

プログラム解析

平成26年度シラバス

2014年1月10日

国立情報学研究所

トップエスイープロジェクト

代表者 本位田 真一

1. 科目名

プログラム解析

2. 担当者

橋本 祐介

3. 本科目の目的

本科目は、形式手法によるプログラム検証手法を実際のソフトウェア開発における開発効率化・品質向上に活かすことのできる人材の養成を目的としている。具体的には、JMLを用いてJavaプログラムの形式仕様を記述できるようにするとともに、JMLを応用した検証/テストツールを現場で適用するノウハウを身に付け、現実の開発でJMLを活用する上での諸課題についての考察を深めさせる。

4. 本科目のオリジナリティ

Java 言語による実践的なソフトウェア開発への適用を意図して形式手法を学ぶ講座・講義はこれまでなかった。本科目では、Java プログラム開発にそのまま適用可能な形式仕様記述言語として JML を取り上げ、それを扱うための具体的なツールについて解説すると共に、ソフトウェア開発の効率化・高品質化という観点からそれらのツールを使いこなすためのノウハウについて解説する。表 1 に、既存の講座・講義の問題点と、本科目における解を示す。

表 1 既存の講座・講義の問題点と、本科目における解

既存の講座・講義の問題点	本科目における解
形式手法について実際のソフトウェア開発に適用することを考慮していなかった。	Java プログラム開発に利用可能な形式仕様記述言語として JML をとりあげ、それを用いた静的検証と単体テストの方法論を具体的なツールに即して解説する。

5. 本科目で扱う難しさ

近年、ソフトウェア開発における生産性向上・品質向上の要求水準がよりいっそう高まっている。その一つの解決策として、形式手法によるプログラム解析技術の研究が行われてきている。そして、いくつかの手法については実際のソフトウェア開発への適用もすすめられてきている。しかし、それらの方法を使いこなすには高度な技能が必要とされ、限られた応用例しかなかった。

そういった状況を打開するために、一般的なプログラミング言語である Java を対象とした形式仕様記述言語 JML が開発され、JML を利用してプログラム解析を行うツールが開発されつつある。しかし、これらはまだ開発途上であるため、一般ユーザ向けの解説資料や、それらを一般的なソフトウェア開発に適用する上での方法論・ノウハウの整備が進んでいない。つまり、これから発展の可能性のある技術やツールを専門家以外が理解できる形に整理して伝えることが、本科目に求められることでもあり、最も難しいところでもある。

6. 本科目で習得する技術

ソフトウェアの生産性、品質に関して、如何にこれを向上させるかという議論がなされて久しい。IT システムはビジネスの隅々で使用されるようになり、システムの停止の及ぼす影響は企業の存続に影響を与えることさえ珍しくない。また、社会インフラとなっているシステムの停止は、人々の社会生活に多大な影響を与えることとなる。近年は組込みシステムにおいても、膨大なソフトウェアが内部に組込まれるようになり、高生産性、高品質の確保はソフトウェア業界にとって至上命題である。今日さまざまな機器にソフトウェアが組み込まれ、ともすれば人命に関わるような事故も現実におこっている。

ソフトウェアの重要性、規模がますます大きくなると予想される中、生産性、高品質を維持・向上させるためには、これまでの設計、製造、テストという一般的な開発プロセス・手法だけでは限界があるように思われる。本科目は、このような状況の中、形式手法をプログラム開発に適用するためのガイドを提供しようとするものである。形式手法の研究の歴史は長いが、これまで本格的に実際のシステム開発に適用されることは少なかった(航空機産業、防衛産業等極めて高い信頼性が要求される分野での適用はあった)。しかしながら、今後は、形式手法が一般的なソフトウェア開発に使えるか使えないかではなく、使える部分から積極的に活用していかなければならない時代に入っているように思われる。

本科目では、形式仕様記述の手段として JML (Java Modeling Language) を取りあげ、形式仕様記述に基づく静的検証や単体テストの手法を修得させる。これらの手法を実現するツールの実践的な利用方法について詳しく指導するとともに、背景にある考え方の理解、さらに、既存の開発プロセスへの適用における課題の考察を行う。

