

Java プログラミング初学者に対する テスト方法の対話的な学習支援

上河内 頌之[†] 松浦 佐江子[‡]

芝浦工業大学大学院 工学研究科電気電子情報工学専攻[†]

芝浦工業大学 システム工学部電子情報システム学科[‡]

1. はじめに

我々は Java プログラミング初学者(以下初学者とする)に対するテスト学習支援の研究を行っている[1]。[1]では、初学者のプログラムを静的解析し、ホワイトボックステスト技法[2]を用いてのテスト方法の学習支援を行なっている。しかし、入力情報がプログラムだけとなっているため、定義内容に対してのみテストデータが生成され、目的であるプログラムの正しさに関する学習が出来ない。本稿では Java プログラムを対象にテストプロセスを「1. テストメソッドの決定、2. それに対するテストデータ作成、3. テストプログラム作成と実行」の3ステップで定義し、初学者の作成したプログラムと、教員が用意するプログラムの仕様情報を用い、初学者が対話的にテストを行いながらテスト方法を学習するテスト学習支援ツールを提案する。

2. テスト学習支援

テスト学習支援の各ステップにおける支援内容を学部2年生のプログラミング演習課題を例に説明する。

2.1 プログラムの静的解析情報の生成

この課題は継承とインタフェースの概念を学ぶもので、Rectangle クラスは属性に幅と高さを持つ長方形を表すクラスであり、PlaceRectangle クラスはそれを継承し、位置情報を加えたものである。これらのクラスを継承しインタフェース Color を実装し、String 型の色の属性値を加えた ColoredPlaceRectangle クラスを作成する事を目的とする。また、インタフェース Color は DEFAULT_COLOR に String の値を設定し、setColor、mergeColor、eraseColor メソッドを宣言している。そして課題では一つのテストケースを与えている。本支援では、定義済みのクラスのメソッドも含め、メソッド全てに対してテストを行う。

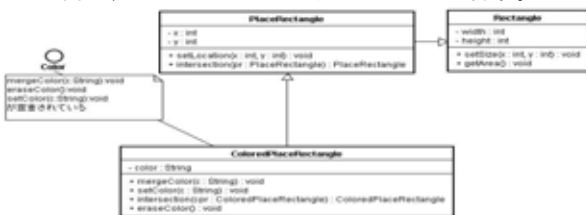


図1. 課題のクラス図

本稿での支援を行う上で、入力対象となるプログラムから各クラスの全てのメソッド(コンストラクタ含む)に対して表1に示した情報を生成する必要がある。なお、記録する対象範囲は全て一つのメソッド内である。

表1の情報Aをテスト順序決定支援に、B~Eをテストデータ作成支援、テストプログラム作成支援に用いる。

表1. プログラムから生成する情報

| |
|---|
| A:他メソッドの呼び出し情報 |
| 他メソッド呼び出しの個数と名前を記録。 |
| B:条件分岐数 |
| if 文と switch 文の個数と判別式を記録。if-else 対応は1つとしてカウント。 |
| C:フィールド参照情報 |
| 参照しているインスタンスフィールド数と名前を記録。 |
| D:引数参照情報 |
| 引数の個数と名前を記録。引数がクラスの場合は、そのクラスのインスタンスフィールド数をカウント。 |
| E:結果情報(フィールド代入文と return 文) |
| インスタンスフィールドの代入文と return 文を記録。 |

2.2 テスト順序決定支援

まず、表1の情報Aを用い、表2の形式で表を与える。この支援は、エラーの発生箇所を特定しやすくするために、呼び出されているメソッドからテストの順序を決定する。よってメソッド内で呼び出しているメソッドを既にテストしているかを調べるチェック欄を設け、判断材料にする。

表2. 具体例に対するメソッド決定支援

| クラス名 | メソッド名 | 呼び出し回数 | 呼び出しメソッド | 呼び出したメソッドがテスト終了しているか |
|----------------|--------------|--------|-------------------------------------|----------------------|
| Rectangle | setSize | 0 | 無し | --- |
| | getArea | 0 | 無し | --- |
| PlaceRectangle | setLocation | 0 | 無し | --- |
| | intersection | 1 | PlaceRectangle (int, int, int, int) | × |
| Colored | setColor | 0 | 無し | --- |
| | eraseColor | 0 | 無し | --- |
| PlaceRectangle | mergeColor | 0 | 無し | --- |
| | intersection | 4 | (メソッド名が4つ続く方欄参照)※ | ×(こちらも4つ続く) |

