

プログラミング学習の他教科への波及効果(2) ～数学と考える力を対象とした調査と検証報告～

井戸坂幸男, 足利裕人, 紅林秀治, 鎌田敏之, 兼宗 進, 久野 靖
松阪市立飯南中学校, 鳥取県立鳥取工業高校, 静岡大学, 愛知教育大学, 一橋大学, 筑波大学

概要

プログラミング学習を行うことで、コンピュータに関する理解が向上することは多く言われている。我々はそれ以外にも、プログラミング学習には「考える力」を伸ばす効果があり、それによって他教科の学習にもよい影響を及ぼすのではないかと考えた。SSS2006では、2005年度にオブジェクト指向言語「ドリトル」を使い実施した、中学校技術・家庭科のプログラミング学習授業およびそれに関するアンケートについて報告した。本稿では、2006年度のプログラミング学習が他教科(とくに数学)の学習にどのような影響を及ぼしているかを検証した結果について報告する。

Effect of Programming Study on Other Subjects(2) ～ classroom survey on logical thinking and mathematics ～

Yukio IDOSAKA, Hiroto ASHIKAGA, Shuji KUREBAYASHI
Toshiyuki KAMADA, Susumu KANEMUNE, Yasushi KUNO
Iinan Junior High School, Tottori Technical High School, Shizuoka University
Aichi University of Education, Hitotsubashi University, University of Tsukuba

Abstract

It is widely accepted that programming lessons lead to better understanding on principles of computer. We have been teaching programming lessons using "Dolittle" language on "Technology and Home Economics" subject of junior-high school for several years. From their experiences, we additionally thought that programming lessons had good effects on other subjects. In our previous classroom surveys (reported on SSS2006), we got many positive answers for the question "Did Dolittle classes have good effects on your mathematics understandings?" Thus, we have conducted more systematic survey using written examination on mathematics and logical thinking. In this presentation, the framework for the survey and its results are reported.

1. はじめに

筆者のひとり、中学校の技術・家庭科において、プログラミング学習を取り入れている。実践を始めてから10年が経過し、長年の実践からプログラミング学習は生徒の思考力や創造力の面でよい影響を受けていることに気づくようになった。そこで、どのような効果があるかを知るために、2005年度より生徒アンケートを実施した。

2005年度のアンケートの結果から、プログラミング学習が図形や関数に関する概念の形成を助ける効果があることがわかった。このアンケート結果とその分析については、[1]で詳しく説明している。そこで2006年度においては、数学における「図形や関数の概念」

を扱う題材のうちから「一次関数」をとりあげ、この内容理解とプログラミング学習との関係について検証していくことを計画した。

2006年度の検証は、2年生の技術・家庭科の時間にプログラミング学習を実施したクラスと他の教材を実施したクラスを用意し、数学「一次関数」の理解にどれくらいの理解度の違いがあるかを調べる。

- ・対象： 中学2年生(A組24名、B組25名)
- ・実施時期：2006年9月～2007年3月
- ・プログラミング時間：12時間(1時間/週)
- ・検証方法：

最初、A組はプログラミング学習を行い、B組は他の教材[Webページの作成]を行う。続いて、A組が他の教材、B組がプログラミ

ング学習を行う。開始前、中間点、終了時点の3時点で「一次関数」の理解判定を含むテストを実施し、両クラスの理解度を比較する。

数学の学習状況との関係は、最初のA組がプログラミング学習を実施している時、数学では、「一次関数」の単元の後半部分を学習していた。検証の流れを図1に示す。

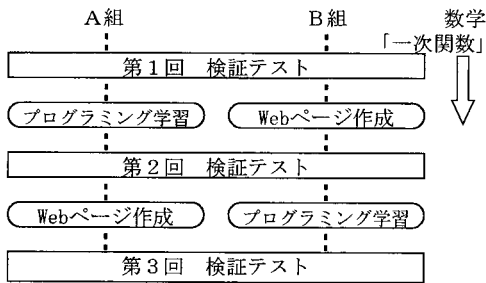


図1：検証の流れ

本稿執筆の段階では、第2回検証テストまでを終了している。第2回検証テストまでの結果について分析する。

2. プログラミング授業の内容

三重県の松阪市立飯南中学校において、プログラミングの授業を12時間(検証テストの時間を含まず)実施した。

言語には教育用プログラミング言語である「ドリトル」[2]を使用した。ドリトルは、日本語で命令ができるなど、初心者にわかりやすく学習できるように配慮されており、また、オブジェクト指向を積極的に取り入れている。

