

JAVA プログラム呼び出しライブラリ

1. ライブラリ要件の定義

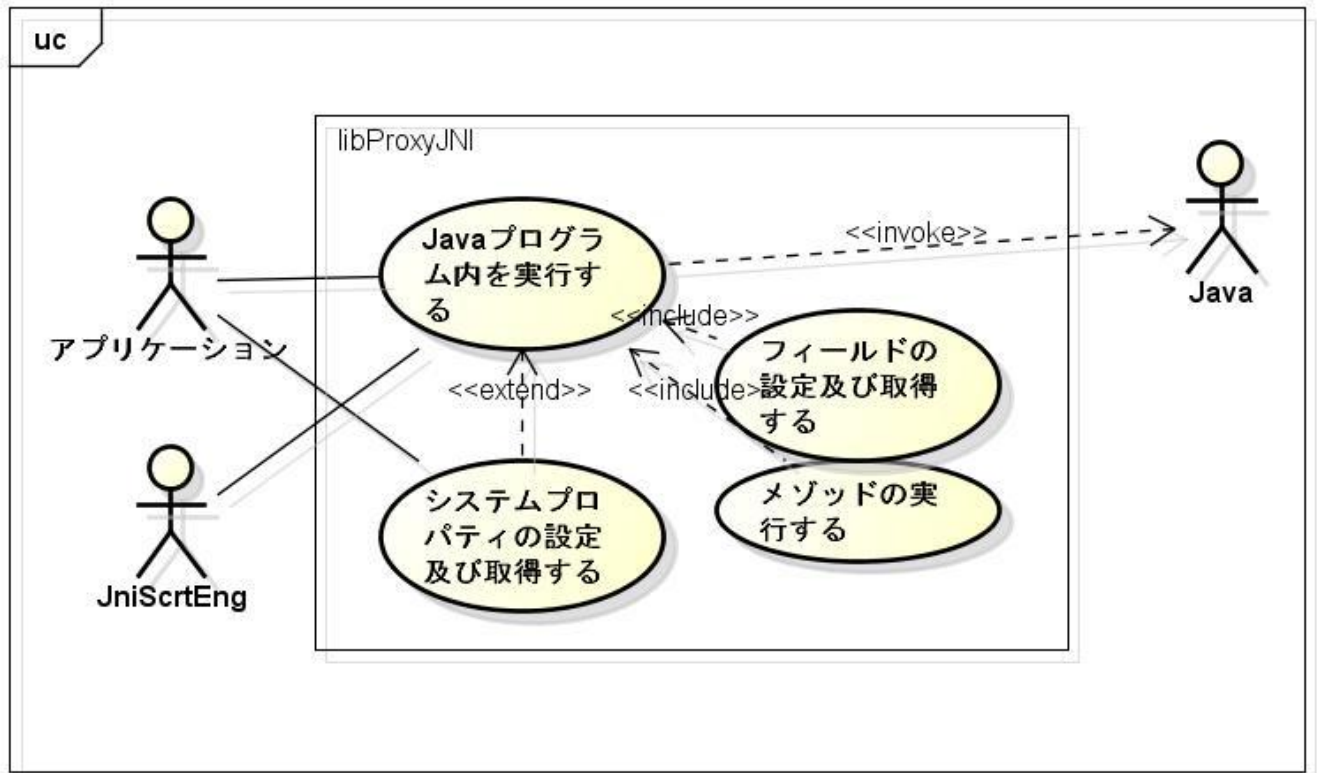
(ア) ライブラリ概要

① 背景と目的

目的

アプリケーションから、JNI ライブラリを利用して、Java クラスのメソッドを実行する為のライブラリです

(イ) システム化の範囲



2. ライブラリ機能要件の定義

(ア) ライブラリ機能要件

ユースケース名		Java プログラム内のメソッドを実行する
目的		アプリケーションから java プログラム内のメソッドを呼び出す
アクター		アプリケーション、JniSqrtEng
事前条件		Java ソフトがインストールしている
事後条件		Java プログラム内のメソッドを呼び出される
フロー	STEP	アクション
基本フロー	1	Java プログラムの初期化する
	2	Java プログラムのメソッドを実行する
	3	Java プログラムの終了する
代替フロー		
例外フロー		

ユースケース名		Java クラス内のフィールドの設定及び取得する
目的		アプリケーションから java プログラム内のフィールド値の設定及び取得出来る
アクター		アプリケーション、JniSqrtEng
事前条件		Java ソフトがインストールしている
事後条件		Java クラス内のフィールド値の設定/取得する
フロー	STEP	アクション
基本フロー		Java プログラムの初期化する
		Java クラス内のフィールドの設定及び取得する
		Java プログラムの終了する
代替フロー		
例外フロー		

ユースケース名		システムプロパティの設定及び取得する
目的		アプリケーションから Java システムプロパティの操作したい
アクター		アプリケーション
事前条件		Java ソフトがインストールしている
事後条件		System プロパティの設定/取得される
フロー	STEP	アクション
基本フロー	1	Java プログラムの初期化する
	2	クラスパスの設定する
	3	システムプロパティの取得する
	4	Java プログラムの終了する
代替フロー		
例外フロー		

3. ライブラリ非機能要件の定義

(ア) システム方式要件

① システム構成の方針

- ・ソフトウェア製品の活用方式
広く流通し、利用実績のあるものを活用する

② システム全体構成

OS の種類 : linux

ミドルウェア:java version 11

(イ) テスト要件

- ① システムの利用開始前に十分なテスト期間を確保し、信頼性の確認を行うこと。
- ② システム利用開始後であっても、テスト不足を認められる場合には、必要なテストを実施すること。また、その結果、システムが本業務仕様に適用しない事実が発見された時は、速やかに、見直しを行うこと。

4. 詳細設計

Java プログラム内のメソッド実行機能 USDM			
Req-01	要求	アプリケーションから、Java プログラム内のメソッドを実行する	
	理由		
	説明	JavaVM のライブラリを利用して、メソッドを実行する メソッドの種類は、以下です。 ・スタティックメソッド ・オブジェクトメソッド データ型は、下記(※1)です	
	<JavaVM の生成>		
	Spec-01-1	JNI ライブラリを利用して、JavaVM を生成する	
	<メソッド実行>		
	要求	Req01-01	メソッドを実行する
		理由	
		説明	Java クラス内にパラメータがある場合は、パラメータを設定し実行する。 戻り値がある場合は、アプリケーションへ受け取る。
	<スタティックメソッド実行>		
		Spec01-01-1	パラメータがある場合は、パラメータを設定する
		Spec01-01-2	スタティックメソッドを実行する
		Spec01-01-3	戻り値がある場合は、結果を受け取る
	<オブジェクトメソッド実行>		
		Spec01-01-4	パラメータがある場合は、パラメータを設定する
		Spec01-01-5	クラスのインスタンスを生成する
		Spec01-01-6	オブジェクトメソッドを実行する
		Spec01-01-7	戻り値がある場合は、結果を受け取る
	<JavaVM の終了>		
	Spec-01-2	JavaVM を終了する	

