

# プログラミング教育のための発展的教材の作成と考察 - Java 言語によるオセロゲームを題材として -

大庭孝則

神奈川県立相模原総合高等学校

Creation and Consideration of Developmental Teaching Materials for Programming Education  
Reversi using the Java language in the curriculum

Takanori OOBA

## Abstract

小・中・高等学校の学習指導要領が改訂され、それぞれの学校段階においてプログラミング教育が導入されることになった。今後はプログラミングの初歩的段階を終えた学習者に対する、発展的なプログラミング教材の必要性が高まると思われる。この報告では、Java 言語で作成したオセロゲームを教材化する過程を明らかにする。このオセロプログラムは、機能を最小限に抑えて簡潔に記述したが、人とコンピュータが対戦する機能は備えている。授業のまとめでは、学習者がカスタマイズしたプログラム同士を対戦させる「オセロリーグ戦」を実施することを想定している。学習者が愛着をもってコンピュータの思考ルーチンの完成度を高め、オセロゲームの勝敗を競い合うことは、より深化した論理的思考力を身につけることにつながるのではないかと考える。

Key Words: Programming Education, Java language, Making teaching materials, Reversi

## 1. プログラミング教育の気運の高まり

2018年3月に告示された高等学校学習指導要領（以下、新学習指導要領）では、高等学校の共通教科情報科の科目が再編され、現行の高等学校学習指導要領にあった「社会と情報」「情報の科学」に代わって「情報」「情報」が新設<sup>1)</sup>された。

現行の高等学校学習指導要領では、全ての生徒に履修させる情報科の科目を『「社会と情報」及び「情報の科学」のうちから1科目』としていた<sup>2)</sup>が、新学習指導要領では「情報」の1科目が必履修とされている。またこの「情報」には、科目を構成する4項目の一つとして、学習内容にプログラミングを扱った「コンピュータとプログラミング」が明示されている。

このことをプログラミング教育の視点からとらえると、現行の学習指導要領では学習内容にプログラミングが含まれていない「社会と情報」の履修者がおよそ8割<sup>3)</sup>であったのに対して、新学習指導要領では、全ての生徒がプログラミングを学習することになるので、大きな変化であるといえるだろう。

2017年3月に告示された小学校学習指導要領、中学校学習指導要領においても各学校段階でプログラミング教育を充実することとされており、今後の学校教育では小・中・高等学校を通じてプログラミング教育が実施されることになる。

2017年12月26日に神奈川県横浜市の岩崎学園情報科学専門学校で「神奈川県高等学校教科研究会情報部会実践事例報告会2017」が開かれた。大庭も報告者の一人として「発展プログラミングの構想と実践」<sup>4)</sup>というテーマで報告を行った。プログラミング教材として「3目並べ」「棒消しゲーム」を用いたJavaプログラミングの授業例を紹介するものであった。その際、来場者の方から声をかけられ、「今後プログラミング教育の導入が進んで、生徒全員がプログラミングの基礎を学ぶようになると、次にはアドバンスな内容を学習したい生徒に何を提供したらよいか、問題になってきますね。思考ゲームの作成は興味深いテーマですね」との感想をいただいた。論理的思考力を身につけさせるには、ひとつの課題を長時間考え続けることが、時には必要になる。考え続けることが楽しく感じられるような教材開発をしたいという

のが、その時考えたことだった。

## 2. プログラミングによって得られるもの

プログラミングはソフトウェアによってコンピュータを自在にコントロールする技術である。コンピュータは人からの指示、すなわちプログラムを高速に実行するが、与えられた指示だけを忠実に実行するものであるから、プログラム作成者は、コンピュータが処理すべき内容を過不足なく、効率のよい順序で示さなくてはならない。

プログラムを実行させると、コンピュータは開始したとたんに、超高速のドミノ倒しのようにプログラムを処理していく。この瞬間のコンピュータは、別の表現をすれば、プログラムの論理的思考の正誤採点機になる。与えたアルゴリズムが正しければ見事な働きをする反面、誤りがあればさまざまなエラーを引き起こす。

おそらくプログラミングの楽しさは、こうしてコンピュータの正誤判定を受けながらより磨かれたアルゴリズムに仕上げていく過程にあるのだろう。つまりプログラミングの学習は、教科書に示されたサンプルコードを実行するだけでは達成できない。コンピュータを用いて解決すべき何かの課題をもち、自らが組み立てたアルゴリズムによってそれが解決できたとき、はじめて達成できるものと考えられる。