授業は、表1のカリキュラムで実施した。以下でその概要について解説する。

表1 カリキュラム

単元	時間数
(1) タートルグラフィックス	4
(2) タイマーによるアニメーション	4
(3) ボタンによる対話的な操作	4

2.1 タートルグラフィックス

最初の4時間は、ドリトルのタートルグラフィックス機能を用いてさまざまな図形を描きながら、基本操作やプログラムの基本を学習した。

具体的には、前進命令(あるく)と角度を変える命令(みぎまわり)を基本として三角形を描くところから始め、くり返し命令(くりかえす)や、描いた図形の移動(いどうする)などを学習する。図2は例題プログラムのひとつで、繰り返しの使った三角形を描き、できた三角形を図形として色を塗り指定位置に置く。

```

かめた=タートル!つくる。
さんかく=「かめた!100ぼ あるく 120ど
みぎまわり」!3かい くりかえす
ずけいにする (赤)ぬる。
さんかく!100 100 いどうする。
    
```

図2：図形を描き配置するプログラム例

ここでは、「プログラムは順序をもって実行される」こと、「くり返し命令を使うことで簡単に同じことを繰り返せる」こと、「オブジェクトを作り、オブジェクトに命令できる」ことを学習した。また、プログラムの中で数学でも扱う図形(三角形、正方形等)の外角、座標、移動距離、回転角度について扱った。

生徒はこの段階までで、三角形、四角形、星、円、などの図形を組み合わせ、さまざまなグラフィックス作品を制作した。図3に生徒の作品例を示す。

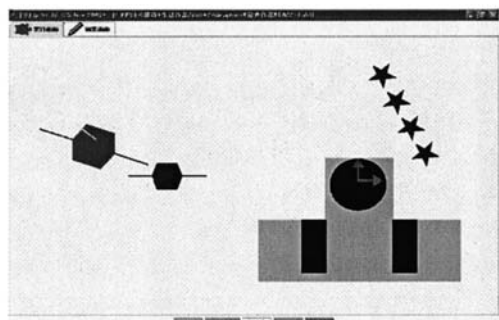


図3：生徒によるグラフィックス作品

2.2 タイマーによるアニメーション

次の4時間では、描いた図形を一定時間移動させたり、回転させるタイマーオブジェクトを学習した。ゲームソフトなどでキャラクターが動くしくみを体感することができる。図4はタイマーを使ったプログラムの例であ

り、時間とともにタートルが回転し、また図形が平行移動するものである。

```

かめた=タートル!つくる。
さんかく=「かめた!100ぼ あるく 120ど
みぎまわり」!3かい くりかえす
ずけいにする(赤)ぬる。
動き=タイマー!つくる。
動き! 1びょう かんかく 10びょう じかん。
動き!「かめた!15ど みぎまわり。
さんかく! -10 -10 いどうする」じっこう。

```

図4：タイマーを使ったプログラム例

この単元では、タイマーによって図形の移動がゆっくりとくり返し実行されることを視覚的にとらえることで、図形の回転移動と平行移動を学習できた。

生徒はこの段階までで、図形が移動するアニメーション作品を制作することができるようになった。図5 に生徒の作品例を示す。

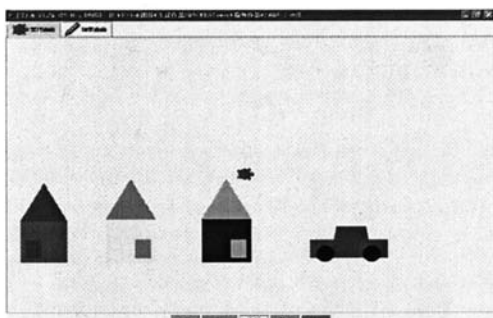


図5：生徒によるアニメーション作品

2. 3 ボタンによる対話的な操作

次の4時間は、クリックすると命令が実行されるボタンオブジェクトを使い、ボタンを押すことで画面に線や図形を描く作品(お絵かきソフト)を制作した。ここでは、メソッドの定義の方法を学習した。また、プログラムの中で直角座標を扱った。図6 にボタンを使ったプログラムの例を、図7 に生徒の作品例を示す。

```

かめた=タートル!つくる。
ボタン1=ボタン!“あるく” つくる。
ボタン1!-200 100 位置 150 50 大きさ。
ボタン1:動作=「かめた! 50ぼ あるく」。