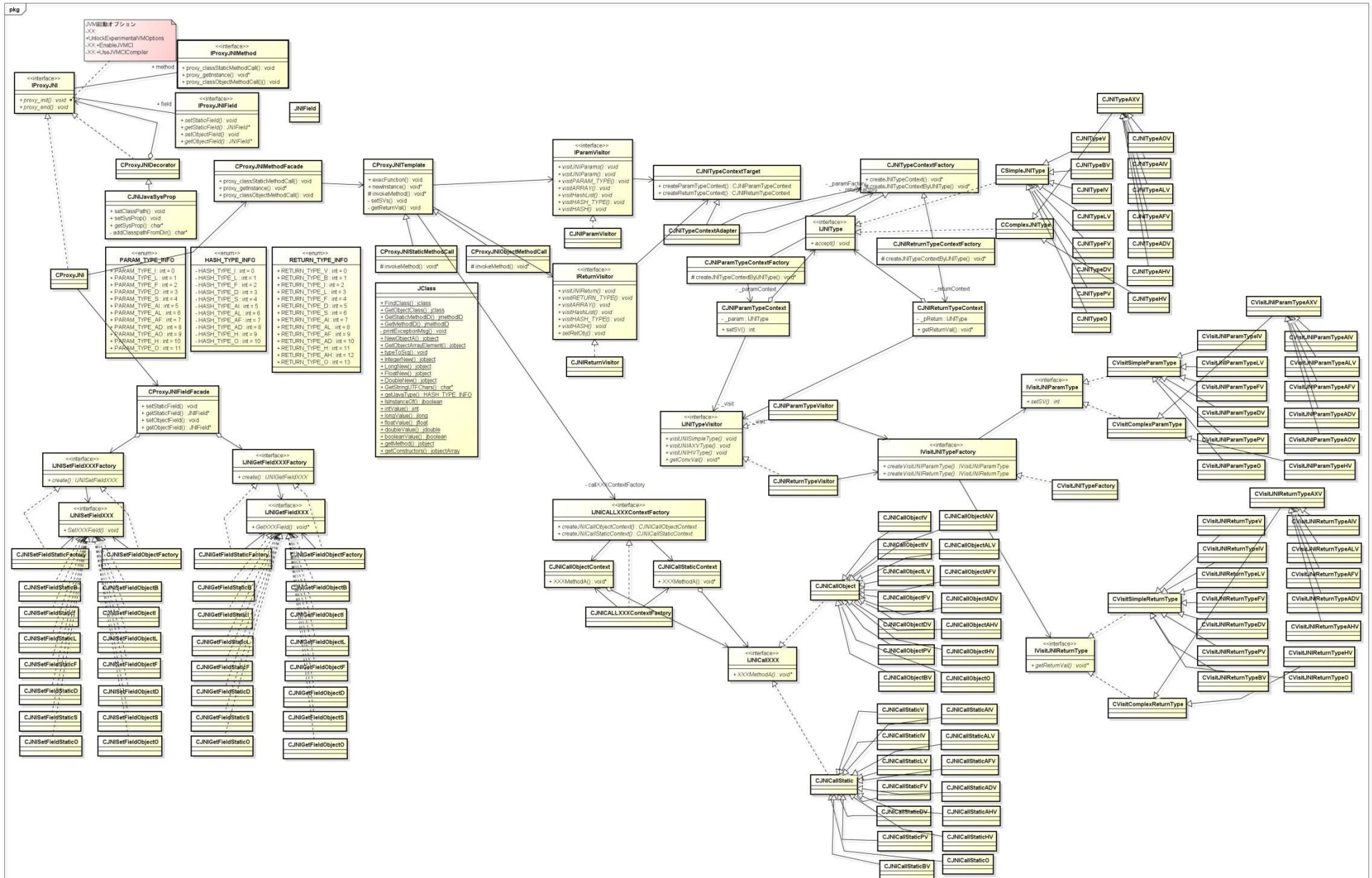
Java クラス内のフィールドの設定及び取得機能 USDM		
Req-02	要求	アプリケーションから、Java クラス内のフィールドの設定及び取得する
	理由	
	説明	<p>JavaVM ライブラリを利用して、Java クラス内のフィールドの設定及び取得するフィールドの種類は、以下です。</p> <ul style="list-style-type: none"> ・スタティックフィールド ・オブジェクトフィールド <p>データ型は、下記(※1)です</p>
<JavaVM の生成>		
	Spec-02-1	JNI ライブラリを利用して、JavaVM を生成する
<スタティックフィールド>		
	Spec02-2	スタティックフィールド値へ設定する
	Spec02-3	スタティックフィールド値を取得する
<オブジェクトフィールド>		
	Spec02-4	オブジェクトフィールド値へ設定する
	Spec02-5	オブジェクトフィールド値を取得する
<JavaVM の終了>		
	Spec-01-2	JavaVM を終了する

システムプロパティの設定及び取得機能 USDM			
Req-03	要求	アプリケーションから、System クラスのプロパティの設定及び取得する	
	理由		
	説明	Java の System クラスのプロパティを利用して、クラスパスの設定もする	
	＜JavaVM の生成＞		
	Spec-03-1	JNI ライブラリを利用して、JavaVM を生成する	
	＜クラスパスの設定＞		
	要求	Req-03-01	クラスパスの設定をする
		理由	
		説明	ディレクトリ内のファイルをクラスパスに設定する
		Spec03-01-1	ディレクトリ内のファイルを取得する
		Spec03-01-2	System プロパティにクラスパスを設定する ・ Key: java.class.path ・ Value: ディレクトリ内のファイル名
	＜System プロパティ値の取得＞		
	Spec-03-2	System プロパティ値を取得	
	＜JavaVM の終了＞		
	Spec-03-3	JavaVM を終了する	

※1 データ型

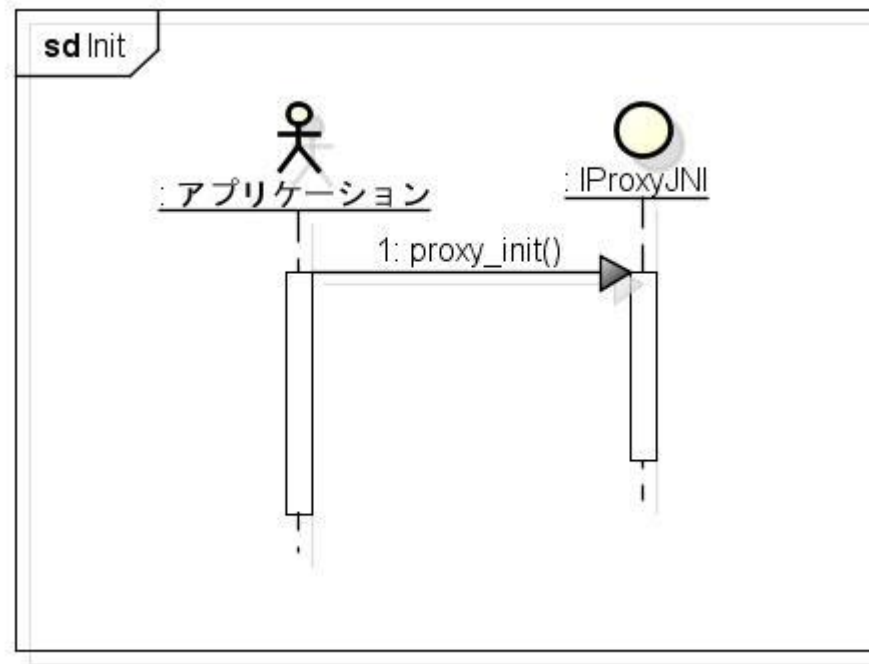
パラメータ値型	ハッシュ値型	戻り値型	フィールド値型
		戻り値無し	
		Boolean 型	Boolean 型
Int 型	Int 型	Int 型	Int 型
Long 型	Long 型	Long 型	Long 型
Float 型	Float 型	Float 型	Float 型
Double 型	Double 型	Double 型	Double 型
文字列	文字列	文字列	文字列
Int 型の配列	Int 型の配列	Int 型の配列	
Long 型の配列	Long 型の配列	Long 型の配列	
Float 型の配列	Float 型の配列	Float 型の配列	
Double 型の配列	Double 型の配列	Double 型の配列	
オブジェクト型の配列			
ハッシュ型	ハッシュ型	ハッシュ型	
		ハッシュ型の配列	
オブジェクト型	オブジェクト型	オブジェクト型	オブジェクト型

