

# Java プログラミング初学者のためのテスト学習支援

## Study-Support for Testing for the first scholar of Programming

上河内 頌之†  
Nobuyuki Kamigochi

松浦佐江子‡  
Saeko Matsuura

### 1. はじめに

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として[4]がある。[4]では、初学者のプログラムを静的解析し、メソッド単体テストにおける順序の決定・データの作成・テストプログラムの提示・実行というサイクルによりテスト方法の学習を行っている。しかし、テストデータの生成に関してプログラムでの定義部分に対するテストデータしか与えられないという問題点がある。本稿では課題を与える者(以下教師とする)が課題に対してのテストケース・テストデータが満たす条件・テストプログラムを用意し、これらを用いてテストデータの作成を支援する事で問題点の解決を図った。テストケースとは何をテストするかを明記した自然言語、テストデータはテストケースを実現するデータ、テストプログラムはプログラムをテストするための簡易プログラムである。本稿で提案する支援により、初学者はテストデータの生成について学習する事が可能となる。

### 2. テスト学習支援

本章では、提案する学習支援について説明する。

最初に提案する学習支援の流れを簡単に説明する。次に、教師が作成する情報について説明を行う。次いでテスト対象メソッドに対してのテスト実行後の支援について説明する。そして示された情報やヒントを用いてテストデータの作成を説明する。最後に学習支援の一連の流れを、具体例を用いて説明する。

#### 2.1 学習支援の流れ

学習支援の流れとしては、以下のようになる。

1. 教師が学習支援に用いる情報を作成。
2. 最初に初学者のプログラムと課題文のテストデータを実行。実行し、テスト完了・未完了箇所の表示・残存テストデータ数の表示を行う。
3. テストを行うメソッドを指定し、そのメソッドのテスト未完了箇所において、教師が作成した情報を用いてテストデータを作成する。
4. 3. で作成したテストデータを用いてテストを実行しテスト完了、未完了箇所の表示を行う。
5. 3.4. を教師が作成した情報に用意されたテストデータがなくなるまで繰り返す。

この流れにおいて学習支援が行なわれるのは 2~5 の段階である。1 の段階は、支援を行う前に教師が支援に用いる情報を用意する作業であり、欠かす事は出来ない。以下の節で 1~3 の段階を説明する。

#### 2.2 教師が作成する情報

本稿では支援を行う前に、教師が本支援に用いる情報(以下ヒントとする)を作成する必要がある。ヒントはメソッドそれぞれに対して、テストケース・テストデータが満たす条件・期待値の 3 点からなるデータ群であるとする。テストケースは課題文を分析する事で作成され、テストケースそれぞれに対しテストデータ・期待値を作

成する。また、テストケースを作成する基準として、教員が課題文と課題文の模範解答プログラムに対し「テスト対象メソッドの条件網羅基準を満たし、限界値分析を用いたもの」とする。条件網羅基準[2][3]とはホワイトボックステスト技法のバステストの基準の一つで「条件分岐の真・偽全ての組合せについて行う基準」である。限界値分析[2][3]とはブラックボックステスト技法の一つで「テストデータを有効同値類・無効同値類に分けこれらの境界値の一つ選ぶ」手法である。作成したヒントの 3 項目をどのように支援に用いるかは次節以降で説明する。

#### 2.3 テストデータを用いたテスト実行後の支援

この支援の機能として

1. テストデータでの対象メソッドの実行箇所の表示

2. 対象の実行すべきテストデータの残り個数の表示

の 2 点を考えた。1. の支援が必要な理由として、初学者はテストを実行してもメソッドのどこを実行しているか分からないと考えたからである。この支援は対象のテストデータがメソッドのどの部分を実行したかを表示する支援を行う。また 2. の支援は、初学者はどの程度数のテストデータを作成すればよいか分からないと考えたからである。この支援はヒントのテストデータの数から残存データ数を計算し、表示する支援を行う。この 2 つの支援を行い、教員がテストを行うメソッドを指定し、未実行箇所についてのテストデータを作成する。

#### 2.4 テストデータ作成支援

この支援の機能として

1. テストデータを作成する際、着目した箇所の表示

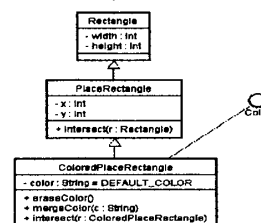
2. そのテストデータに対するテストケースの表示

を考えた。1. の支援が必要な理由として、初学者はテストデータを作成する際に、着目する箇所が分からないと考えたからである。これはヒントのテストデータを用いて着目する箇所を明示し、どこから作られるのかを示す支援を行う。2. の支援が必要な理由は、テストデータを作成しても、初学者はどのようなテストケースか分からないと考えたからである。この支援は、ヒントのテストケースをテストデータ作成時に表示し、テストデータとテストケースを結びつける支援を行う。2.3 の支援と 2.4 の支援を繰り返し行う事でテストデータ作成の学習支援を行う。

#### 2.5 学習支援の具体例

具体例を用いて学習支援を示す。

図 1 に支援対象のクラス図を示す。



Rectangle クラスは長方形を表すクラスである。PlaceRectangle, ColoredPlaceRectangle は、Rectangle クラス、PlaceRectangle クラスをそれぞれ継承したクラスである。Color はインタフェースである。