```

図6：ボタンを使ったプログラム例

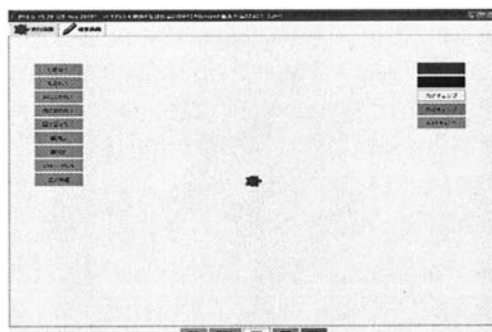


図7：生徒によるボタンを使った作品

2. 4 授業における留意点

このプログラミング学習の授業は、技術・家庭科で実施しているため、ソフトウェアの原理などを理解するための教材として取り入れている。また、扱う題材については「創意工夫する力」をつけるため、グラフィックスを中心に扱い、タートルや図形の動きや配置を工夫したり、プログラムの記述を工夫することに力を入れている。

今回の検証では、他教科への波及ということで、数学における学習効果の検証を行っているが、数学の理解を助けるための補助授業にならないように注意した。生徒アンケートからは、「パソコンの授業が好き」と答えた生徒は93.0%であり、「数学を嫌い」と答えた生徒は67.5%、「数学が不得意」と答えた生徒は69.8%であった。つまり、ほとんどの生徒はパソコンによる授業が好きであるが、数学を不得意と感じたり嫌いと思っている生徒は2/3を超えている。この生徒の中には、数学という言葉だけで拒否反応を示す生徒もいる。数学に偏った授業をするとパソコン嫌が増えることも予想されるため、数学での授業は意識せずに授業を行った。

また、作品においては、プログラムも大切ではあるが、発想・アイデアを重視した作品になるように工夫を促した。このことは、プログラムの思考では優位になれない低学力の生徒も、発想・アイデアでは優位に立つことができるため、すべての生徒が制作意欲をもって取り組める。すべての生徒が、意欲的に取り組み、プログラミングを通して論理的思

考を身につけていくように考えた。

3. 検証テストの問題

グラフィックス中心の授業のため、数学の単元としては、図形、関数への波及効果が期待できる。今回の検証では、関数の分野に絞って問題を作成した。作成にあたっては、数学の先生の要望もふまえ、基礎・基本問題を中心に選んだ。また、数学嫌いの生徒に配慮して、問題数は最小限にとどめた。

3回の検証テストを予定しているが、数学の問題については、数値を入れ替えるだけで同じ問題を使う。論理的思考問題に関しては、他学年で予備調査を実施した上で、同じレベルになるように調整して実施する。

以下に問題を示す。

3. 1 視覚的にとらえる問題

視覚的にとらえる問題として、次の2つの問題を用意した。

まず、図8のような座標上に与えられた点をとる「座標問題」である。

(1) 次のア～エの点を書きなさい。点には、どの問題の点かがわかるように、点のそばにア～エの記号を書いてください。

ア. 点 (5, 3)
 イ. 点 (2, -3)
 ウ. 点 (-3, -5)
 エ. 点 (-4, 4)

図8 : 座標問題

座標については、中学1年の「比例・反比例」の単元で学習する。この問題は、ドリトルのプログラミングではボタンの配置などを座標で指定するため、自然と座標のとらえ方が身につくと思われたため選んだ。

次に、図9のような一次関数の式を与え、グラフを描かせる「グラフ問題」である。

(4) 次の一次関数のグラフを書きなさい。

ア. $y = 3x - 2$
 イ. $y = -\frac{1}{2}x + 2$
 ウ. $x = 2$

図9 : グラフ問題

「一次関数」の単元の中では重要なところであり、教科書ではグラフ理解のために多くのページを使っている。

プログラミング学習のタイマーによる図形の動きが、グラフの直線上の動きになっていることから理解できるようになると予想した。図形の動いた軌跡が画面上に直線で描かれるわけではないが、動きの指定がxyの変化量であるため、間接的に直線を理解できると考えた。プログラミングで、一次関数の式を意識するわけではないが、図形を動かすことによって間接的に理解できる問題と考えた。