(ア) クラス図

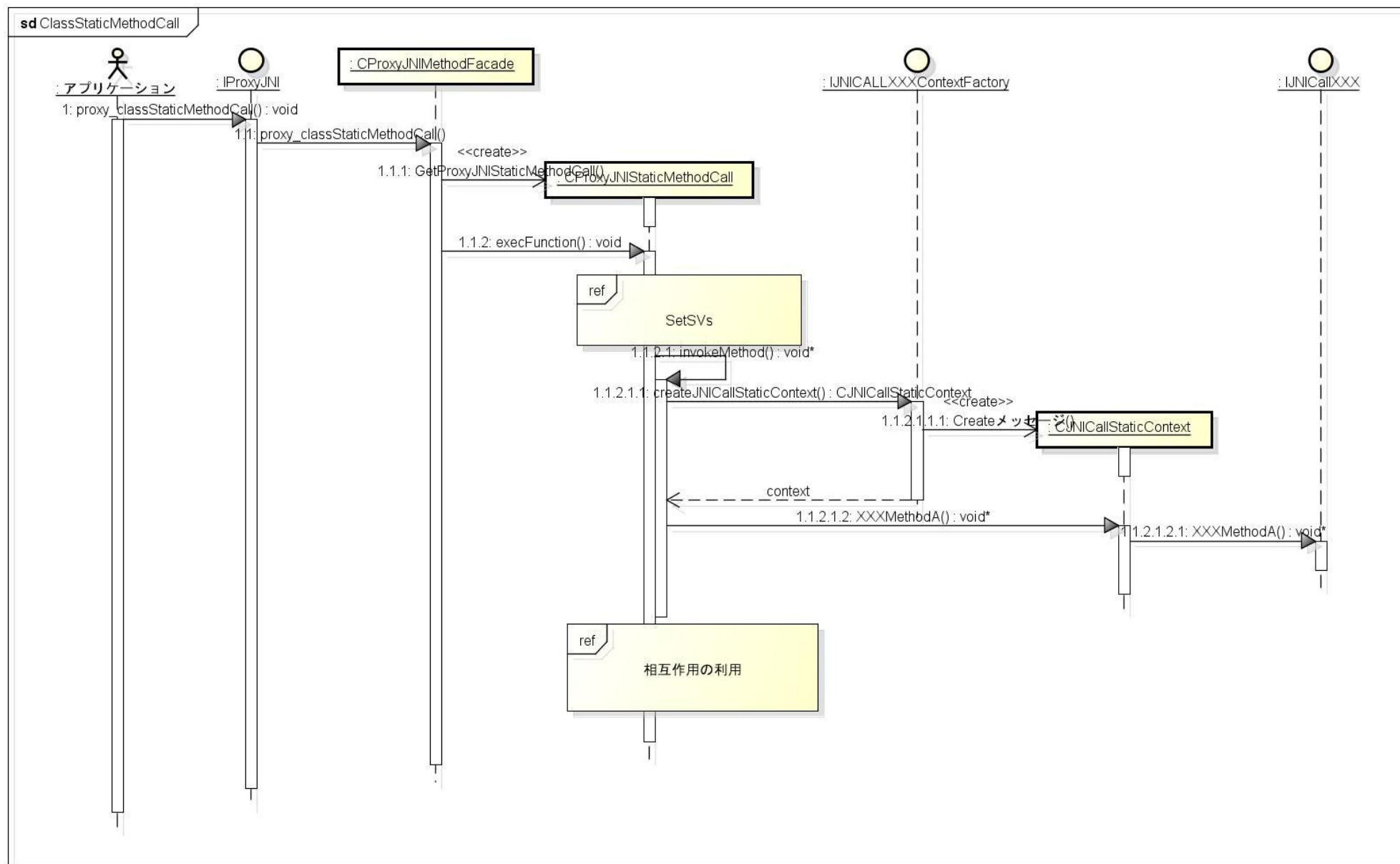


(イ) シーケンス図

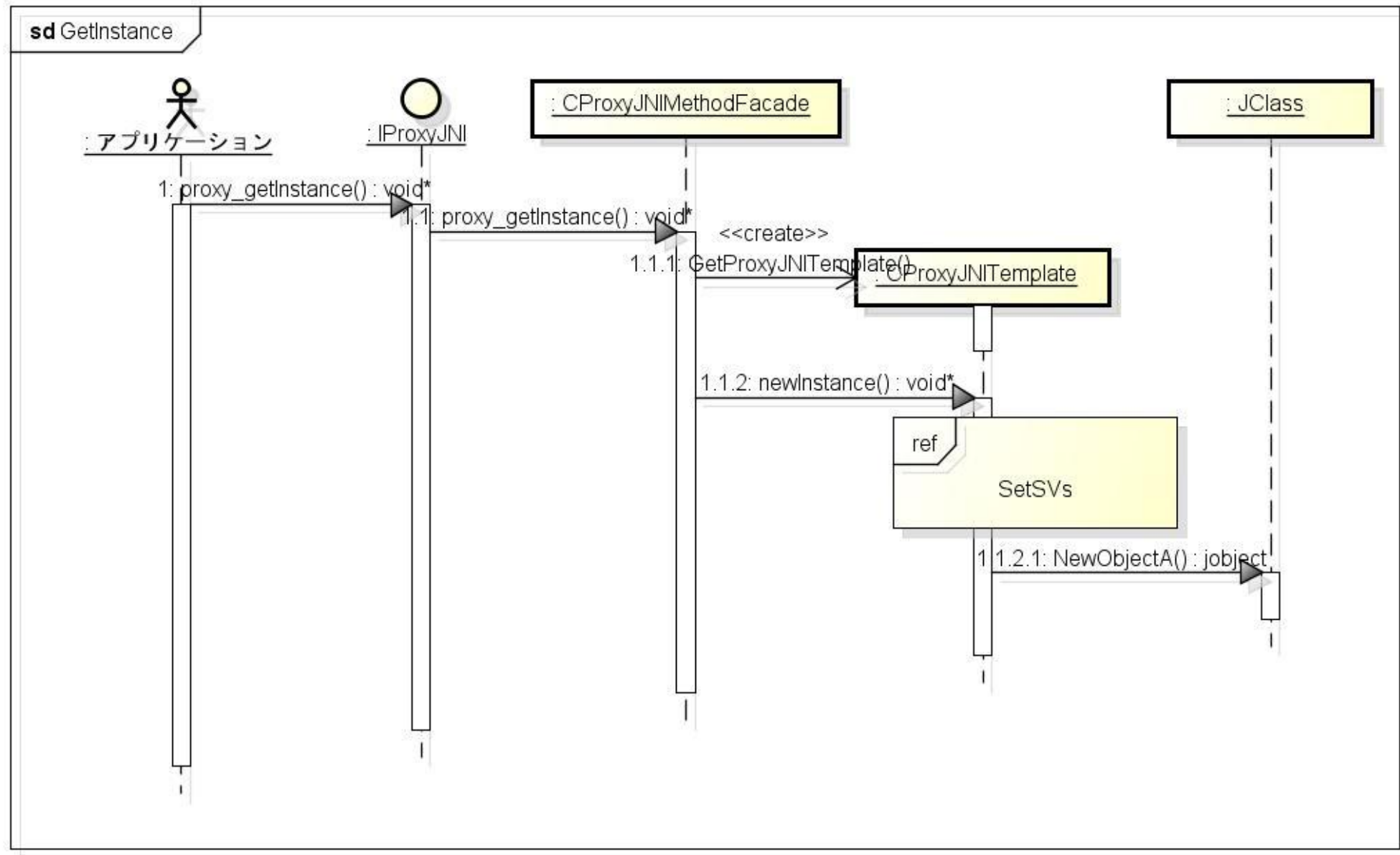
① JNI の初期化



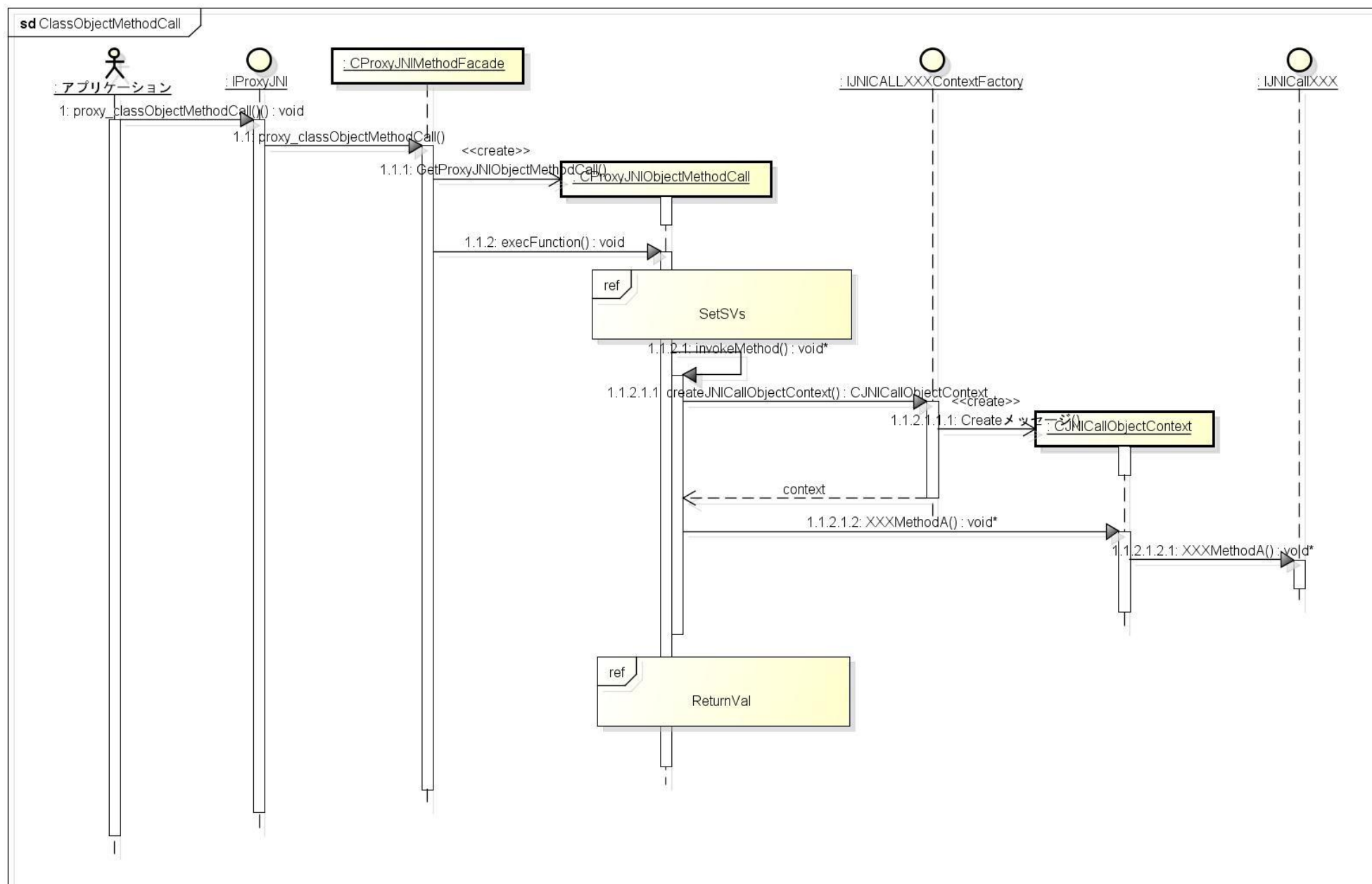