図 1. 支援対象のクラス図

†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

ここでは、ColoredPlaceRectangle クラスの mergeColor メソッドを対象とする。このメソッドは色を混ぜた時の処理の定義したもので、比較する2つの変数値により異なる値を得るものである。図2に mergeColor メソッドを示す。

```
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";
    }
}
```

図2.mergeColorメソッド

最初に教員がこのメソッドのヒントを作成する。以下の3つが作成されたこのメソッドに対するヒントである。

- Case1:変数の1つが DEFAULT\_COLOR.  
Data1: DEFAULT\_COLOR・任意の文字列  
Return1: 任意の文字列  
Data2: 任意の文字列・DEFAULT\_COLOR  
Return2: 任意の文字列
- Case2:2つの変数が yellow と blue。  
Data1: yellow・blue  
Return1: green  
Data2: blue・yellow  
Return2: green
- Case3:変数の1つが DEFAULT\_COLOR,yellow,blue 以外。  
Data1:DEFAULT\_COLOR,yellow,blue 以外の文字列・任意の文字列  
Return1: white

ここで、Case がテストケースを、Data はテストデータ、Return は分析・設計段階での期待値である。また Data と Return はセットになっており、複数個宣言する事が出来る。メソッドをテストする際、複数の変数が必要な場合、“…”で区切って値を区別する。

これらのヒントを用意して支援を行う。まず、課題文に記述されているテストデータが、Case2 の Data1 を用いたものであったとする。そして、このデータを用いて初学者のプログラム上実行した箇所を表示する。その時、2.3の支援1で述べた問題点から、ここでは条件分岐を図示したものと同プログラム上の実行箇所を同時に表示する。また、実行後の残存テストデータ数も表示する。図3に課題文のテストデータを実行した後に表示するものを示す。

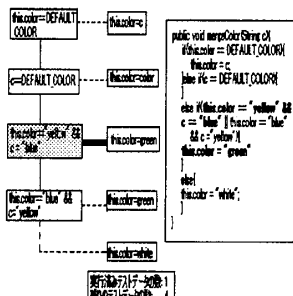


図3.課題文のテスト実行直後

次に、点線で表示されている部分に着目してテストデータの作成を行う。図3を用いてテストデータを作成する箇所を選択する。ここでは例として Case1 の Data1 のデータ

を作成する事にする。まず、図3で対応した箇所を強調して表示し、また左図の点線で示した部分が直線で示される。そしてヒントから、テストケース名として“変数の1つが DEFAULT\_COLOR”と表示する。そして、そのデータはヒントから、DEFAULT\_COLOR と任意の文字列である事が分かる。しかし、このままを出力するのではなく、図3で対応した箇所を明示する事で、データを自ら作成する。ヒントのテストデータの条件と照合し、正しければ作成したテストデータを完成したテストデータとして出力する。ここでは、color に DEFAULT\_COLOR を c に “red” と入力したものとす。入力した値がヒントに用意したテストデータと一致しているかを判定する。そして今回は合っているので、入力した値を完成したテストデータとして出力する。そしてそのテストデータを実行し、図3を更新する。図4にデータの作成を行っている図を、図5に図3を更新した図を示す。図4において、一番上の条件式から返り値まで伸びる線が直線になっている。また、下の部分は作成しようとするテストケースとテストデータを入力させる箇所である。また、コードで対応した部分を斜線で表示する事でコードのどこに対応しているかを示す。また図5においては、図4で作成したテストデータによりテストを実行している。先ほど直線で示した部分が太字になっており実行されている。また下のテストデータの数も変動している。これらを残存データ数が0となるまで繰り返す事でそのメソッドに対し十分なテストデータ作成について学習支援した事になる。

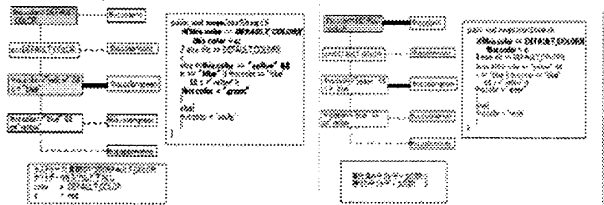


図4.テストデータ作成時

図5.更新した図3

### 3. おわりに

テストには、クラス単体テストや統合テストなどのテストがあるためメソッド単体テストだけでは終了したとは言えない。しかし、今回行ったメソッド単体テストを十分行う事で、クラス単体テストなど、複数のメソッドを組み合わせたテストを行った際にバグが発生した箇所をメソッドの組み合わせの問題と特定出来る事から容易になる。

今回の支援で、メソッド単体テストに関してのテストデータの作成が十分に行われたことになる。そして初学者はプログラムのどこに着目してテストデータを作成すればよいか、またデータを作成する際に、テストケースを表示する事で作成しようとするテストデータがどのようなテストケースかを学習する事が可能になると考える。

今後の課題として、今回はテストを行うメソッドを教員が指定してテストデータを作成したが、本来はメソッドの決定もテスト作業の一部である。そのためテストを行うメソッド順序の決定についての学習支援も必要となる。この仕組みが完成した後ツールを作成し実際に初学者に対して運用する事での評価を考えている。

#### 【参考資料】

- [1] GLENFORD J. MYERS 長尾真 監訳 松尾正信 訳:ソフトウェア・テストの技法,近代科学社,1980
- [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
- [3] Elfriede Dustin, Jeff Rashka, John Paul : 自動ソフトウェアテスト,ピアソン・エデュケーション,2002
- [4] 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,2005