## 3. Java 言語で教材用オセロゲームを作る

オセロゲームは、図 1 のような 8 マス × 8 マスのオセロ盤上に、黒または白の石を打つことによって行われる。自分の石で相手の石をはさむと、はさんだ石を裏返すことができる。オセロ盤には 64 マスあるが、そのうち 4 マスには黒と白の石が 2 つずつあらかじめ配置されているので、実際には 60 マスに黒と白の石を打っていく。相手の石をはさむマスがなければパスになる。すべてのマスに石を打つか、両者とも打つことのできるマスがなくなったところでゲーム終了となる。勝敗はそれぞれの石の数によって決まる。

オセロゲームは「覚えるのに 1 分、極めるのに一生」といわれるように、実際には奥深いゲームなのだが、表面上のルールはシンプルなので、コンピュータのプログラミングの題材としては扱いやすい。

今回教材に用いたオセロゲームは、Java 言語で新たに作成した。Java 言語を選んだのは高等学校の選択授業「プログラミング」で Java 言語が採用された教科書<sup>5)</sup>を使用しているためである。この教科書では、例題プログラムを記述・実行するプログラミング環境として WindowBuilder をインストールした Eclipse を使うように構成されている。そこで選択授業の「プログラミング」を履修した者ならば容易に取組むことができるように、同じプログラミング環境を用いて実行できるようにした。

また、あくまでも教材用のプログラムであるので、簡潔に、わかりやすいアルゴリズムを用いて記述するよう

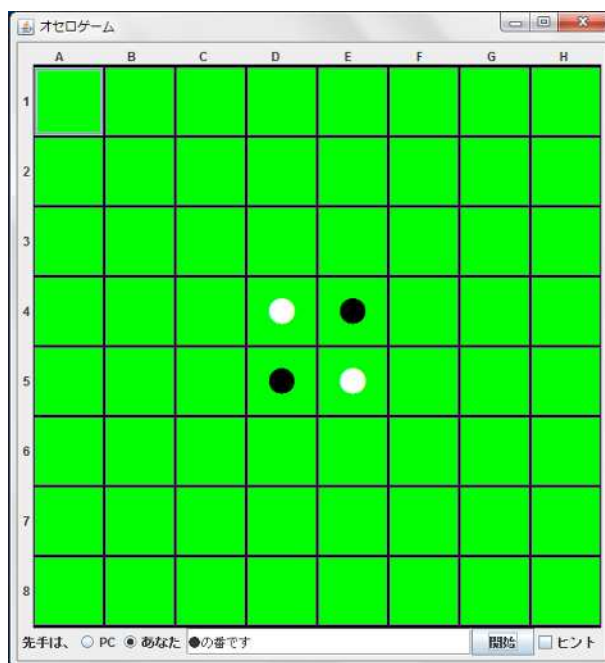


図 1 Java 言語で作成したオセロ盤。

に心がけ、全体のコードがなるべく短くなるようにした。

ただ一か所だけ、教科書が扱う範疇を超える部分があった。それはオセロ盤のマス目がクリックされたときのイベント処理で、前記の教科書では扱われていないマルチスレッドの記述が必要になった。

オセロ盤のマス目がクリックされた後の処理として、

- 1 人が石を打つ
- 2 はさんだ石を一つずつ裏返す
- 3 コンピュータが石を打つ
- 4 はさんだ石を一つずつ裏返す

この4つの動作を順に行わせたいのだが、これを JButton が押されたときに呼び出される actionPerformed メソッドに記述して実行させると、1~4 の動作が省略されて、最後の結果だけがいきなり現れる。つまり人とコンピュータの両者が石を打ち、それぞれがはさんだ石を裏返した、その最後の結果だけが一ぺんに表示されるようになってしまうのである。

原因は Swing の、描画処理とイベントへの応答処理が一つのスレッドで実行されていることであった。actionPerformed メソッドから描画処理を実行しても、イベントキューにその処理が並ぶだけで、実際の描画は actionPerformed メソッドが終了した後に行われる。そこでその対策として、actionPerformed メソッドから JButton のクリックを処理する clickCell メソッドを直接呼び出すのではなく、SwingWorker クラスを用いて別スレッドとして呼び出すことで、期待通りの動作を行わせることができた。