② スタティックメソッドを実行



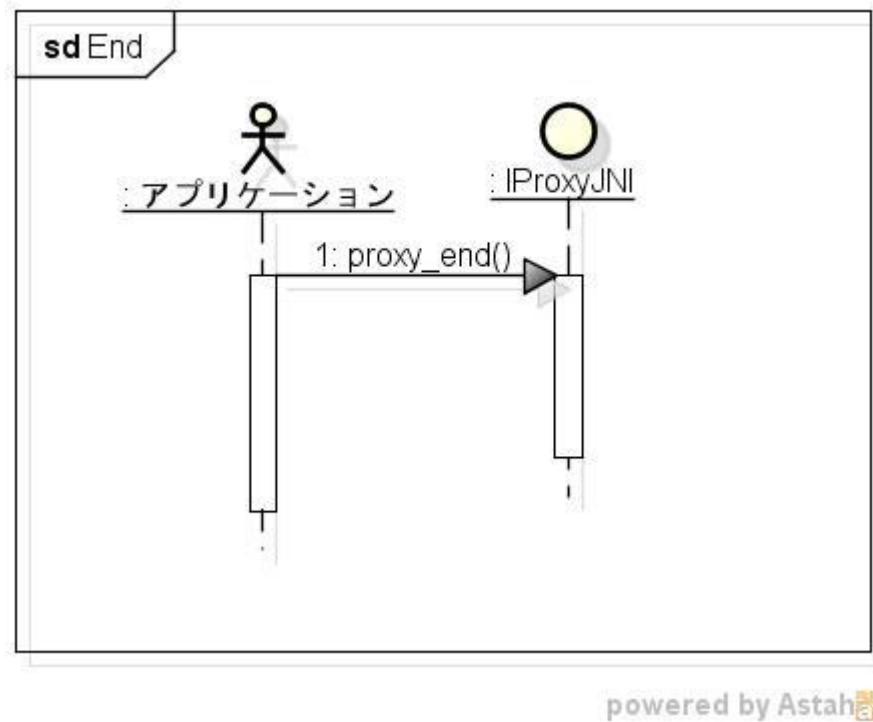
③ クラスのインスタンスを生成



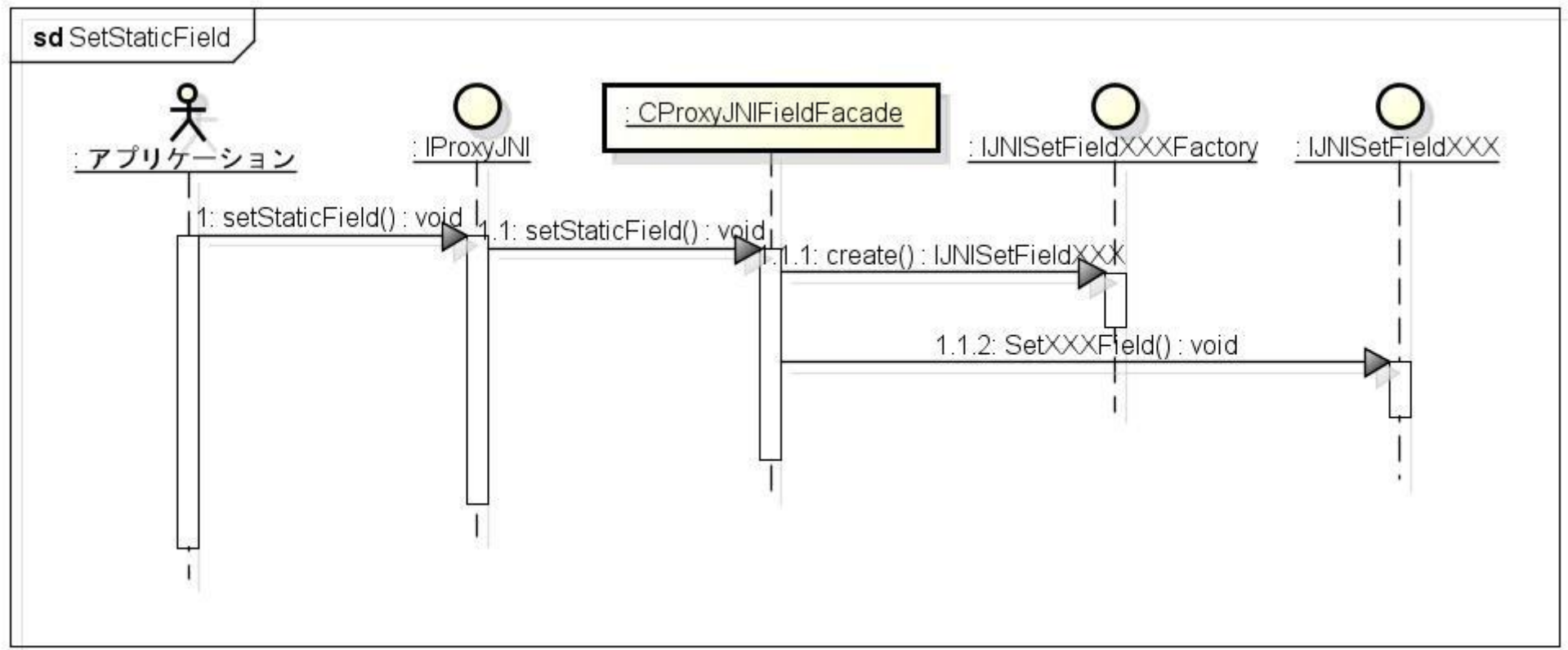
④ インスタンスに対するメソッドを実行