3. 2 視覚的にとらえられない問題

数学の教科書から、「一次関数」における基本問題を選んだ。関数の変化を数としてとらえる問題である。

まず、図10のような一次関数の「表の空欄埋め問題」である。

(2) y が x の一次関数で、 x の値に対応する y の値が次の下の表のようになっている。表のア、イの空欄にあてはまる数を予想しなさい。

x	-2	-1	0	1	2	3	イ
y	-4	ア	2	5	8	11	17

図10 : 表の空欄埋め問題

この問題は、「一次関数」の導入のところで使われている基礎的な問題であり、教科書問題集によく出ている。表の見方がわかり、となりの数値との差を考えることにより空欄の数値を予想する。

次に、図11のような一次関数の式から変化の割合を求める「変化の割合問題」である。

(3) 一次関数 $y = \frac{1}{3}x + 1$ について、次の問いに答えなさい。

① x が6のときの y の値を求めなさい。
 ② x の値を-3から3まで変化させるとき、 y はいくら増えますか。

図11 : 変化の割合問題

この問題は、中学2年の「一次関数」の教科書にある基礎問題であるが、「変化の割合」という言葉が理解できない生徒も多い。そこ

人が事後テストで正解するようになったかを調べた。(以下、この生徒を「学力向上者」と呼ぶ。)表3に事前テスト不正解かつ事後テスト正解の人数/事前テスト不正解の人数を示す。

表3：座標問題の学力向上者

問題	学習クラス	非学習クラス
①点(5, 3)	1/2人(50.0%)	4/5人(80.0%)
②点(2, -3)	1/2人(50.0%)	3/6人(50.0%)
③点(-3, -5)	0/2人(0.0%)	3/8人(37.5%)
④点(-4, 4)	0/3人(0.0%)	3/8人(37.5%)

学習クラスの対象生徒が少なく、学習クラスと非学習クラスに差があるとは言えない。

次に、事前テストで正解した生徒のうち、何人が事後テストで不正解になったかを調べた。(以下、この生徒を「学力低下者」と呼ぶ。)表4に事前テスト正解かつ事後テスト不正解の人数/事前テスト正解の人数を示す。

表4：座標問題の学力低下者

問題	学習クラス	非学習クラス
①点(5, 3)	2/19人(10.5%)	6/17人(35.3%)
②点(2, -3)	2/19人(10.5%)	6/16人(37.5%)
③点(-3, -5)	2/19人(10.5%)	5/14人(35.7%)
④点(-4, 4)	1/18人(5.6%)	7/14人(50.0%)

人数は少ないが学習クラスと非学習クラスに差が認められる。学習クラスの方が、学力低下者の人数・割合とも少ないことがわかる。非学習クラスの方の約1/3～半数の生徒は、事前テストで正解した問題を間違えるようになっていることがわかる。

次に、正答率が全体として減少する傾向にあるため、不正解のパターンを調べた。すると、図14のような座標の点を描く問題を直線で答えるという点と直線を混乱している間違いが多く見られた。このような生徒は、事後テストでは、学習クラス1名、非学習クラス5名であった。不正解者のほとんどである。

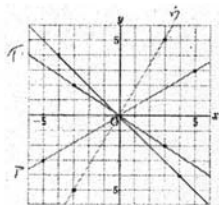


図14：座標問題の間違い例

「グラフ問題(図9)」の正答率は、表5のようになった。

表5：グラフ問題の正答率

問題	学習クラス	非学習クラス
	事前→事後	事前→事後
① $y=3x-2$	52.4→66.7%	72.7→81.8%
② $y=-1/2x+2$	47.6→66.7%	40.9→63.6%
③ $x=2$	19.0→57.1%	22.7→50.0%

学習クラスも非学習クラスも正答率が増加している。

「学力向上者」と「学力低下者」を調べると、表6と表7のようになった。

表6：グラフ問題の学力向上者

問題	学習クラス	非学習クラス
① $y=3x-2$	5/10人(50.0%)	2/6人(33.3%)
② $y=-1/2x+2$	6/11人(54.5%)	7/13人(53.8%)
③ $x=2$	10/17人(58.8%)	8/17人(47.1%)

