

情報教育の音楽化による高校生へのプログラミング入門¹

辰巳 丈夫(東京農工大学) 兼宗 進(一橋大学) 小原 格(東京都立町田高等学校)
野部 緑(大阪府立桃谷高等学校) 酒徳 峰章(ウノウ) 山澤 昭彦(荘銀総研)

概要

情報教育の音楽化とは、情報の授業に音楽を利用する概念であり、音楽の授業にコンピュータなどを利用することではない。筆者らは、高校生を対象としたプログラミング入門の授業において、オブジェクト指向プログラミング環境であるドリトルの音楽機能を利用した授業を設計し、実際に授業を行なった。本発表では、その概要について述べる。

Musical methodology of Programming Education in Japanese highschools
TATSUMI Takeo (Tokyo Univ. of Agri& Tech.), KANEMUNE Susumu (Hitotsubashi Univ.)
OHARA Itaru (Machida Highschool of Tokyo), NOBE Midori (Momodani Highschool of Osaka)
SAKATOKU Mineaki (Unoh Inc.), YAMAZAWA Akihiko (Shougin souken)

abstract

In 2001, Tatsumi pointed out that the musical methodology is useful in computer literacy education. In 2007, we exercised our method in two different type classes. We used "Dolittle" which is OOP(object oriented programming) enviroment for beginner. In this paper, we will describe our works and reports about classes of highschools.

1 情報教育の音楽化

情報処理技術の発展は我々の生活を変えている。この変化は教育の世界でも例外でない。教育現場は情報処理機器を使うことで「効率」を追求したり、新たな教育内容を付け加えたり、教育方法を変更するなどの変化を遂げた。

本論では、中学校～高校程度におけるプログラミング教育の方法として、音楽の授業にプログラミング教育を取り入れる方法、すなわち「情報教育の音楽化」を考える。

1.1 アルゴリズムによる処理の定式化、プログラミング

情報教育を支える3つの目標のうちの「情報の科学的理解」の内容には、「自らの情報活用を評価・改善するための基礎的な理論や方法の理解」という文が含まれている。

現在の高等学校での教科「情報」の指導要領では、「プログラミング」が正式な内容として取り上げられていないので、この部分の学習は「定式化した処理が存在すること」として取り上げられ、具体的には「アルゴリズム」の学習で構成される。

1.2 著作権

「情報社会に参画する態度」の内容のなかでも「著作権」を学ぶことは非常に重要とされている。しかし、「自分が書いた電子メールの著作権」や「他人が書いたwebページの著作権」という指摘をしても、実感が伴わないことがある。

高校生、あるいは中学生にとってもっとも身近な著作物とは「楽曲」である。

¹本稿は、財団法人ヤマハ音楽振興会による2007年度研究活動支援『「音楽教育の情報化」と「情報教育の音楽化」―音楽文化を広める情報教育を目指して―』(研究代表者:辰巳 丈夫)で行なわれた研究活動の報告である。

音楽の授業で生徒が作曲した曲を web ページに掲載する場合にどのような権利の問題が生じるのか、あるいは著名な曲を耳で聞いて譜面をおこした場合、その曲を web ページに掲載していいのか、歌詞はどのように取り扱うべきなのか、といったことを取り上げれば、生徒の著作権についての関心も高くなり、著作権を実感をもって学習することができる。

2 プログラミング教育の音楽化

前節において、「情報教育」の内容を「音楽教育」の形態をとりながら行なうことが可能であることを指摘した。本論文では今後、その手法を「情報教育の音楽化」と呼ぶこれは、「あくまでも目標は情報教育の目標、実施は音楽教育の手法」となる授業を行なうことである。

本節では、特に「『プログラミング教育』において音楽的題材を用いる場合」にどのような展開が可能かについて分析、提案を行なう。

2.1 初等中等教育に必要なプログラミングの体験の要素

初等中等教育においてプログラミングを扱う際に、一体どんな要素が必要となるだろうか。

初等中等教育におけるプログラミングの体験は、専門的な技術者の育成を目標とするものであってはならない。「プログラミングを学ぶ体験」は、ほぼすべての学習者にとって、コンピュータやネットワークの動作原理を、より印象深く、より短い時間で（効率的に）学ぶことに寄与する。その目的を達成するために必要な構成要素について議論する。

