

# プログラミング教育の可能性

愛知県立岩倉総合高等学校教諭 横井 尚美

## 1 はじめに

2015年、日本の通信事業自由化から30年、インターネットの普及から20年を迎えた。人工知能(AI)やクルマの自動運転、ドローンなど、ICTの進化がビジネスや社会の仕組み、雇用環境を大きく変えようとしている。背景には、あらゆるものがネットワークにつながるIoTや、大量のデータを扱えるクラウド技術の広がりがある。我が国の将来を担う生徒たちは、こうした変化を乗り越え、高い志や意欲をもつ自立した人間として他者と協働し、未来を切りひらいていく力を身に付けることが求められている。

私たちは、社会生活を営む上で、多くの選択肢から選択し行動している。選択肢には、唯一の正解と断定できるものはない。個々人が、その時々に応じた最適解を主体的に選択できる能力(=論理的思考力)を身に付けなければならない。教育には「逆算」の発想が必要である。教師には、商業教育を通して、「どのような能力を身に付けた生徒を育てたいのか」という目標から逆算し、日々の教育活動を実践することが求められている。プログラミング教育は、生徒の論理的思考力、想像力を主体的に育てるのに適している。

そこで、今後のIT教育の方向性を明らかにし、本校のプログラミング教育を、授業改善による指導内容の標準化についてまとめ、プログラミング教育のもつ可能性について考察する。

## 2 IT教育の方向性

### (1)システム開発言語は、Javaへ移行する傾向

IPAの『ソフトウェア開発データ白書2014-2015』によれば、開発されたシステムの種別において「アプリケーションソフト」が93.9%を占め、そのほとんどは業務システムの構築である。開発言語では、「Java」が30.4%と最も多く、次いで、「COBOL」が15.3%、「VB」、「C」、「C++」が続く。

開発言語を経年比較すると、「VB」、「COBOL」から、「Java」への移行が見受けられる。

### (2)スマートフォンOSの主流は「Android」、Androidアプリケーション開発環境は「Android Studio」へ変化

現在、デジタルデバイスの中で、最も急速に浸透しつつあるのがスマートフォンである。Net Applicationsの『2015年8月モバイルOSシェア発表』では、Androidは52.1%、iOSは40.8%であった。両者のシェア合計は92%を超えており、モバイルプラットフォームのOSとしてAndroidとiOSが強い影響力をもっていることが分かる。Androidは、今やスマートフォンの代名詞となりつつある。Android最大の特徴は、誰でも無料でアプリケーションの開発ができ、作ったアプリケーションを世界中に公開できることである。2013年5月、Googleは新たなAndroidアプリケーション開発環境として「Android Studio」というIDEを公開し、現在は、ADTのサポートを終了している。今後開発環境は、「Eclipse + ADT」から、「Android Studio」へ変化していくと思われる。

このように、ICT分野においては日々新しい流れが誕生している。我々教師は、「実学」である商業教育の「プログラミング教育」において、「流行」部分に目を向けた学びを実現していく姿勢と努力が求められている。

## 3 本校の現状と今後の取組

### (1)本校の「プログラミング教育」は、COBOLからJavaへ変更

本校では、平成26年度から科目「プログラミング」の指導言語をCOBOLからJavaに変更した。その理由は、Java言語の応用的活用方法の一つである「Android上で動作するアプリケーション開発」という「ものづくり」を通して、生徒に「実学」を習得させることである。Java最大の特徴は「Write Once, Run Anywhere」である。つまり、

生徒がもっている多種多様な Android スマートフォン画面に、自ら開発したアプリケーションを表示・動作させること、互いにアプリケーションを試し合うこと、校外の家族や友人に見せ、評価してもらうことが可能である。このことは、「感動→興味・意欲→クリエイティブな発想→さらなる知識の習得→感動……」といった「学びの連鎖」を生み、積極的な学びへのモチベーションを上げ、ひいては、進路選択の幅が広がることにもつながると考える。