## 4. オセロプログラム教材化のマイルストーン

自作のプログラムをプログラミング教材に用いる場合の手順は次のようなものである。

プログラミングの過程を振り返り、解決が必要だった技術的な課題を列挙する。その中で学習者に教材として学ばせるものを取捨選択する。これまでの学習内容の延長上にある内容は優先的に扱うこととし、逆に高度な技術を要する内容は解説の中で、発展的な内容であることを示すようにする。

このオセロゲームでは表 1 に示した 6 つの指標を設けることで、段階的に学習できるようにした。各ステップでは、学習者のプログラム作成へのモチベーションを維持できるように、実行ファイルを作成してそれぞれの段階でのプログラムの動作を確認できるようにした。

表 1 オセロゲーム教材化の指標

1	オセロ盤の作成
2	打てる場所の確認
3	はさんだ石の反転
4	パスとゲーム終了の判定
5	コンピュータ対戦機能の追加
6	コンピュータ思考ルーチンの強化

ステップ 1 では、オセロ盤の作成を行う。オセロ盤はマス目の数が多いので WindowBuilder の GUI 編集機能を使わずに、for 文を用いたコードを記述してコンポーネントを配置させる。またプログラムに必要な変数を考えるのも、この段階で行うことにした。

ステップ 2 では、自分の石で相手の石をはさむことができる場所を探すアルゴリズムを考える。はさむことができるとは、隣接するマス目には相手の石があり、その先には自分の石があるということである。このことを調べるにはどのようなアルゴリズムを用いるか、がポイントになる。作成したプログラムでは、リスト 1 に示した countReverse1 メソッドを使って、指定したマス目の周

囲 8 方向について、反転可能な石数を調べさせることにした。

ステップ 3 では、はさんだ石を反転させるアルゴリズムを考える。リスト 2 のように、ステップ 2 で作成したメソッドを利用すると簡潔に記述できる。また石を一つずつ裏返すのが分かるようにゆっくり描画させるにはどうしたらよいかも、あわせて考える。

ステップ 4 では、パスとゲーム終了を判定するアルゴリズムを考える。自分の石を打てる場所がなければパスになるが、相手も打つ場所がなければゲーム終了となる。それを判定するにはどのようなアルゴリズムが必要か考える。

ステップ 5 では、人对コンピュータの対戦ができるように、プログラムの変更を行う。作成した各メソッドの役割を再確認しながら、コンピュータの思考ルーチンをどう記述するか、またそれをどこから呼び出せばよいか、考える。ただしここで実装するコンピュータの思考ルーチンは、簡易版である。コンピュータが石を打つ場所は、石を打てるマス目の中から最初に発見したマス目を機械的に選ぶだけの単純なものにしておく。

ステップ 6 では、コンピュータが戦略的に打ってくるように、思考ルーチンを改良する。リスト 3 の comThink メソッド（各マス目の評価に基づいて石を打つ思考ルーチン）を例示するが、さらに強い思考ルーチンを考えるためのヒント、例えばそれぞれのマス目に石を打った場合、いくつの石を裏返すことができるかカウントするメソッド等を示し、学習者の創意工夫でより強いオセロゲームを作ることができるよううながす。

## 5. おわりに（授業実践に向けて）

プログラムが完成してゲームが可能になったら、盤面に表示される「オセロゲーム」という表示を、学習者のそれぞれの名前を使い、例えば「大庭オセロ」のような

```
398 > // 指定位置(x,y)にcol色の石を置いたとき、反転できる石数を指定方向 sx,sy で調べる←
399 > private int countReverse1(int x, int y, int sx, int sy, int col){←
400 >     int count = 0;←
401 ←
402 >     int col_Aite = 1; > > > // 相手の色←
403 >     if(col == col_Aite) col_Aite = 2;←
404 ←
405 >     // 相手の石が何個並んでいるか数える←
406 >     x += sx;←
407 >     y += sy;←
408 >     while(getColor(x, y) == col_Aite){←
409 >         x += sx;←
410 >         y += sy;←
411 >         count++;←
412 >     }←
413 ←
414 >     // 未入力・範囲外等、自分の石以外に到達したらカウントをリセット←
415 >     if(getColor(x, y) != col){←
416 >         count = 0;←
417 >     }←
418 ←
419 >     return count;←
420 > }
```