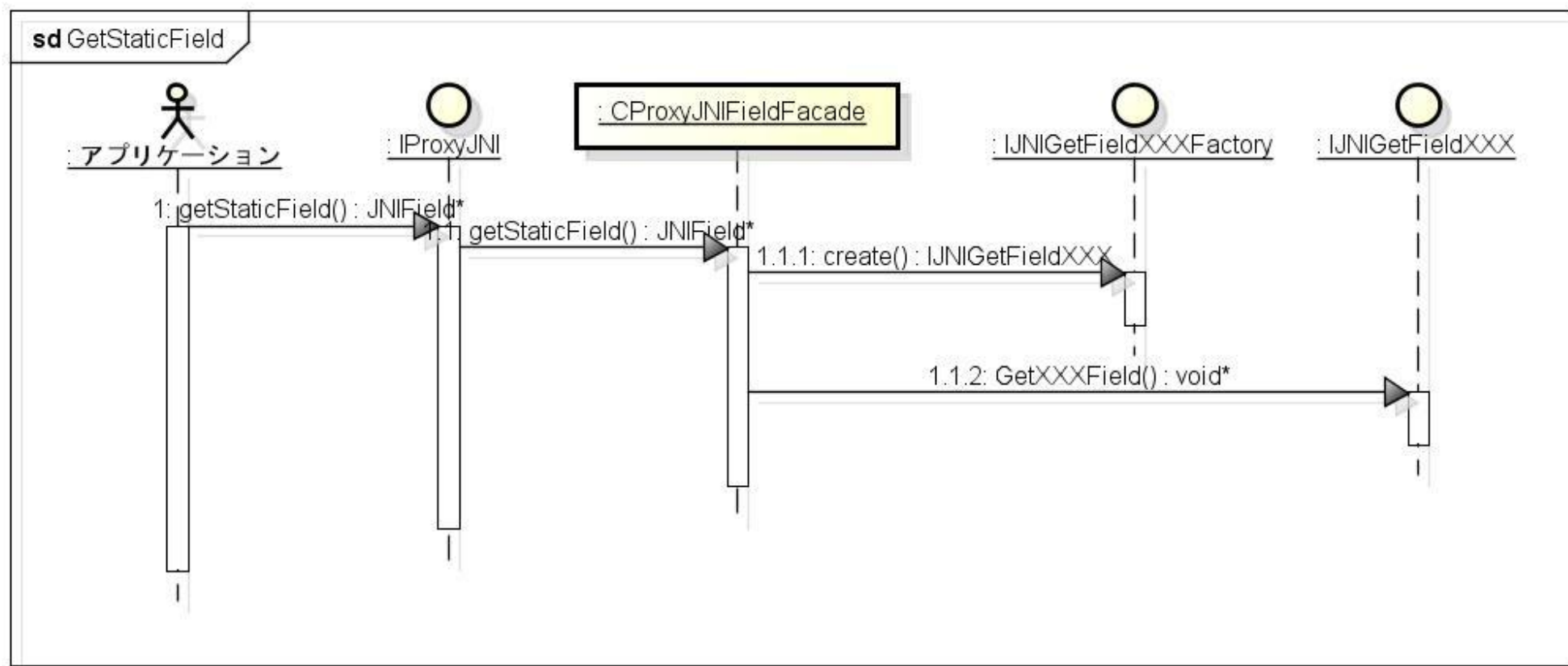
⑤ JNI の終了



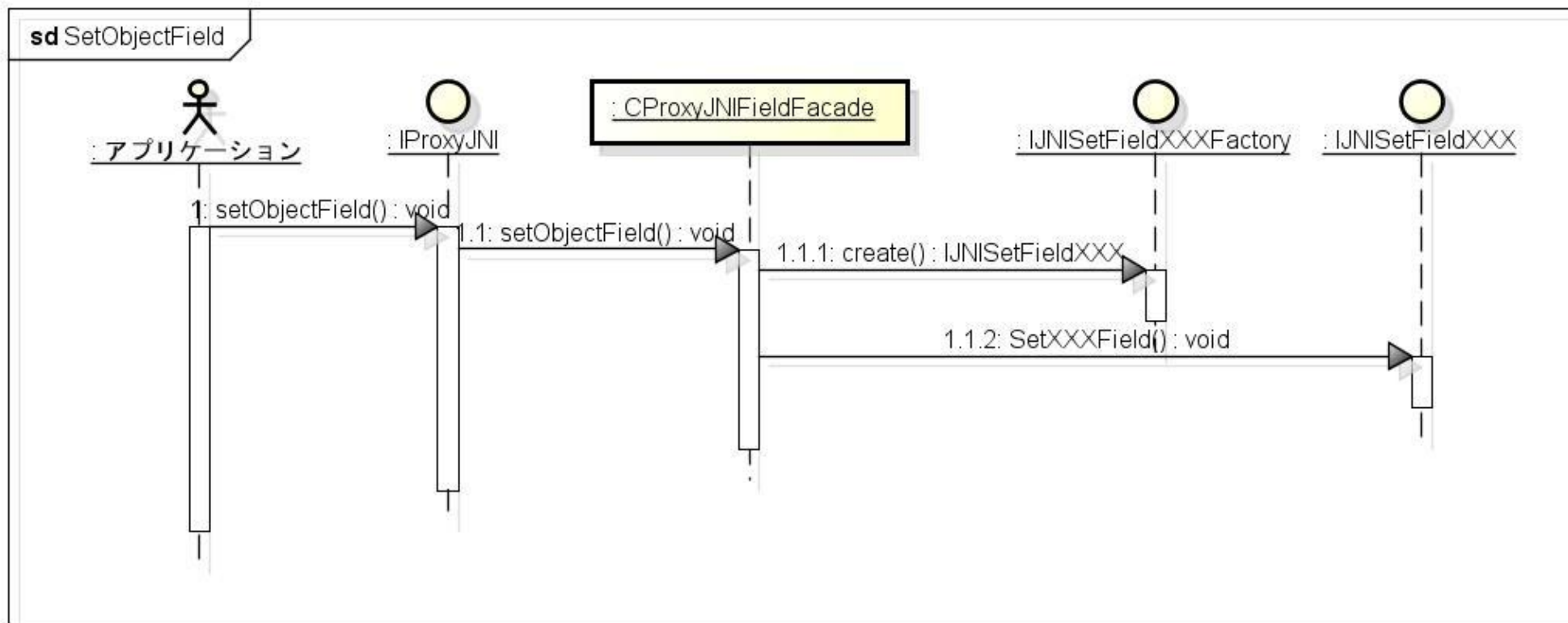
⑥ スタティックフィールド値の設定



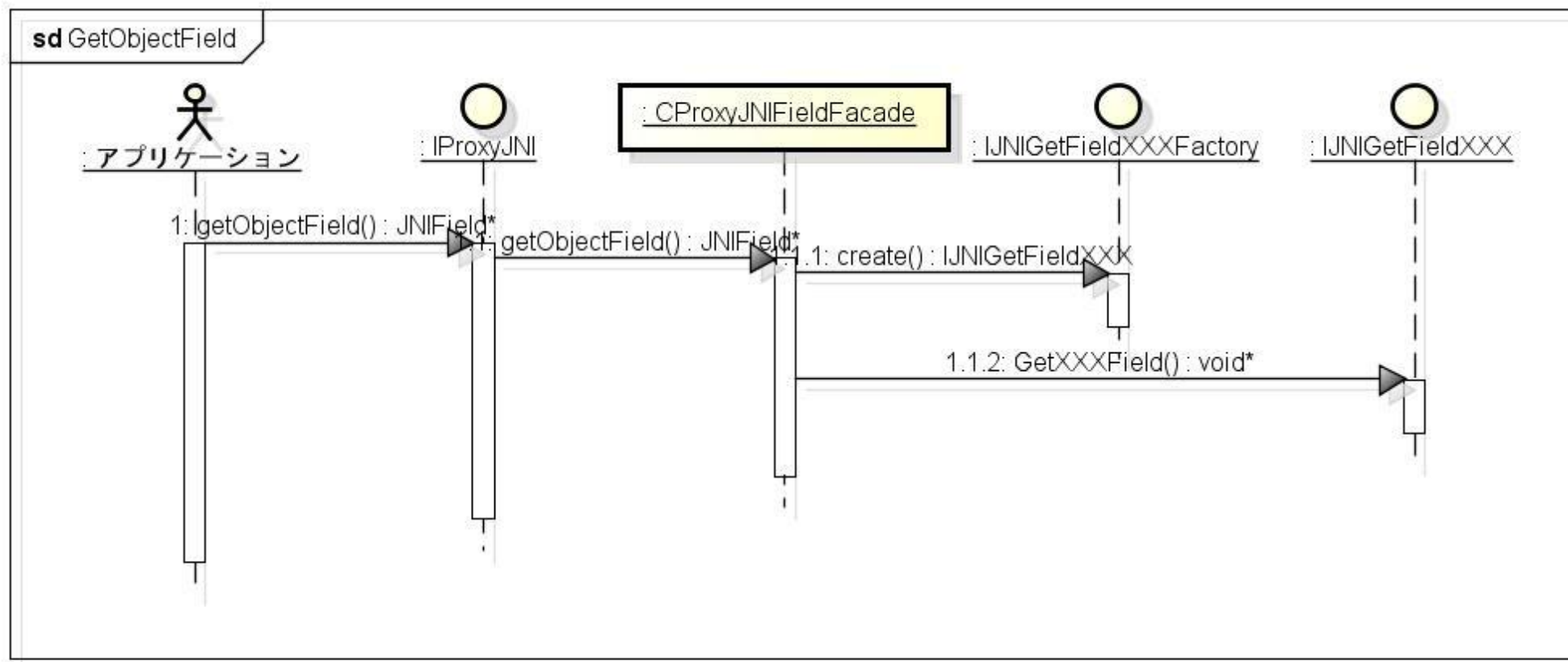
⑦ スタティックフィールド値の取得