## (2)指導の現状

総合学科である本校において、商業科目は全て選択科目であり、科目「プログラミング」は、1年次又は2年次に4単位を履修することができる。履修者数は、毎年変動するがおおよそ20名程度であり、その進路希望は、情報系専門学校進学と就職が半々で、大学進学希望者も3名程度いるという傾向である。授業目標は、9月に、全商情報処理検定プログラミング部門2級、1月に1級取得できる知識の習得である。「Eclipse」を用いたJava実習を軸にアルゴリズム習得を実施している。

## (3)問題点及び授業改善により指導内容の標準化を行う背景

### 【Java言語指導上の問題点】

- ア 生徒にとって、Javaプログラムの記述中には、見慣れない決まり事が多く、難解である。
- イ Java言語指導をオブジェクト指向から始めるのは難しい。
- ウ 手続き型言語としてJava言語指導を始めることが、Java本来の開発スタイルと乖離する。
- エ 「Android Studio」によるアプリケーション開発は、初学者にとってハードルが高い。
- オ Java言語指導者が少ない。
- カ オブジェクト指向の概念やデータ型など、COBOLプログラミングとの相違点が多く、指導者にとって学習することが負担となっている。
- キ 指導者は、視覚的な学習を推進するために、提示用教材を開発する必要がある、負担となっている。

### 【本校カリキュラム展開上の問題点】

多くの生徒は、情報処理技術者試験「基本情報」合格を目指し、科目「プログラミング」を履修する。2・3年次は学校設定科目を履修し、2年次10月に「ITパスポート」、3年次4月に「基本情報」、3年次10月に「応用情報」の合格を目標としてい

る。したがって、1年次にJava言語を習得しても、Android上で動作するアプリケーション開発実習ができるのは、3年次10月下旬からとなる。つまり、1年半以上のブランクを経て、Java実習に取り組むことになる。

### 【授業改善により指導内容の標準化を行う背景】

カリキュラムに定めてある選択科目でも、科目特性と選択希望者の人数によっては、非開講になる科目もあり、継続して毎年開講されるとは限らない。基礎科目履修を前提として位置付けられている科目は、年次進行で増加し、選択者が少なくなる傾向もある。商業科目は、開講講座数が複数となるものは少なく一人で授業担当することが多い。そのため、教科指導は、教師の力量に大きく依存し、人事異動により、学校内の教科指導レベルが変化する。

## (4)取組

以上の問題点・背景を踏まえ、現在二つの取組を実施している。一つ目は、「Android Studioによるアプリケーション開発」に向けての取組である。3年次10月下旬からのJavaアプリケーション開発授業（学校設定科目）の導入として、AWTを利用したJavaアプリケーションを作成し、基礎文法の確認をしている。(Swingのコンポーネントを利用したGUIプログラムは、科目「プログラミング」で学習済み。)その後、ボタンやテキストフィールドなどのGUIコンポーネントを利用するプログラム学習を通して、インタフェース、イベント処理の仕組み、コンテナ、レイアウトマネージャについて学ぶ。その理由は、UI（ユーザインタフェース）は、Androidアプリケーション作成において非常に大切な要素であり、レイアウト、アクティビティのライフサイクル、イベント処理の理解が不可欠であるからだ。その後、「Android Studio」による開発環境へ移行する。

プログラミング教育に求められるのは、「クリエイティブな発想力」と「何かを作りたいという意欲」である。プログラミングは、楽しいという体験が、難解なソースコードやアルゴリズムを理解しようとする探求心につながり、学びの原動力となる。「学びの連鎖」を育て、「論理的思考力、想像力」を引き出す仕掛けが必要である。

別の導入方法として、アメリカのMITが運営している「App Inventor」を活用し、Androidアプリケーション開発を実践することもできる。「App

Inventor」では、クラウド上の「App Inventor サーバ」と同期させることで、学校で作ったアプリケーションを、自宅のPCからでも、同じ環境を使って「修正」や「更新」といった作業が可能になる。最初から、プログラム記述を目指すのではなく、タイルプログラミング手法によりアプリケーション開発を行い、何ができるのかを教え、何ができないのかを実感させることができる点において有効である。