リスト 1 指定したマス目が打てる場所か、反転できる石数で判定するメソッド。

```

341 > // 1方向に向けて石を反転させる←
342 > private void reversel(int x,int y,int sx,int sy,int col){←
343 >     int c;←
344 >     c = countReversel(x, y, sx, sy, col);←
345 >     while (c>0){←
346 >         waitTime();>>>>>>>>> // 0.3秒待つ←
347 >         x += sx;←
348 >         y += sy;←
349 >         c--;←
350 >         ishi[x][y] = col;←
351 >         setColor(x, y, col);←
352 >     }←
353 > }←

```

リスト 2 指定したマス目から指定方向に向けて石を裏返すメソッド。

表記に改めさせようとしている。

プログラム作成者の名前が表示されることは、教室内でトーナメント戦を行う際に画面の区別がしやすいことが第一の理由だが、もうひとつの理由は、学習者が作成したプログラムに愛着を持てるようにするためである。

通常、学習者は、学習用に示されたソースコードに愛着を持つことはない。教科書に示された例題プログラムは、入力して動作確認ができれば、すぐに次のプログラムに興味がる。しかしコンピュータのアルゴリズムは一通りではなく、同じ処理をさせるには何通りものやり方がある。プログラムを作成するための論理的思考力を鍛えるならば、ここが時間をかけて考えるべきところである。よりよいアルゴリズムを編み出し、採用するアルゴリズムを選択するには、そのプログラムに愛着を持って、組み立てたアルゴリズムを繰り返し吟味するような、題材がふさわしいのではなかろうか。自分の名前を冠したオセロプログラムは、愛着を持ってアルゴリズムを磨き上げるための教材になるのではないかと、期待している。

この自主制作教材を用いた授業は、神奈川県立相模原総合高等学校の学校設定科目「発展プログラミング」で、高校3年生の科目選択者を対象に、2018年度の後期に実施することを計画している。

また授業で使用するプログラム及びテキストは、神奈

川県立相模原総合高等学校のホームページ<sup>6)</sup>からダウンロードできるようにする予定である。

#### 参考文献

- (1) 文部科学省，高等学校学習指導要領(平成 30 年 3 月)  
[http://www.mext.go.jp/a\\_menu/shotou/new-cs/1384661.htm](http://www.mext.go.jp/a_menu/shotou/new-cs/1384661.htm)  
(アクセス日 2018 年 7 月 24 日)
- (2) 文部科学省，高等学校学習指導要領(平成 21 年 3 月)  
[http://www.mext.go.jp/a\\_menu/shotou/new-cs/youryou/1304427.htm](http://www.mext.go.jp/a_menu/shotou/new-cs/youryou/1304427.htm) (アクセス日 2018 年 7 月 24 日)
- (3) 文部科学省，教科書の種類数・点数・需要数（平成 30 年度用）  
[http://www.mext.go.jp/a\\_menu/shotou/kyoukasho/gaiyou/0406090/1235103.htm](http://www.mext.go.jp/a_menu/shotou/kyoukasho/gaiyou/0406090/1235103.htm) (アクセス日 2018 年 7 月 24 日)
- (4) 河合塾「キミのミライ発見」授業事例,【事例 73】発展プログラミングの構想と実践  
<https://www.wakuwaku-catch.net/jirei1873/>  
(アクセス日 2018 年 7 月 24 日)
- (5) 中澤興起ほか 7 名，最新プログラミング オブジェクト指向型言語，実教出版株式会社，2017.
- (6) 神奈川県立相模原総合高等学校ホームページ,  
<http://www.sagamiharasogo-ih.pen-kanagawa.ed.jp/>

```

165 > // コンピュータが考える STEP6 で改訂←
166 > private void comThink(){←
167 >     int c = 0;←
168 >     int x2 = 0;←
169 >     int y2 = 0;←
170 <
171 >     if(setHint((nanTe % 2) + 1)>0){>>> // 石を打てる場所を調べる←
172 >         for(int y=0; y<8; y++){←
173 >             for(int x=0; x<8; x++){←
174 >                 if(hint[x][y] == 1){>>> // 石を打てる場所のうち，←
175 >                     if(c < point(x, y)){>>> // ポイントの高い場所の x,y を←
176 >                         c = point(x, y);>>> // x2,y2 に代入する←
177 >                         x2 = x;←
178 >                         y2 = y;←
179 >                     }←
180 >                 }←
181 >             }←
182 >         }←
183 >         // リバース処理←
184 >         clickCell2( x2, y2 );>>>> // 最高ポイントの場所 x2,y2 に打つ←
185 >     }←
186 > }←

```

リスト 3 それぞれのマス目に対する評価ポイントを使ったコンピュータの思考ルーチン。