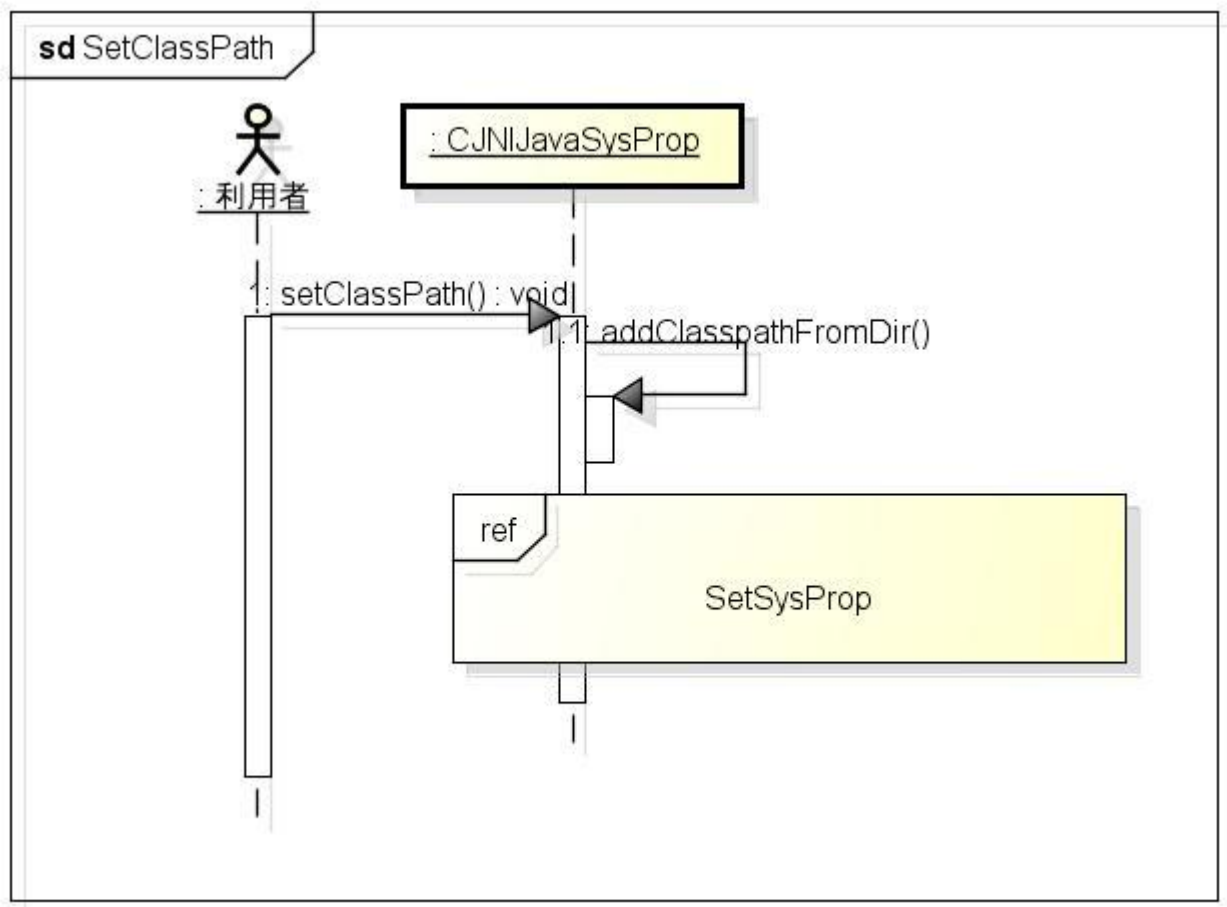
⑧ オブジェクトフィールド値の設定



⑨ オブジェクトフィールド値の取得

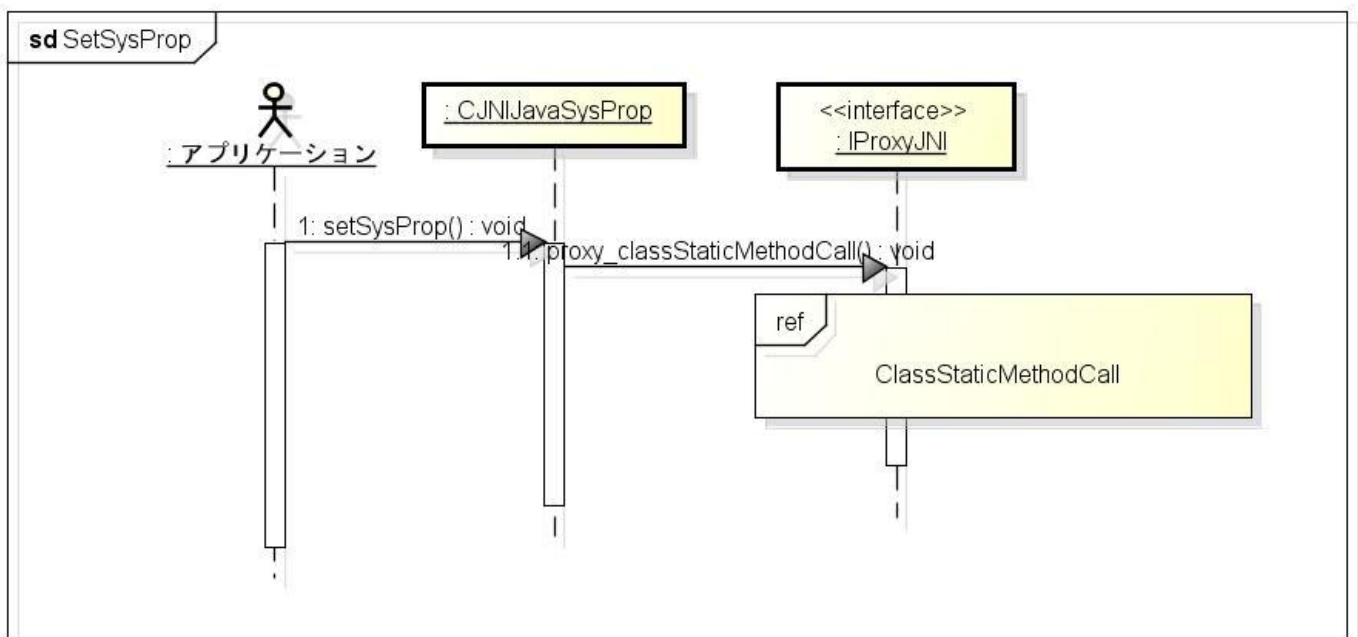


⑩ ClasPath 設定