### 【Android Studio によるアプリケーション開発】

アプリケーションを開発するとき、操作するファイルは主に下記の3つである。

#### ①ソースコード

主に Java 言語で記述したプログラムで、プロジェクトの java フォルダ以下に配置する。

#### ②リソースファイル

画像、音声、表示する画面のレイアウトをリソースとして管理し、プロジェクトの res フォルダ以下に配置する。

#### ③設定ファイル

AndroidManifest.xml や build.gradle など、アプリケーションや開発の設定に関するファイル。

### 【簡易なアプリケーションの作成手順】

動作：画面の数字をタップし、正解時、数字が消える

#### ①画面レイアウトを作る

「レイアウト XML」ファイルを作成し、レイアウトを作る。画面表示する際に使う部品としては、「Layout」と、「View」があり、レイアウトの中にビューを配置する。大まかな設定を GUI で行い、細かい設定は、テキストで記述することができる。また、XML を用いず、Java ソースコード内にレイアウトを直接記述することも可能である。下図は Component Tree に GUI で、button を 4 つ配置し画面レイアウトを作成したものである。



#### ②ソースコードを記述する

プロジェクトウィンドウ内の「Project → app → src フォルダ → main フォルダ → java フォルダ」にある Java で記述されたプログラムに、乱数でボタンに数字設定する処理、タップ時のイベント処理のソースコードを記述する。

#### ③エミュレータ、実機で動作確認をする

Android SDK のエミュレータにより動作確認が可能であるが、立ち上がりに時間を要する。それは、ARM をエミュレートして動かしているためで、Android アプリケーションを開発する上で、大変非効率になる。これを解消するために「Genymotion」（ジェニーモーション）という VirtualBox（仮想化プログラム）を利用し動くエミュレータを活用する。純正の Android SDK のエミュレータと比べ、圧倒的に速く実行が可能になる。

Android Studio に Genymotion 利用のためのプラグインをインストールする。Genymotion により仮想デバイスを作成し、そこからエミュレータを実行することができる。下図は Android Studio のプロジェクトを Genymotion のエミュレータ上で実行したものである。



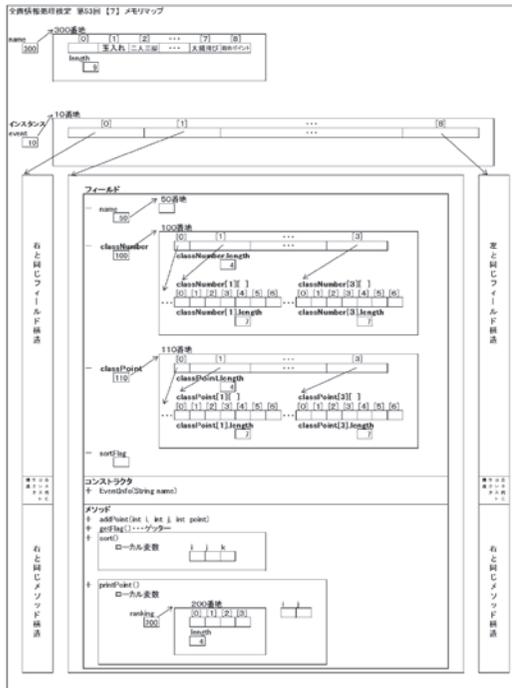
二つ目の取組は、授業改善、指導内容の標準化を目的とした「開発教材の共有化と校内研修の実施」である。

### 【開発教材の共有化と校内研修の実施】

私たち教師は、組織に属し、「個」ではなく、多くの人と関わり、支え合いながら職務を遂行している。教育活動には、「ビジョンをもち、それをメンバーに示し、臨機応変に実行していく姿勢」と、「周りを巻き込むため、コミュニケーション力により、チーム力を向上させること」が何よりも重要だと考える。

