

グラフィック出力を備えたブラウザベースの オブジェクト指向言語の軽量実行環境

鳥居 隆司*1・田村 謙次*2・安藤 明伸*3・杵淵 信*4・森 夏節*5
川崎 直哉*6・大岩 幸太郎*7・中野 健秀*8・藤尾 聡子*9・古金谷 博*9

Email: torii@sugiyama-u.ac.jp

*1: 梶山女学園大学 文化情報学部
*2: 中央学院大学 商学部
*3: 宮城教育大学 教育学部
*4: 北海道教育大学 教育学部
*5: 酪農学園大学 農食環境学群

*6: 上越教育大学 学校教育学部
*7: 大分大学 教育福祉科学部
*8: 愛知学院大学 商学部
*9: シンカーズ・スタジオ

◎Key Words プログラミング, オブジェクト指向, グラフィック出力, Web ベース

1. はじめに

近年、情報技術が急速に進化し、我々の生活に関わるあらゆる場面において活用されている。しかし、小型化したコンピュータは、コンピュータそのものが目に見える形で存在するのではなく、スマートフォンに代表される携帯情報端末や家庭電化製品にまで搭載されている。また、クラウド化されたそれは、携帯情報端末や住宅、自動車等に搭載されている様々なセンサからのデータをネットワークで結ばれた遠隔地において、いわゆるクラウドコンピューティングとして、演算処理を行い、ユーザにフィードバックするなどして利用されている。それらの情報とユーザをつなぐインタフェースもそれぞれの情報がメタファーとして表現されたアイコンなどを直接的に操作しやすいタッチパネルで構成されているため、情報システムがブラックボックス化されていることにさえ気づかず、後戻りもできない時期に来てしまった。

現代社会は、あらゆる産業が効率化され、集約化されており、日常生活で利用する様々なモノやサービスが、どこでどのように生産され、どのような流通によって、我々に届けられ使うことができるのか、実際にそのサービスは、どこで誰が行っているのか知ることが困難になってきて久しい。このことと同様に、コンピュータを構成する半導体や記憶装置などの機器がどんな材料から生産され、どのような技術によってハードウェアを形成できるのかの知識を持って活用している人は少ないし、集積回路をDIYで作成できる人⁽¹⁾は、ほとんどいない。さらに、情報そのものも、どこでセンシングされ、どの経路や仕組みを経て、どう処理され、どのように我々に届けられて利用できているのか正確に把握することも非常に困難である。また、センサやカメラ、SNS、電子商取引等で収集されたデータや画像等の情報は、増加し蓄積され続けるため、ビッグデータとして新たに活用されるようになる。そして、これら进行处理しているソフトウェアは、細分化されており、部分部分がそれぞれ永遠に改良され続け、バージョンアップを繰り返すため、仮に誤りがあったとし

ても、どの時点のどのソフトウェアの不具合なのかユーザが正確に判断することも極めて困難である。

これから将来に向かって、情報技術は、加速的に進化発展し、現在では、適用されていない機器やサービスに対しても活用されるようになり、我々が気づかないところに浸透し、人々は、どのような情報をどう選択することが適切なのかについて判断することすら困難になるのではないかと考えられる。

このような背景から、次期学習指導要領⁽²⁾⁽³⁾においては、プログラミングが、やや遅すぎたと思われるものの、コンピュータに意図した処理を行なわせることができる体験をしながら、時代を超えて普遍的に求められる力としてのプログラミング的思考を育むために、小学校段階で必修化される。そして、高等学校における学習指導要領⁽⁴⁾においても、「コンピュータとプログラミング」の項目の資質・能力（指導内容の構造）において、「プログラミングにより、コンピュータを活用する力、事象をモデル化して問題を発見したりシミュレーションを通してモデルを評価したりする力を育む」及び、「プログラミング」が明記される方向性が示されている。また、「問題の発見・解決に向けて適切かつ効果的にプログラミングしたり、モデル化やシミュレーションをしたりする力」とし、学習活動（課題設定）の例においても、ソフトウェア内部での情報処理の仕組みを扱う場合、「基本的な機能を実現するアルゴリズムについて考え、プログラムを作成するとともに最適化も行う」こととしており、具体的にプログラミングを行うこととなる。

2. プログラミング教育

最近、プログラミング教育⁽⁶⁾が、学校教育の現場、多くの民間団体や様々なワークショップ等で活発に行われているが、教養としてのプログラミング的思考と、産業的な意味でのプログラミング教育と分けて考えた方がよいかもしれない。情報産業に関わる人材育成としてのプログラミング教育としては、第4次産業革命に向け、AI, IoT, ビッグデータ, セキュリティ及び、