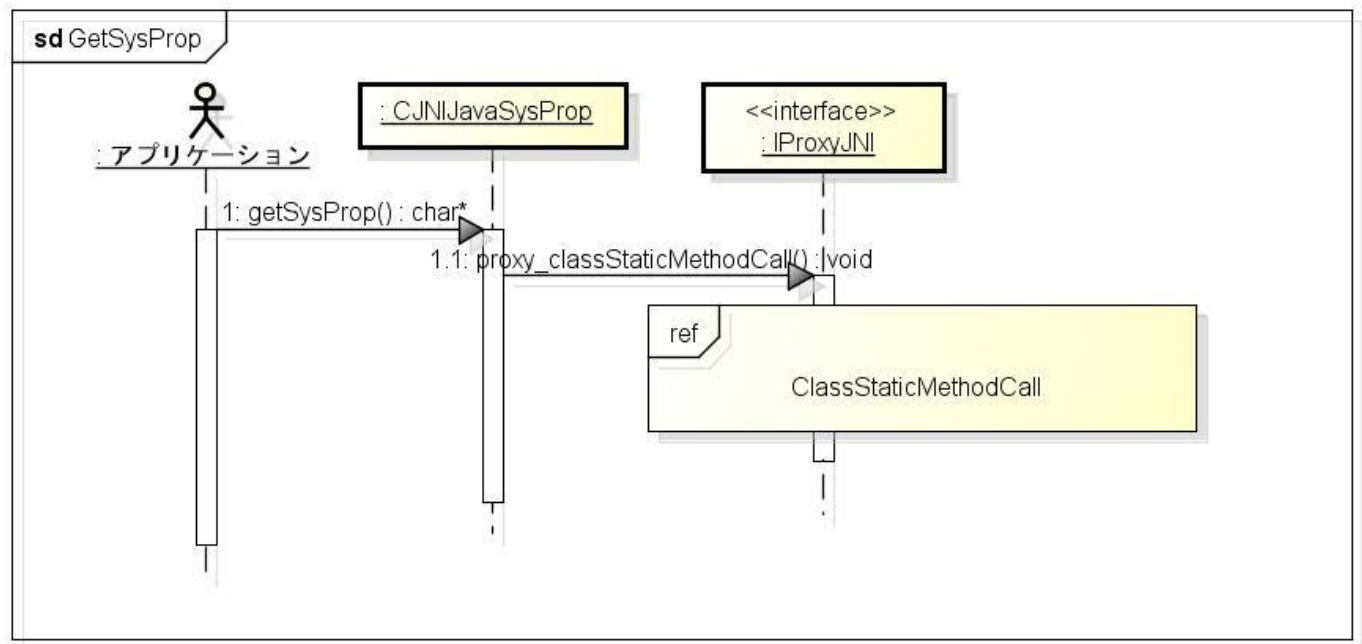
powered by Astah

⑪ System プロパティ設定



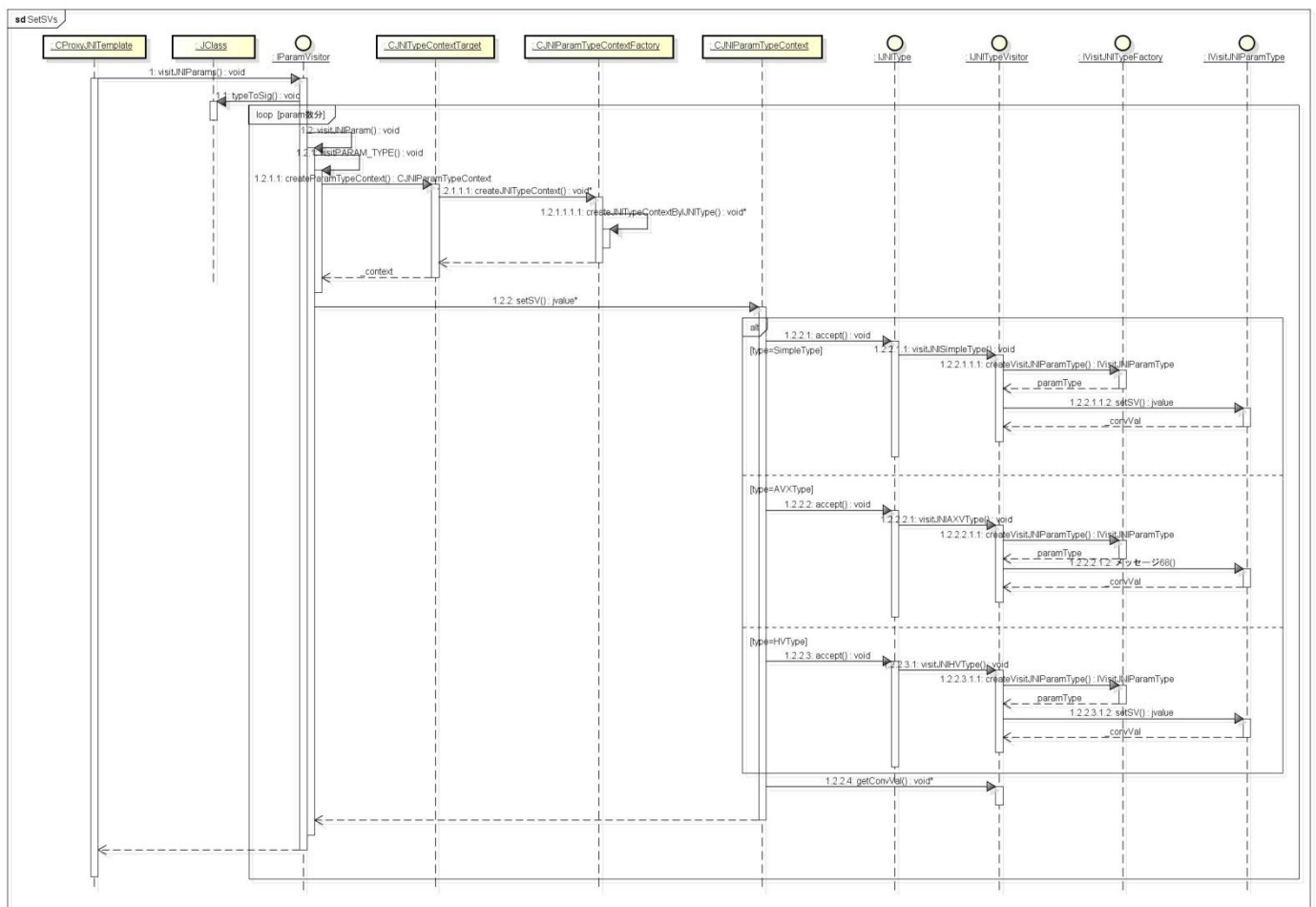
powered by Astah

⑫ System プロパティ取得



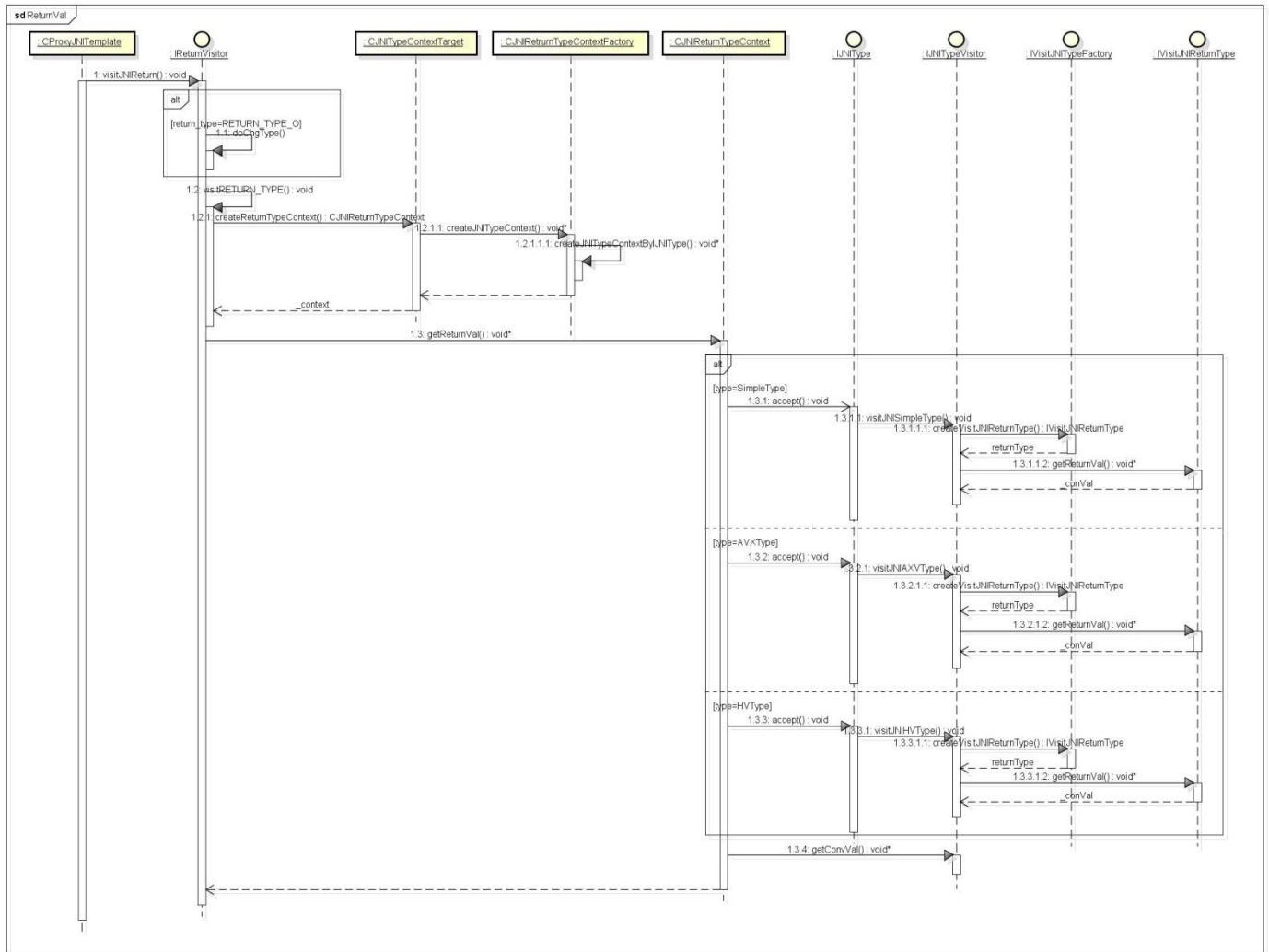
