トップエスイー修了制作

プログラムスライシング技術を応用したテストメソッド概要抽出

株式会社富士通研究所　上村　学　kamimura.manabu@jp.fujitsu.com

開発における問題点

【目的】
保守のために個々のテストメソッドを理解したい
【課題】テストメソッドの概要を示す情報がない
- 良い名前がついているとは限らない
- 中身を見なければ理解が出来ない
- 中身が長い上に特徴となる構造がない
  （構造：条件文や繰り返し文などの特徴）

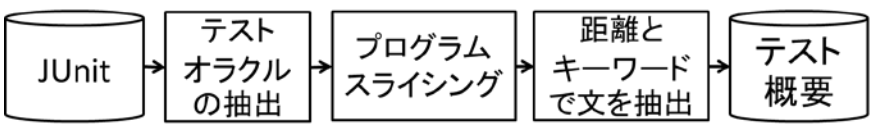
手法・ツールの提案による解決

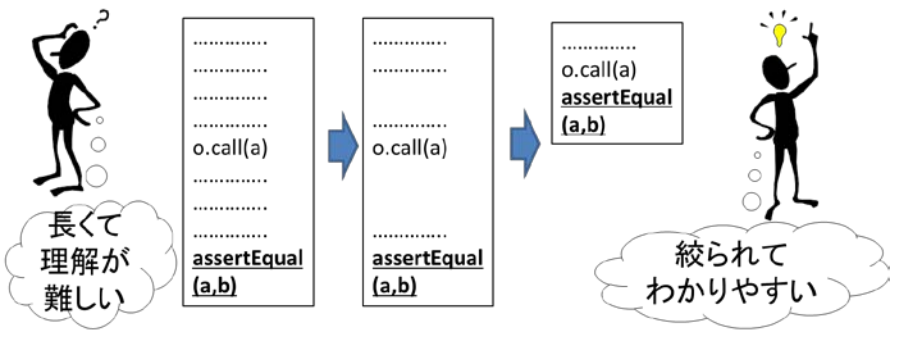
テストメソッドに特化した特徴化の方法を考案し
長いテストメソッドの特徴を示す概要を生成
- 期待値と実行結果を評価するテストオラクル(assert文)に注目
- テストオラクルに関係する文を取得
- 連続して同じ変数へ行われる処理を省略

OSS資産（5本）に適用して実験し概要を評価

手法の概要

JUnitテストコードを入力としてテスト概要を自動生成





評価結果

概要の短縮率
- 平均して元のテストメソッドの約半分の記述量に短縮

評価対象	全体行数	概要の行数	短縮率（%）
5本の平均	99	67.8	37.2～93.4

被験者実験
- 理解支援に向けて更なる検討が必要

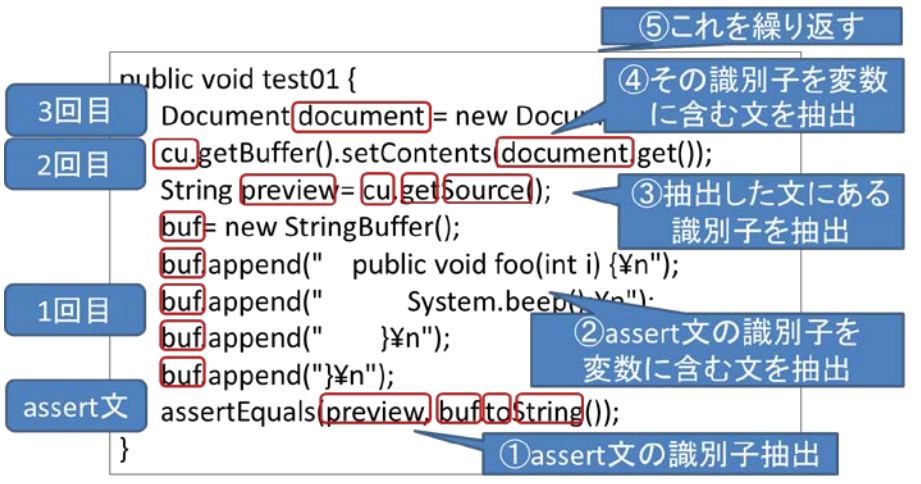
グループ1（4名）	1回目（全体）		2回目（概要）		時間の差
	正答率	時間	正答率	時間	
平均	90%	6分42秒	70%	2分55秒	3分47秒

グループ2（4名）	1回目（概要）		2回目（全体）		時間の差
	正答率	時間	正答率	時間	
平均	65%	4分44秒	70%	2分20秒	2分24秒

アプローチ

1. テストオラクルであるassert文を抽出
- 意図した値が得られているかを確認する箇所

2. assert文を基準にプログラムスライシング



3. assert文からの距離（回数）とキーワード（識別子情報）から文を絞り込み
- 距離（例：1回目）とキーワード（例：buf）が連続して同じ箇所は，最後の文を残す

概要の生成例

```
Document document = new Document("Testfile");
cu.getBuffer().setContents(document.get());
String preview= cu.getSource();
buf.append("}¥n");
assertEquals(preview, buf.toString());
```