2.1.1 「作成→動作確認→修正」の手順化された作業

コンピュータプログラムは機械によって作られるものではなく、人間が作成するものであるということを実感させるために、「人間がプログラムを実際に書いて、それを実行させては修正をし、また実行させては修正をする」という手順化された作業を実際に経験することは効果的である。

このような授業を実現するためには、学習が容易で授業の最初の時間から使い始められるプログラミング環境が必要になる。

2.1.2 テストとデバグ

上に述べた「手順化された作業」の合間に、動かないプログラムを作成してしまったり、プログラムが突然動かなくなってしまうたり、予期しない動作をしてしまうという体験をすることになる。そこで行なわれる作業はプログラムの動作テストとデバグである。

このような作業を行うためには、短いプログラムで意味のある動作を書けるプログラミング言語が必要になる。長い記述が必要な言語では、エラーを発見して修正することが飛躍的に困難になるからである。

2.1.3 動作エラーに対する取り扱い

もし、システムプログラム開発や、高価な実験機器のプログラミングを扱うとするならば、エラーは大きな被害に直接的に結び付く。

一方、タートルグラフィックスや音楽演奏などのプログラムにエラーがあっても、それは「画面の書き間違い」「演奏の間違い」という状況にのみ現れる。そのとき、教室に笑い声がすることはあっても、エラーを作成してしまった人を責める雰囲気は生じない。だからこそプログラムにエラーがあるかどうかを気軽に試すことができる。

一方、グラフィックスや音楽のプログラムでは、プログラムのエラーの存在は誰にとっても明らかである。特に音楽の場合には、音程やリズムなどの違いで曲が正しく演奏されていないことは、本人だけでなく教師や同級生にも明らかになる。よって、バグの検知と、デバグによって正しく修正されたことの確認が、極めて正確に行われることになる。

このような学習を行うためには、グラフィックスや音楽をプログラミングの題材として使えるプログラミング環境が必要である。

2.1.4 コンパイラ・インタープリタによる解釈の幅

例えばC言語では、各命令をセミコロンで終了することになっているが、これを2バイト文字（いわゆる全角文字）で記述するとエラーになり実行できない。空白文字についても同様である。このように解釈の幅が非常に狭いプログラミング言語は多い。一方、コンピュータの仕

組みをよく理解できていない学習者は「なぜ正しく空白が空いている（ように見えるのに）動かないのか」ということも理解できない。

そこで、学習に用いる実行環境系では、これらの小さなミスを吸収する解釈の幅が必要である。ただし、あまりに多様な解釈が可能となると、「コンピュータは文字通り（杓子定規に）しか動かない」という性質の体験をせずに体験が終了してしまう可能性があることは注意を要する。

2.1.5 プログラムの構成要素の体験

プログラム作成に当たっては、「データ構造の析出」「アルゴリズムの意識化」「接続」「繰返しと脱出」「制御」「部品化と再帰」「同期」などの概念が必要となる。これらのうち、基本的な概念については、簡単な記述で扱えることが必要である。

ただし、初等中等教育におけるプログラミングの授業では、上記の概念のいくつかを取り入れれば十分である。そこで興味・関心を持った一部の児童・生徒が学習を進めればよい。

2.2 情報教育にプログラミングを取り入れることの是非

初等中等教育における情報教育の中に、プログラミングを取り入れることの是非については、情報教育の中心がアプリケーションの操作方法の習熟を目的とする「情報リテラシー教育」に移行した1990年代後半頃から、既に多くの議論があった。その内容の紹介については、2005年12月、辰己らによるCE82での発表[1]において既に述べたので、ここでは割愛する。

3 「ドリトル」の音楽機能

ここでは、今回の授業で使用した、初心者のためのオブジェクト指向プログラミング言語「ドリトル」[2]で使用されている音楽機能について述べる。

3.1 メロディオブジェクト

ドリトルでは、最初にオブジェクトを生成し、それに命令を送ることでプログラムを実行する。音楽では、メロディオブジェクトを利用して、次の音楽を演奏することが可能である。

```
きらきらぼし=メロディ! 作る。
きらきらぼし!『ドドソソララー』追加。
きらきらぼし!『ファファミミレドー』追加。
きらきらぼし!『ソソファファミミレー』追加。
きらきらぼし!『ソソファファミミレー』追加。
きらきらぼし!『ドドソソララー』追加。
きらきらぼし!『ファファミミレドー』追加。
僕の楽器=楽器!『ピアノ』作る。
僕の楽器!(きらきらぼし)設定。
僕の楽器!演奏。
```