その基礎となるデータサイエンス等の人材の育成や確保は、当然必要であるし、情報機器や情報を活用したサービスが発達し、情報環境が現在よりさらに幅広く普及することになれば、当然、それらの情報機器の生産、メンテナンス、ソフトウェアの開発、設定等のための人材が必要となる。しかし、それらの人材の仕事は、次第にコンピュータによって自動化されるため、先進国としての存在感を維持するには、高度なレベルの人材が求められることとなり、大学院レベルにおいて、数理や情報の教育の強化を行う必要が出てくる⁽⁶⁾⁽⁷⁾。

一方、教養としてのプログラミング的思考としては、前述の高度なレベルの人材を育成するために幅広く多くの人々が基礎的な素養を身に付け、高度なレベルへと展開できるための裾野を厚く広くするという考えもある。また、社会にコンピュータが広く深く浸透したことから、日常的にコンピュータを使いこなすことで、自らの生活が便利になるので、単なる使い方ではなく、コンピュータの本質的な部分を知った上で使いこなすことは重要である。さらに、人の能力を補完するツールとして活用し、視野を広げ、人としての質を高めることも可能となる。たとえば、絵画として写実的に日常を描写することが技術的に必要であった時代から、写真技術が発明されたものの、写実的な絵画の生産の意味がなくなったわけではなく、新しい絵画の表現が生まれたように、モノづくりのためにプログラミングを活用すること、プログラミング技術を応用したメディア・アートとしての活用等によって、創造性や表現を行うこともできることになる。

次期学習指導要領において、小学校段階でプログラミング的思考を育むことは、「小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」の議論を経て、平成28年6月に「議論の取りまとめ」として公表され、その後、プログラミング的思考においては、教科等横断的な視点に立った資質・能力の育成として、児童の発達の段階を考慮した情報モラルを含んだ情報活用能力育成を図るものであるが、コンピュータや情報通信ネットワークなどの情報環境を整え、適切に活用した学習活動の充実を図るだけでなく、児童がプログラミングを体験しながら、コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動を行うものとして記述された。

中学校段階では、技術・家庭（技術分野）の情報の技術において、「生活や社会における問題を、ネットワークを利用した双方向性のあるコンテンツのプログラミングによって解決する活動を通して、次の事項を身に付けることができるよう指導する」とされ、その中で、「情報通信ネットワークの構成と、情報を利用するための基本的な仕組みを理解し、安全・適切なプログラムの制作、動作の確認及び、デバッグ等ができること」及び、「問題を見いだして課題を設定し、使用するメディアを複合する方法とその効果的な利用方法等を構想して情報処理の手順を具体化するとともに、制作の過程や結果の評価、改善及び、修正について考えること」と示されている。さらに、「生活や社会における問題を、計測・制御のプログラミングによって解決す

る活動を通して、次の事項を身に付けることができるよう指導する」とし、具体的内容として、「計測・制御システムの仕組みを理解し、安全・適切なプログラムの制作、動作の確認及び、デバッグ等ができること」及び、「問題を見いだして課題を設定し、入出力されるデータの流れを元に計測・制御システムを構想して情報処理の手順を具体化するとともに、制作の過程や結果の評価、改善及び、修正について考えること」と記述されている。

高等学校段階では、まだ、学習指導要領が確定していないが、平成28年8月に出された、教育課程部会の情報ワーキンググループにおける審議の取りまとめで、情報分野における科学的な理解を伴った情報教育を行っていく方向性であり、現行の「社会と情報」「情報の科学」の2科目の選択必修ではなく、仮称ではあるが、「情報I」を設け、その基礎の上に、問題の発見・解決に向けて、情報システムや多様なデータを適切かつ効果的に活用する力や情報コンテンツを創造する力を育む選択科目としての「情報II（仮称）」を設けるとしている。具体的には、「情報I（仮称）」は、情報と情報技術を問題の発見と解決に活用するための科学的な考え方を育成する共通必修科目であり、問題の発見・解決に向けて、事象を情報とその結び付きの視点から捉え、情報技術を適切かつ効果的に活用する力を育む科目で、「情報II（仮称）」は、発展的な内容の選択科目であり、「情報I（仮称）」において培った基礎の上に、問題の発見・解決に向けて、情報システムや多様なデータを適切かつ効果的に活用し、あるいは情報コンテンツを創造する力を育む科目である。

次期学習指導要領においては、「コンピュータとプログラミング」の項目の資質・能力（指導内容の構造）においては、「プログラミングにより、コンピュータを活用する力、事象をモデル化して問題を発見したりシミュレーションを通してモデルを評価したりする力を育む」と「プログラミング」が明記されており、「問題の発見・解決に向けて適切かつ効果的にプログラミングしたり、モデル化やシミュレーションをしたりする力」となっている。さらに、学習活動（課題設定）の例においても、ソフトウェア内部での情報処理の仕組みを扱う場合、「基本的な機能を実現するアルゴリズムについて考え、プログラムを作成するとともに最適化も行う」としており、具体的なプログラミングを避けて通ることができない。