本校のデザイン、情報処理国家資格などの学校設定科目や、Java 言語、上級簿記資格対応科目においては、科目指導者がほぼ固定状態である。初めて授業を担当する者には、教材研究が大きな負担となっている。解決策として、本校教師による「デザイン」や「Java」学習会を週1時間の教科会内で企画・実施し、研修の動機付けとした。それを機に、自ら積極的・継続的に研修に取り組んだ教師や実習教員は、技術や知識を習得し、僅かではあるがチーム力は向上し教科指導レベルが標準化された。残念ながら翌年以降、教科会は放課後の設定となり、部活動など生徒指導のため、研修を継続することが困

難になった。そこで、新たな方法で教科指導レベルの標準化に取り組む必要が生じ、教師間で開発教材の共有化を図ることとした。その一つは、Java 授業提示用教材の共有である。内容は、情報処理検定 1・2級の範囲であり、「Javaの基礎、アルゴリズムとJava文法の関連、トレース」などである。また、作成教材を用いた校内研修を長期休業中、定期考査期間中に企画・実施している。下図は、作成教材の一例「1級プログラムのメモリマップ」である。



#### 4 可能性

プログラミング教育は、商業の学びの中で独立したのではない。科目「プログラミング」を他のビジネス情報分野科目と連携し、「実学」として深化させることが求められている。成果として「どのような力が身に付いたか」を評価するため、実際に、Androidアプリケーションを作成し、コンセプトや工夫した点などプレゼンテーションすることで言語活動を充実することも大切だ。作成したアプリケーションは、自分のWebサイトのみでなく、Google Playで公開することもできる。Google Playでは、アプリケーションを公開した後、インストールしたユーザーがレビューを行う。ユーザーのレビューは、永久に残るため、初期に低い評価がつくと後のダウンロード数に大きく影響する。公開できるクオリテ

ィであるかを「バグの有無」・「あらゆる端末での動作」・「操作の標準性」・「利便性」の観点で確認する必要がある。また、Google Playでは、アプリケーションの利用状況について解析機能が用意されており、インストールユーザー数、平均評価などを時系列で見ることが可能である。アプリケーション開発を通して、「情報分析」・「システム開発手法」・「デザイン」・「起業家精神」・「セキュリティ」・「法律」・「規範意識」・「倫理観」を学ばせることができる。

「プログラミング」を履修した生徒たちは、「論理的なものごとの考え方」や「最小の手間で正確な仕事をこなすための思考法」、「表計算ソフトウェアにおけるマクロプログラミング機能の理解」を学習の成果として挙げている。「Javaをはじめとしたプログラミング言語は、世界共通の言語であるため、将来どの国でも仕事ができる可能性が生まれた」と述べる者もいる一方で、就職活動を終えた3年生の中には、「職種に直結しない」とデメリットを挙げた者もいる。求人開拓は、今後の課題である。

#### 5 おわりに

プログラミングはテクノロジーの進歩とともに圧倒的に簡単になり、今やプログラムは世界中に溢れている。巷では、プログラミング塾が流行り、通塾する小学生が増加している。ビジュアル・プログラミング(Scratch)から始め、プログラム言語によるアプリケーション開発まで自由にこなす子どもたちが、今後ますます増加していくことは想像に難くない。一昔前まで教養としていわれていた「読み・書き・そろばん」が、「読み・書き・プログラミング」に取って代わられる日もそう遠くないかもしれない。

プログラミングとは一言でいえば「スマートフォンを含むコンピュータなどを思い通りに動かす方法」のことであり、どのように動かすかは、プログラミングを書く人に任されている。

商業教育における「プログラミング教育」は、3年間の完成を目指すのではなく、高等教育さらには、生涯教育の入口的役割を果たすものと位置付けることができる。プログラミング教育を通じて論理的思考力、想像力、コミュニケーション力を養い、日常生活、他の学習への応用につなげ、「知っている知識」から「使える知識」へと学びを深化させていくことを目指す必要がある。