1行目では、「作る」で新しいメロディオブジェクトを生成し、「きらきらぼし」という名前をつけた。2行目以降では、そのオブジェクトに旋律を追加している。続いて「僕の楽器」という名前の楽器オブジェクトを生成し、楽譜として「きらきらぼし」を指定して演奏させている。

このプログラムは上から1行ずつ逐次的に実行される。よって、プログラムの「接続」の概念が取り上げられている例になっている。

3.2 音楽の構造

「きらきらぼし」のメロディを観察すると、音楽が構造を持っていることがわかる。

```
ドドソソララーファファミミレドー
ソソファファミミレーソソファファミミレー
ドドソソララーファファミミレドー
```

1行目と3行目は同じ旋律であり、2行目は短い旋律が2回繰り返されていることがわかる。仮に1行目の旋律を「Aメロ」、2行目の短い旋律を「Bメロ」と名づけると、きらきらぼしは、次のように記述することができる。

```
Aメロ
Bメロ Bメロ
Aメロ
```

プログラムで音楽を表現する場合には、ここで見たように、音楽の構造に着目して、それをプログラムの構造にうまく置き換える学習が効果的である。

3.3 繰返しの利用

きらきらぼしをプログラムで構造化するために、「Bメロ」を繰り返して表現する。

次の例では、プログラムの一部を括弧（「...」）で囲むことで

```
「きらきらぼし!『ソソファファミミレー』追加。」
という形のブロックオブジェクトにし、そのオブジェクトへ「自分を2回繰り返す」という
```

メッセージを送り繰り返しを実現している。

```
きらきらぼし=メロディ! 作る。
きらきらぼし!『ドドソソララソー』追加。
きらきらぼし!『ファファミミレドー』追加。
「きらきらぼし!
『ソソファファミミレー』追加!」
2回 繰り返す。
きらきらぼし!『ドドソソララソー』追加。
きらきらぼし!『ファファミミレドー』追加。
僕の楽器=楽器!『ピアノ』作る。
僕の楽器!(きらきらぼし)設定。
僕の楽器!演奏。
```

これは、プログラムの「繰り返し」の概念が取り上げられた例になっている。

3.4 条件分岐の利用

「Aメロ-Bメロ2回-Aメロ」という構造については、「Aメロ-Bメロ2回」を2回繰り返す形で表現できる。ただし、「Bメロ2回」は繰り返しの1回目だけ実行する。

次のプログラムでは、条件分岐を使ってBメロを1回目の繰り返しのときだけ追加している。

```
きらきらぼし=メロディ! 作る。
「|Aメロ回数|
きらきらぼし!『ドドソソララソー』追加。
きらきらぼし!『ファファミミレドー』追加。
「Aメロ回数==1!」なら「
「きらきらぼし!『ソソファファミミレー』追加!」2回 繰り返す。
」実行。
」!2回 繰り返す。
僕の楽器=楽器!『ピアノ』作る。
僕の楽器!(きらきらぼし)設定。
僕の楽器!演奏。
```

音楽をプログラムで扱うことで、音楽が持つ構造をプログラムで表現することができる。そして、プログラムを学ぶときに必要な基本構造を自然な形で学ぶことができる。

3.5 乱数

ここまでで、音楽をプログラムで扱うための基本的な構造を説明した。コンピュータを利用することで、創造的な作業が可能になることを体験的に学ぶことが可能である。この節では、プログラムを用いることで、自動的なメロディの生成を実現する例を示す。