7. 前提知識

本科目の受講生は、以下の項目を習得済みであることが望ましい。

- Java プログラミング（必須）
- テスト手法（推奨）
- 形式仕様記述（推奨）

とくに実習に際しては、Java プログラムの読み書きやコンパイル・実行、開発環境 Eclipse におけるプロジェクトやプラグインの管理、JUnit、ANT についての基本的な知識・技能が必須である。

8. 講義計画

・ 概要

第1回：形式手法によるプログラム解析

第2回：表明による形式仕様記述

第3回：表明に基づく静的検証（1）－ 基本編－

第4回：表明に基づく静的検証（2）－ 応用編－

第5回：プログラム解析実習（2）－ 静的検証－

第6回：形式手法によるプログラム解析技術の動向

第7回：表明に基づくテスト（1）－テストプログラム作成の自動化－

第8回：表明に基づくテスト（1）－テストプログラム作成の自動化－（続き）

第9回：表明に基づくテスト（2）－網羅性／効率性確保－

第10回：表明に基づくテスト（2）－網羅性／効率性確保－（続き）

第11回：プログラム解析実習（1）－自動テストによる検証－

第12回：プログラム解析実習（1）－自動テストによる検証－（続き）

第13回：プログラム開発への応用（1）－実システム開発－

第14回：プログラム開発への応用（1）－実システム開発－（続き）

第15回：プログラム解析と開発プロセス（グループ討議）

・ 詳細

第1回：形式手法によるプログラム解析

- JML の概要、開発プロセスにおける位置づけについて解説するとともに、主要な JML 応用ツールを紹介する。また、各自の PC におけるツール利用環境を整備する。

第2回：表明による形式仕様記述

- JML を利用したプログラム開発の考え方を簡単に説明するとともに、その利用局面、記述すべき内容、ツールを利用した具体的な記述方法について解説する。

第3回：表明に基づく静的検証（1）－ 基本編－

- JML をプログラムの静的検証に利用する方法について解説する。

第4回：表明に基づく静的検証（2）－ 応用編－

- 定理証明技術に基づく静的検証ツール ESC/Java2 の仕組みと特性の概要を説明し、ツール利用に際しての制限事項やその回避のノウハウについて解説する。

第5回：プログラム解析実習（2）－ 静的検証－

- 具体的なプログラム例を用いて ESC/Java2 による静的検証の実習を行う。

第6回：形式手法によるプログラム解析技術の動向

- 形式手法によるプログラム解析手法全般についての、現状と今後の動向について説明する。現時点で開発途上のツールを含む、さまざまなツールの機能につ

いて概要を紹介する。

第 7～8 回：表明に基づくテスト（1）－テストプログラム作成の自動化－

- JML による表明をテストの自動化に応用する方法について解説する。

第 9～10 回：表明に基づくテスト（2）－網羅性／効率性確保－

- JML をテスト自動化に応用するに際して、網羅性や効率を改善する方法について解説する。

第 11～12：プログラム解析実習（1）－自動テストによる検証－

- 第 7 回から第 10 回までで解説した内容に関し、プログラム開発への適用方法を事例に基づいて解説する。

第 13～14 回：プログラム開発への応用（1）－実システム開発－

- 第 9 回までに修得した知識をオンラインショッピングシステムの検証、単体テストへ適用する実習を行う。

第 15 回：プログラム解析と開発プロセス（グループ討議）

- 形式手法を応用した開発プロセスのあり方について考察すべき課題を提示し、討議、発表を行う過程を通じて、これまでの学習内容への理解を深める。

9. 教育効果

本科目を受講することにより、JML による形式的仕様の記述方法について理解できるとともに、JML を応用する各種ツールを活用してソフトウェア開発を効率化する技能を習得することができる。

10. 使用ツール

Java7 上で動作する JML 対応ツールは未だ品質が高くないため、**本講義では Java5,6 で動作するツールを使用する。**

ESC/Java2 : JML に基づいて静的検証を行うツール。Java 5 でのみ動作する。

- ・ 使用する上での難しさ
 - ツールの完成度が十分でない
- ・ 使用上必要なノウハウ
 - ESC/Java2 の機能制限、不具合を回避するためのノウハウ
- ・ 選択理由、実用性 : 完成度は十分ではないが、JML に基づく静的検証ツールのうちで最も機能が充実している。