powered by Astah

⑬ パラメータ設定



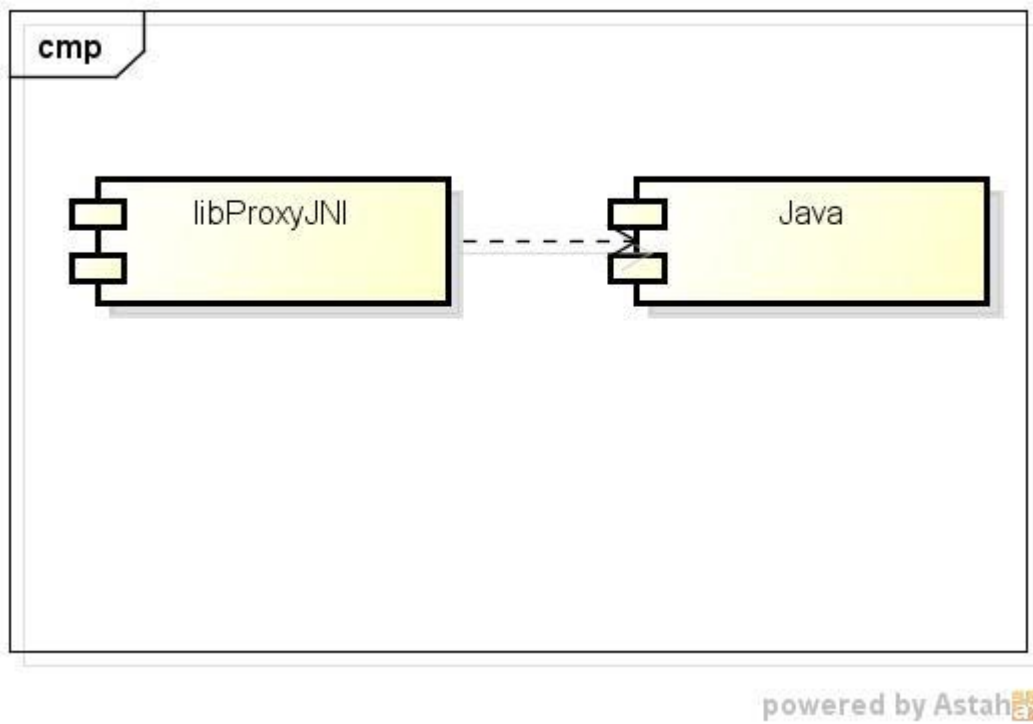
powered by Astah

⑭ 戻り値取得



5. 他のソフトとの対比

- libproxyjni



- swig