次のプログラムでは、特徴のある音階として琉球音階の音階を配列に入れ、それを乱数を使

って組み合わせることで自動作曲を実現している。「明るい楽句」と「暗い楽句」は「Aメロ」と「Bメロ」に相当するメロディの断片であり、これらを「明るい楽句」「明るい楽句」「暗い楽句」「明るい楽句」の順に組み合わせることで、ひとつの曲を構成している。

```
琉球音階=配列!作る。
琉球音階!『ド』入れる『ミ』入れる
『ファ』入れる『ソ』入れる
『シ』入れる。
```

```
僕の楽譜=メロディ!作る。
明るい楽句=メロディ!作る。
「明るい楽句!
(琉球音階!(random(5))見る)追加。」
!6 繰り返す。
明るい楽句!『ドー』追加。
```

```
暗い楽句=メロディ!作る。
「暗い楽句!
(琉球音階!(random(5))見る)追加。」
!8 繰り返す。
```

```
僕の楽器=楽器!『ギター』作る。
「僕の楽譜!
(明るい楽句)追加。」!2 繰り返す。
僕の楽譜!(暗い楽句)追加。
僕の楽譜!(明るい楽句)追加。
僕の楽器!(僕の楽譜)設定。
僕の楽器!演奏。
```

3.6 合奏

複数の旋律を演奏するときには、複数の楽器を組み合わせるバンドオブジェクトを利用して合奏を行なうことが可能である。

次のプログラムでは、「たろうの楽器」と「はなこの楽器」という2つの楽器にそれぞれのメロディを設定し、「マイバンド」という名前のバンドオブジェクトに「たろうの楽器」と「はなこの楽器」を追加して同時に演奏させている。

```
たろうの楽譜=メロディ!作る。
「たろうの楽譜!『ドドソソララソー』追加。
たろうの楽譜!『ファファミミレドー』追加。」!2 繰り返す。
たろうの楽器=楽器!『ピアノ』作る。
たろうの楽器!(たろうの楽譜)設定。
```

```
はなこの楽譜=メロディ!作る。
「はなこの楽譜!『ドソーラーソー』追加。
はなこの楽譜!『ファーミーレドー』追加。」!2 繰り返す。
はなこの楽器=楽器!50 作る。
```

はなこの楽器！（はなこの楽譜）設定。

マイバンド = バンド！ 作る。
マイバンド！（たろうの楽器）追加。
マイバンド！（はなこの楽器）追加。
マイバンド！ 108 テンポ。
マイバンド！ 演奏。

このプログラムでは、二つの楽譜を作りだし、それらを楽器に設定し、楽器を集めたバンドを作り、バンドにテンポを設定して、演奏命令を送っている。現実の楽器演奏とほぼ同じモデルでのプログラミングが可能になっている。

4 実証授業

我々が提唱する「情報教育の音楽化」の授業が実現可能であることを確認するために、2つの学校において実証授業を行なった。ここでは、その概略を述べる。

4.1 (財)ヤマハ音楽振興会による研究支援

本プロジェクトは、2007年2月に財団法人ヤマハ音楽振興会の研究支援に採択された。本プロジェクトの「情報教育の音楽化」のコンセプトが、「音楽文化を広めるための活動」としての認定を受けた結果である。この結果、研究助成金を受けたプロジェクトとして活動することで、より本格的な教材作成や、有識者による支援を受けることが可能になった。

4.2 全体計画

行った実証授業のねらいは次の通りである。

- 2つの学校で実証実験を行なう。
さまざまな学校で授業を行えることを確認するために、タイプの異なる学校で行う。
- 担当する教員は、音楽に造詣が深い情報科の教員とする。
高校の普通教科「情報」で授業を行う。音楽を題材に授業を行うため、音楽の基本的な知識が必要となる。
- 標準的なテキストを作成する。
授業を通して、「情報教育の音楽化」を実現するための標準テキストを作成し、授業を通して改良する。

- 使用する楽曲の著作権にも注意し、JAS-RACと調整を行なう。
音楽の著作権処理を行うことで、生徒の作品プログラムをインターネットで公開することを可能にする。この体験を通して、著作権などを含めた総合的な情報の学習につなげていく。
- 時間的に余裕があればドリトル以外の処理系についても扱う。
音楽を演奏できるプログラミング言語である「ドリトル」だけでなく、プログラミングも可能な音楽ソフトウェアである「サクラ」についても授業で扱ってみる。