例えば最初に setSize メソッドを選んだ場合は、後の支援に進めるが、最初に PlaceRectangle クラスの intersection メソッドを選んだ場合、表2からまだメソッド内部で呼び出している PlaceRectangle コンストラクタが未テストなため不適切な選択と判断する。不適切な選択をした場合、表2の情報から表3のように理由を示す。

表3. 不適切な選択に対する根拠を表示

| クラス名 | 選択メソッド | 不適切である理由 |
|-----------------|--------------|--|
| Place Rectangle | intersection | PlaceRectangle(int, int, int, int)がテストされていない |

このように表示した後、表2に戻り正しくメソッドが選択されるまで繰り返す。今回は色を混ぜた処理を定義した mergeColor メソッドを選択した。

```
public void mergeColor(String c){
    if(this.color == DEFAULT_COLOR){
        this.color = c;
    }else if(c == DEFAULT_COLOR){
        }else if(this.color == "yellow" && c == "blue"
            || this.color == "blue" && c == "yellow"){
                this.color = "green";
            }else{this.color = "white";}
    }
```

An Interactive Learning Method for Testing Java Programs

[†] Kamigochi Nobuyuki ; Shibaura Institute of Technology graduate school engineering research course

[‡]Matsuura Saeko ; Shibaura institute of technology Department of electronic information system

このメソッドは混ぜる色の片方が DEFAULT_COLOR の場合、もう片方の色に設定し、yellow と blue の組み合わせなら green を設定する。それら以外の組み合わせであれば、white を設定する事を仕様としている。

2.3 教員が用意するプログラムの仕様情報

テストデータ作成支援、テストプログラム作成支援ではプログラム以外の情報を用いる。これは「テストケース・期待する結果」の2点より構成され、課題文を分析しメソッドそれぞれについて用意する。これを用いてテストを行う事で十分テストが行われるものとする。また、課題文で与えたテストケースは区別しておく。以下の表4に mergeColor メソッドに対し用意する情報を示す。太線の上にある情報が課題文で与えたものである。

表4. mergeColor メソッドに対し用意する情報

| テストケース | 期待する結果 |
|----------------------------|------------|
| 色の属性値が yellow, 引数 c が blue | green |
| 引数 c が DEFAULT_COLOR | 値は変わらず |
| 色の属性値が DEFAULT_COLOR | 引数 c の値に変化 |
| 色の属性値が blue, 引数 c が yellow | green |
| これら以外の組み合わせ | white |

2.4 テストデータ作成支援

ここでは最初に、表5のように対象メソッドに対するテストケースを表示し、選択させる。

表5. テストケース選択

| 課題本文のテストケース | 課題本文以外のテストケース |
|----------------------------|--|
| 色の属性値が yellow, 引数 c が blue | 引数 c が DEFAULT_COLOR 色の属性値が DEFAULT_COLOR ①色の属性値が blue, 引数 c が yellow これら以外の組み合わせ |

を選択したとすると、表6の形でテストデータ作成を指示する。テストケースは表5で選択したテストケースで、項目 color, c とは表1の情報 C, D で記録した名前から得る。

表6. 選択したテストケースのテストデータ作成

| テストケース | color | c |
|-------------------------|-------|------|
| 色の属性値が blue, 引数が yellow | (入力) | (入力) |

