyntax
+ getBlockPrefix(): BlockSyntax
+ getBlockSuffix(): BlockSyntax
SeparatorSyntaxPair
- filterSeparator: SeparatorSyntax
- valueSeparator: SeparatorSyntax
+ getFilterSeparator(): SeparatorSyntax
+ getValueSeparator(): SeparatorSyntax
ExpressionAnalyzeDTO
- filterNames: string[]
- valueNames: string[]
+ getFilterNames(): string[]
+ getValueNames(): string[]
create
SyntaxConfig
- blockPiar: BlockSyntaxPair
- separatorPair: SeparatorSyntaxPair
+ getBlockSyntaxPair(): BlockSyntax
+ getSeparatorSyntaxPair(): SeparatorSyntaxPair
ExpressionAnalyzer
+ analyze(text: string): ExpressionAnalyzeDTO
use
use
use
BlockAnalyzer
+ analyze(text: string): BlockAnalyzeDTO[]
create
use
use
use
BlockAnalyzeDTO
- blockText: string
- expressionText: string
+ getBlockText(): string
+ getExpressionText(): string
+ hasExpression(): string
SyntaxAnalyzer
+ analyze(text: string): Template
use
DefaultEnvironmentCreator
- BlockPrefix: string
- BlockSuffix: string
- FilterSepataror: string
- ValueSeparator: string
+ create(): Enviroment