4.3 東京都立町田高等学校での授業

2007年6月に授業を行った。授業を担当したのは、情報科の小原格教諭で、希望者のみの「特別講義」として放課後に実施された。参加した生徒は合計42名で、そのうち女子が29名であった。町田高校では、男子生徒のほうが女子生徒より多いことから、この授業に対する女子生徒の興味が強かったことを表している。



写真1：町田高校での様子

本授業ではオンライン版ドリトル²を使用した。これは、Java JRE を利用可能な web ブラウザさえあれば動作する。音源も Java 組み込みの MIDI 音源を利用することが可能である。オンライン版ドリトルは、インストールが不要であり授業環境の準備が簡単であったが、一方、Java アプレットの制限から、作成したファイルをローカルのディスクに保存するためにコピー&ペーストが必要になった。

以下に、授業のカリキュラムを示す。

²<http://dolittle.eplang.jp/pconline/applet.html>

- 1日目(70分) ドリトルの紹介。簡単なプログラム作成。
- 2日目(70分) Moodle への登録。繰返しの導入。制御(「もし……ならば、」)の導入。
- 3日目(100分) 自分が好きな音楽作品を入力して楽しむ。テキスト音楽サクラの紹介。

辰己が作成した24ページの冊子を授業テキストとして用いた。授業を担当した教員や生徒からは「概ねわかり易い」という評価であったが、「メニューを利用した演奏は、意図がわからない」「合奏の説明がない」「繰返しの部分に工夫が必要」「もし、楽譜を読めない生徒がいたら、難しい(当該授業は希望者のみの選択なので、楽譜を読めない生徒はほとんどいなかった。)」という意見があった。

4.4 大阪府立桃谷高等学校での授業

2007年10月に授業を行った。授業を担当したのは情報科の野部緑教諭である。また、1~2名の教員がTTとしてサポートした。「情報B」の「コンピュータによる処理」の一部として行なわれ、授業に参加した生徒数は約20名であった。



写真2：桃谷高校での様子

本授業ではインストール版ドリトルを使用した。作成したファイルの保存は容易であったが、PC教室への事前のインストール作業が必要であった。

この授業では、町田高校との相違点、教材の反省点を踏まえ、次の変更・改良を行なったテキストを使用した。

- インストール版への対応。

³本稿の末尾(図1)に「楽譜とドリトルの音名との対応プリント」を示す。

- メニューを利用して楽器を選んで演奏するページを削除。
- 合奏の説明を追加。

希望者のみが参加した町田高校と異なり、「情報B」の授業の中で実施された。桃谷高校では、「情報B」の履修前提として「情報Aを履修済が望ましい」としているが、音楽的な知識を前提としているわけではない。そのため、授業を担当した野部が補助プリント³を作成・配付した。これらの補助プリントは有効であり、授業を効率よく進めることができた。

- 楽譜とドリトルの音名との対応プリント。
音楽に興味を持つ生徒と持たない生徒が参加しているため、楽譜を読めない生徒に配慮した。
- 演奏手順とフローチャートの関係を穴埋め形式で解答する補助プリント。
単に音楽を演奏して終るのではなく、授業内容が「情報B」の学習の狙いと合致するように配慮した。

4.5 結果と考察

以下は、授業の結果である。定量的な分析は完了していないため、生徒の感想と教員の観察した結果からの考察を紹介する。

4.5.1 学習態度の変化

どちらの授業でも、多くの生徒が熱心に学習に参加した。町田高校の授業は希望して参加した生徒が対象だったこともあるが、桃谷高校の授業は正規の授業であり、通常の授業に集中して参加することが難しい生徒も多く含まれていたことから、今回の授業が特別な授業であったことがわかる。担当した教員の間から、「普段集中できない生徒が熱中して学んでいる」という驚きの声が上がっていた。

4.5.2 著作権に関する体験的学習

高校生の多くは、携帯電話で利用できるwebサイトなどから、著作権の許諾を得ていない音楽ファイル(野良MP3)のダウンロードを経験している。

今回の授業では、自分で入力した曲を Web の学習支援システム (Moodle) に登録することを通して、音楽を公開することの著作権を意識することができた。

また、授業の途中で、担当教員が曲を公開する際の著作権を説明したところ、一部の生徒は、自分自身が利用しているサイトの違法性について、深刻に考えている様子を観察することができた。