JMLUnit : JML による形式仕様記述に基づいて単体テストを行うツール。Java 6 で動作する。

- ・ 使用する上での難しさ
 - JML による形式仕様記述に熟練している必要がある
- ・ 使用上必要なノウハウ
 - 単体テスト設計技法
- ・ 選択理由、実用性 : JML による単体テストツールのうちで最もよく使われている

Mobius PVE : Eclipse 3.4 (Ganymede)ベースのプログラム検証環境。ESC/Java2 と JMLUnit をプラグインとして組み込める。ESC/Java2 と JMLUnit を使う場合には Mobius の起動時にそれぞれ Java 5 と Java 6 の JVM を指定する。

1 1. 実験及び演習

第 1～10 回は講義を中心としているが、講義の途中にはツールの使用方法を習得するための演習問題・実習を配している。

第 11～12 回ではある程度複雑なプログラムを題材として使用し、JML・ツールの適用方法を習得するための実習を行う。

第 13～14 回では、本格的なアプリケーションを題材として形式手法を適用するための開発プロセスの実践について実習を行う。

第 15 回では形式手法を現実のソフトウェア開発に適用する上で生じる様々な課題についてグループ単位で討議を行い、その結果をレポートとして提出させる。

12. 評価

実習課題への答案（第8回）、グループ討議結果レポートを総合して評価する。

1 3. 教科書/参考書

JML、ESC/Java2、jmltool についての適切な参考書はまだない。以下には、本科目のテキストの内容をより深く理解するうえで有用な参考書を示す。

- バートランド・メイヤー, オブジェクト指向入門 第2版 原則・コンセプト、2007
※本書は、契約に基づくプログラミングの考え方を学ぼううえで有用である
 - ✧ pp429-525(第11章:契約による設計:信頼性の高いソフトウェアを構築する)
 - ✧ pp527-586(第12章:契約が破られるとき:例外処理)
 - ✧ pp728-741(第16章1節:継承と表明)
- ロジャーS. プレスマン, “実践ソフトウェアエンジニアリング”, 2005
※本書は、開発プロセスについての基礎知識を学ぼううえで有用である
 - ✧ pp17-34(第2章:プロセス・概要)
 - ✧ pp35-53(第3章:規範的なプロセスモデル)
 - ✧ pp55-72(第4章:アジャイル開発)
 - ✧ pp251-274(第13章:ソフトウェアテスト戦略)
 - ✧ pp275-306(第14章:ソフトウェアテスト技術)
 - ✧ pp565-585(第28章:フォーマルメソッド)
 - ✧ pp587-600(第29章:クリーンルーム開発)
- Watts S.Humphrey, “パーソナルソフトウェアプロセス技法”
※本書には、開発プロセスに関して参考になる記述が含まれる
 - ✧ pp277-296(第13章:ソフトウェアプロセス定義)
- Tim Koomen, Martin Pol, “テストプロセス改善”,
※本書は、本科目でとりあげるテスト手法の背景を理解するうえで有用である
- Kent Beck, “テスト駆動開発入門”
※本書は、本科目でとりあげるテスト手法の背景を理解するうえで有用である
- Tom Demarco, Timothy Lister, “ソフトウェアエンジニアリング論文集 80's”
※本書は、本科目テキスト第4章の背景を理解するうえで有用である
 - ✧ pp057-099(Barry W. Boehm, Phillip N. papaccio, ソフトウェアコストの理解および制御)
- 玉井哲雄, “ソフトウェア工学の基礎 pp195-231(第12章:検証技術)”
※本書は、本科目で学ぶテストや静的検証技術の位置づけを把握するうえで有用である
- Boris Beizer, “ソフトウェアテスト技法”
※本書は、テストで検出すべきバグについて考察するうえで参考になる記述が含まれている
 - ✧ pp28-48(第2章:バグの分類)

- ☆ pp381-399(付録:バグの統計と分類)
- Albert Endres,Dieter Rombach, “ソフトウェア工学・システム工学ハンドブック
エンピリカルアプローチによる法則とその理論
※本書は、本科目で学ぶテストや静的検証技術を実際のソフトウェア開発に適用
する各種のアプローチについての考察が含まれる
- ☆ pp151-221(第 5 章:妥当性検査と静的検証, 第 6 章:テストと動的検証)