学習クラスの方が、正解する生徒の増加が大きいことがわかる。

表7：グラフ問題の学力低下者

問題	学習クラス	非学習クラス
① $y=3x-2$	2/11人(18.2%)	0/16人(0.0%)
② $y=-1/2x+2$	2/10人(20.0%)	2/9人(22.2%)
③ $x=2$	2/4人(50.0%)	4/5人(80.0%)

両クラスとも学力低下者の人数が少なく、学習クラスと非学習クラスに明らかな差があるとは言えない。

4.2 視覚的にとらえられない問題

「表の空欄埋め問題(図10)」の正答率は、表8のようになった。

表8：表の空欄埋め問題の正答率

問題	学習クラス	非学習クラス
	事前→事後	事前→事後
①yの値	71.4→71.4%	45.5→54.5%
②xの値	66.7→52.4%	40.9→54.5%

正答率を比較すると、学習クラスは変化していないが、非学習クラスは増加している傾向がある。

次に、「学力向上者」と「学力低下者」を調べると、表9と表10のようになった。

表9：表の空欄埋め問題の学力向上者

問題	学習クラス	非学習クラス
①yの値	3/6人(50.0%)	6/12人(50.0%)
②xの値	0/7人(0.0%)	6/13人(46.2%)

表10：表の空欄埋め問題の学力低下者

問題	学習クラス	非学習クラス
①yの値	3/15人(20.0%)	2/10人(20.0%)
②xの値	3/14人(21.4%)	3/9人(33.3%)

非学習クラスの正答率が増加しているのは、学力向上者が多かったことによるものであることがわかる。

「変化の割合問題(図11)」の正答率は、表11のようになった。

表11：変化の割合問題の正答率

問題	学習クラス	非学習クラス
	事前→事後	事前→事後
①yの値	47.6→61.9%	54.5→59.1%
②変化の割合	42.9→23.8%	18.2→27.3%

正答率を比較すると、学習クラスは変化していないが、非学習クラスは増加している傾向がある。

次に、「学力向上者」と「学力低下者」を調べると、表12と表13のようになった。

表12：変化の割合問題の学力向上者

問題	学習クラス	非学習クラス
①yの値	4/11人(36.4%)	5/10人(50.0%)
②変化の割合	3/12人(25.0%)	4/18人(22.2%)

表13：変化の割合問題の学力低下者

問題	学習クラス	非学習クラス
①yの値	3/10人(30.0%)	2/12人(16.7%)
②変化の割合	8/9人(88.9%)	2/4人(50.0%)

学習クラスの正答率が減少しているのは、学力低下者が多かったことによるものであることがわかる。

4. 3 論理的思考問題

「論理的思考問題(図12、図13)」の正答率は、表14のようになった。事前テストでは小問数が3、事後テストでは小問数が2と違ったため、事前テストの小問(2)をはずして比較している。小問についての一貫性がとれて

いないため、問題を全体で1つとしてとらえた完全正答でも比較している。

表14：論理的思考問題の正率表

問題	学習クラス	非学習クラス
	事前→事後	事前→事後
①小問1	42.9→71.4%	22.7→50.0%
②小問2	14.3→33.3%	4.5→36.4%
完全正答	9.5→23.8%	0.0→27.3%

正答率は、学習クラス、非学習クラスともに増加している。全体に占める割合は、ほぼ同じで違いは見られない。

次に、「学力向上者」と「学力低下者」を調べると、表15と表16のようになった。

表15：論理的思考問題の学力向上者

問題	学習クラス	非学習クラス
①小問1	7/12人(58.3%)	9/17人(52.9%)
②小問2	3/18人(16.7%)	3/21人(14.3%)
完全正答	4/19人(21.1%)	6/22人(27.3%)

表16：論理的思考問題の学力低下者

問題	学習クラス	非学習クラス
①小問1	1/9人(11.1%)	2/5人(40.0%)
②小問2	2/3人(66.7%)	0/1人(0.0%)
完全正答	1/2人(50.0%)	0/0人(0.0%)