今回は color に対し blue, c に対し yellow と入力した。

2.5 テストプログラム作成支援

テストプログラムとは、テストデータを用いて

1. オブジェクトを生成し、それに対して状態を設定
2. メソッド実行
3. 結果確認

の3フェーズで構成される簡易プログラムである。

ここでは図3のようにテストプログラムテンプレートを示し、作成を支援する。テストプログラムのクラス名はテスト対象メソッド名の前に Test を付けたもの、1.と2.のフェーズに該当する箇所では2.4で作られたテストデータと対象メソッドを定義するクラス名から作られるコメントを出力し、3.のフェーズでは assert 文を出力し、/*期待結果*/箇所に2.3の期待する結果情報を入力する。

```
public class TestMergeColor{
    public static void main(String[] args){
        //ColoredPlaceRectangle オブジェクト cpr を作成
        //color の値を"blue"に設定
        //mergeColor を"yellow"を引数にして実行
        assert cpr.getColor() == /*期待結果*/: "期待結果:green ではない。¥n テストデータ:color=blue,c=yellow,実行結果="+cpr.getColor();
        System.out.println("期待結果と実行結果が一致しました");
    }
}
```

図3. テストプログラムテンプレート(mergeColor)

コメントに従い入力が済んだテストプログラムを実行しエラーが発生しなければ、そのテストケースに対してテストが行われた事になる。さらにテストプログラムが正常に動作を終了すれば、図4のように、表1のBとEの情報を用いてプログラム構造を明示したものを表示し、そのテストデータがどの分岐式に入るかを示す。判別式は六角形、代入されるフィールドは四角、代入値と return 値(変数含む)は丸で示す。また except とは、プログラム上の else に当たるものである。

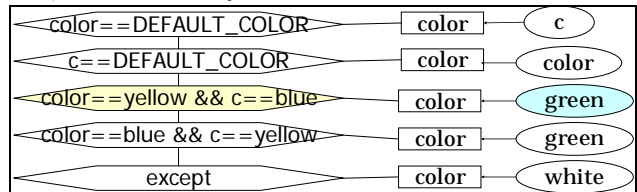


図4. mergeColor メソッドの構造表示

エラーが発生した場合、2.4で作成したテストデータが誤っている可能性があるため、表7のように条件網羅によるパステスト[2]を行った表を示す。結果の欄の"〃"箇所は、それに対応する値が結果となる事を示す。

表7. mergeColor の条件網羅によるテストデータ

| color | c | 結果 |
|-----------------------------|---------------|---------|
| DEFAULT_COLOR | (String 値) | "c" |
| (String 値) | DEFAULT_COLOR | "color" |
| yellow | blue | green |
| blue | yellow | green |
| yellow, blue, DEFAULT_COLOR | (String 値) | white |
| 以外 | | |

これと作成したデータを正しいかを判別し、異なれば作り直させ、テストプログラムが正常動作するまで繰り返す。

3. まとめと今後の課題

プログラミング初学者は「テストをどこからどのようにどの程度行うか分からない」という問題がある。本稿では3つのステップのテストプロセスを繰り返し行うことで、テストの手順とテストの方法を学習する方法を提案した。テストプログラムを定義するためにどの情報に着目して、何を決定すればよいのかを表や図を用いて学習者に示すことで支援を行った。

本支援は課題文において指定したクラスで、定義する全てのメソッドのシグネチャと仕様が定まっている前提で行なった。しかし、そのメソッド内部で仕様に無いクラスやメソッドを作成して定義する場合もある。仕様の無いメソッドに対しては、あらかじめ仕様情報を用意出来ない。しかし、初学者が定義したメソッドは、初学者自身が仕様をそうとは意識せずに定義しているはずである。よってそのメソッドに対し初学者自身で表4の情報を作る事が可能になれば、本支援で提案した流れでテストが行なえるはずである。また、そうする事でメソッドには仕様がある事を認識する事が出来るのではないかと考える。

また、表7を用いてのデータの判定が現時点では初学者の目視によるため、判断が主観的になってしまう。よってデータの正誤性を客観的に定義し、判別を行う必要がある。そして初学者に対し、実際に運用してみて効果を測る必要がある。これらが今後の課題となる。

【参考資料】

- [1] 上河内頌之, 松浦佐江子: Java プログラミング初学者に対するテスト学習支援, 情報処理学会全国大会第67回, 3P-1, 2005
- [2] Cem Kaner, Jack Falk, Hung Quoc Nguyen: 基本から学ぶソフトウェアテスト, 日経 BP 出版センター, 2001