当初の計画では、生徒が入力した曲の権利関係を判定し、JASRAC などの権利団体に手続きを取って許諾を得た後にインターネットに公開することを予定していた。しかし、生徒に「曲を自由に入力して演奏してみるの、楽譜を持参して欲しい」と連絡したところ、生徒が授業専用掲示板に入力した曲名が不完全なために JASRAC 管理の曲かどうかを判定できなかったり、一部の楽曲が JASRAC ではなく、ゲームソフト製造会社管理の曲であることも判明した。そのため、著作権の有無に関わらず、作成されたプログラムの外部への公開は今回は行わないことにした。

4.5.3 ネットワークの通信を理解した

ほとんどの生徒たちは、生まれて初めての音楽プログラミングの体験を通して、「ネットの音楽はプログラムが演奏している」ということと、「ネットの音楽は誰かが打ち込んでいる」ということを理解した。これは、プログラミングを体験することで、はじめて実感を持って理解することができたと考えている。

4.5.4 楽譜が読めない生徒への対応

町田高校の授業では、音楽プログラミングを希望して受講した生徒が対象であったため、ほとんど全員が楽譜を読みながらメロディを入力することが可能であった。

一方、桃谷高校の授業では、半分近くの生徒が楽譜を読むことに慣れていなかった。そこで、図 1 のプリントなどを配布してサポートした。授業の最初は音楽に興味を持っていなかった生徒も、楽譜からのメロディの入力ができるようになると、興味を持って取り組む姿が見られた。これは、楽譜からの音符の入力が自分にも可能であると感じられるようになったこと

と、自分で入力した楽譜が演奏できたことで、音楽が身近になったためではないかと考えている。

このような音楽プログラミングの体験を通して、音楽そのものへの興味につながることも期待している。

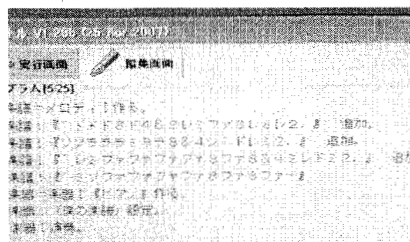


写真 3：生徒が作成したプログラム

5 まとめ

本稿では、「情報教育の音楽化」を提案した。そして、実証授業を通して、ドリトルの音楽機能という適切な教材や、今回作成したテキストのような適切な資料、そして今回担当してくれた音楽と情報を理解している教員がいれば、情報の授業が可能であること、そして生徒が実際に多くのことを学べることを示した。

今後は、授業実施前後のアンケートについて分析を行なうとともに、教材の改良と公表を進める予定である。

謝辞

本実証授業に参加してくれた、すべての生徒、教員、関係者の皆様、および、財団法人ヤマハ音楽振興会に感謝をします。

参考文献

- [1] 辰己丈夫 (東京農工大学), 兼宗進 (一橋大学), 久野靖 (筑波大学). ドリトルと「情報教育の音楽化」. 情報処理学会研究会「コンピュータと教育」, pp. 77-84, 2005. 2006-CE-82-12.
- [2] 兼宗進, 中谷多哉子, 御手洗理英, 福井眞吾, 久野靖. 初中等教育におけるオブジェクト指向プログラミングの実践と評価. 情報処理学会論文誌, Vol. 44, No. 13, pp. 58-71, 2003.

自分で楽譜を入力するために

楽譜と音階 (基本)



音階のオクターブの横目では、上げる、下げるといった命令が必要



(半音あがる) や ♭ (半音下がる) がついた音は、音階のあとに#や♭をつけます。



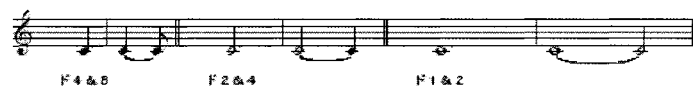
楽譜の最初に調号があるときは、それに対応した音階に#や♭をつけます。



音の長さや命令 音階のあとに続けます。(4については省略できます)



付点は長さが半分の音を追加します。(右と左は同じ長さです)



休符の長さ命令 休符は・であらわします。



^音記号の音階 (実際の音の高さは、1オクターブ低くなります)



図 1: 楽譜とドリトルの音名との対応プリント