非学習クラスには、事前テストでの完全正答者がいなかったため、比較できない。また、人数が少数のため、明らかな違いや変化があるとは言えない。

5. 検証テスト結果の考察

本節では、実施した検証テストについて考察し、課題について検討する。

5. 1 視覚的にとらえる問題

「座標問題(図8)」については、全体的に正答率が減少している。生徒の解答から、学習後、時間が経つと座標の正負の位置や(x, y)のどちらが横軸か縦軸かを忘れたり、一次関数のグラフを学習したために点と直線を混同して考える生徒ができたりするためであると考えられる。学習クラスの学力低下者が少なかったのは、プログラミング学習で、グラフィックスの位置を示すために常に座標を指定

するなどの作業で、座標の概念をきちんと身につけていったためであると考えられる。

「グラフ問題(図9)」については、全体として正答率が増加しているが、学習クラスの方が学力向上者がやや多い。これは、「タイマーによるアニメーション」の学習において、図形の移動が直線としてとらえられたため、視覚的に直線のイメージが身につけていったものではないかと考えられる。

以上の結果により、「視覚的にとらえる問題」については、学習効果があると考えられる。

5. 2 視覚的にとらえられない問題

「表の空欄埋め問題(図10)」「変化の割合(図11)」ともに、非学習クラスの方がよい結果が出ている。これは、プログラミング学習以外の要因が考えられる。ただ、事前テストと事後テストを同じ条件で実施することは難しく、非学習クラスの事前テスト実施の際に十分な時間がとれなかったことが原因となっている可能性もある。

「変化の割合問題」で、学習クラスの学習低下者が多い原因は特定できないが、この問題は、「 x の値を変化させるとき、 y はいくら増えますか」という言葉の意味が理解できるかどうかと言う要素が大きく、学習直後には覚えていたが、学習後2ヶ月の間に忘れたものと思われる。

「表の空欄埋め問題」「変化の割合問題」とも、学習効果は認められない。ただ、全12時間と少ない時間での検証であるため、長期の学習によっては効果があるかもしれない。

5. 3 論理的思考問題

正答率としては、学習クラスも非学習クラスも向上しているが、とくに差は認められない。事前テストと事後テストにおいて、全く違う問題のため比較が難しい。事前に予備調査を実施し、難易度はほぼ同じに設定したが、同じになったかどうか、非常に難しい。よって、論理的思考問題に関して効果があるかどうかは不明である。

6. まとめ

今回の検証は、プログラミング学習をすることによって、検証テストの正答率があがるだろうという予想をもとに行った。これは、学習が着実に積み上げられるものであり、効果がない場合は正答率はそのまま、効果がある場合には上がるのと考えていた。しかし、実際は学習したことを忘れてたり知識が混乱することによって、全体の正答率が下がることもあることを知った。

検証テストについては、客観的なデータを得る目的で行ったが、テストを実施したことによっていろいろなことがわかった。まず、同じ条件で2回のテストを実施する難しさである。たくさんの数学嫌いの生徒にいかにも真剣に取り組ませるかという問題があった。また、週1回の授業において、テストの時間を確保することは難しい。論理的思考問題については、同じ難易度の問題を作ることは非常に難しいこともわかった。論理的思考問題は、問題によって思考方法が異なり、同じように比較するためには問題に関するデータをたくさん集めなくてはならない。

プログラミング学習の授業時間については、技術の授業は週1時間である。学校行事や休日などにより欠ける場合があり、間隔が2週間になると復習に時間を要する。今回の検証では、時間割を組み替え、必ず週1時間の授業を確保した。全職員の理解と協力ができないところである。

今回の検証では、全12時間と少ない時間での効果を検証したが、長期にわたる学習効果や学習後の経過については、今後、研究を深めていきたい。

参考文献

- [1]井戸坂幸男, 足利裕人, 紅林秀治, 鎌田敏之, 兼宗進, 久野靖, プログラミング学習の他教科への波及効果～数学と考える力を対象とした調査と検証計画～, 2006年8月, 情報処理学会論文誌:S5S2006 情報教育シンポジウム論文集, vol.2006, No. 8, pp. 89-96, 2006.
- [2] 兼宗進, 御手洗理英, 中谷多哉子, 福井眞吾, 久野靖, 学校教育用オブジェクト指向言語「ドリトル」の設計と実情報処理学会論文誌: プログラミング, vol. 42, No. SIG11 (PR 0 12), pp. 78-90, 2001.