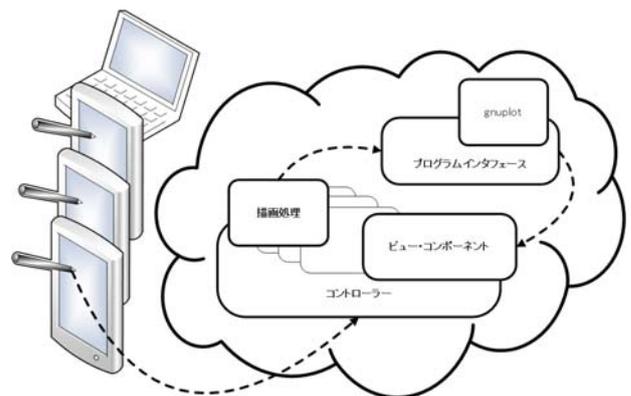


Fig.1 グラフィック描画の仕組み

3. グラフィック出力を備えたオブジェクト指向言語の実行環境の構築

我々は、これまで、オブジェクト指向型のプログラム言語として、Java 言語を選択し、Web ベースのコンパイル・実行が可能な環境をクラウドに構築⁹⁾してきたが、本研究では、さらに、グラフィック出力を可能とする機能を追加し、電子書籍リーダーとして e インクの携帯情報端末程度の性能であっても、構成主義的な学びも実現できる非常に軽量な実行環境を構築した。

グラフィック出力を行う場合、本実行環境としては、学習者が意図するかしないかに関わらず、描画領域を確定する必要がある。学習者がグラフィック出力を行った経験がない場合には、自らが指定した領域内におさまるように描画出力をさせるプログラミングを行うことは、容易でないことが多く、領域外に描画を行ってしまった結果、何も表示されない場合や、描画領域と比較して、出力図形の大きさがあまりにも小さく、なにも表示されないか、単なる点が表示されるのみの結果になることは多々ある。

Table 1 グラフィック描画の動作

グラフィック描画メソッド	動作
gt.move(double x, double y)	座標 (x, y) に移動 (ペンは上がった状態で、描画されない)。
gt.move(double x, double y)	ペンで座標 (x, y) まで描画 (ペンは下がった状態で、描画される)。
gt.fd(double s)	ペンのある位置からその時点の進行方向に向かって距離 s だけ前進。 s が負の値の場合には、後退する。
gt.bk(double s)	ペンのある位置からその時点の進行方向と逆に距離 s だけ前進。 s が負の値の場合には、前進する。
gt.lt(double deg)	その時点の進行方向から反時計回りに deg だけ回転。現在位置は変化しない。
gt.rt(double deg)	その時点の進行方向から時計回りに deg だけ回転。現在位置は変化しない。

このような場合においても、我々がこれまで開発してきたプログラミングの学習環境の在り方と同様で、プログラミング学習以前の問題が問題である。コンピュータの仕組みや働きを理解するために、コンパイル型のコンピュータ言語を選択し、その言語を記述しながら、学習を行う場合、プログラムをコンパイルして実行させる環境を学習者の利用するコンピュータに構築する必要がある。かつては、そもそも、コンピュータのディスプレイが高価であり、ディスプレイのないコンピュータでも学習はできるし、ノートにプログラムを文法に従って記述することで、プログラミングの学習などは可能であるかもしれないが、現在では、コンピュータを活用しながら、プログラミングを行うことが圧倒的に多いであろう。

初学者の場合には、特に自らの PC でプログラミングしたいプログラムを実行できる開発環境を構築することが困難な場合が多い。また、PC 教室の場合においても、教授する側が、多くの学習者の利用するコンピュータに開発環境を構築することは、それらのコンピュータに対するある程度の実行権限を学習者に与える必要もあり、不特定多数の学習者の別の環境への配慮を細かく行う必要もあり、大きな負担となる。

グラフィック出力の場合においても同様である。グラフィックを出力させるためには、コンピュータのグラフィック機能が活用できる環境を構築する必要があり、適切な環境を構築するには、コンピュータに関する基本的な知識が必要であるが、たとえば、構成主義的な考え方に基いて、タートルグラフィックを行うことのできる環境が構築されたとする。

しかし、文字入力のみならずと同様で、グラフィックを出力する命令そのものに誤りがない場合であっても、前述のように基本的に描画位置や出力図形の大きさの設定が、学習の障害になることが、初学者の場合に多く見られる。こうなってくると、プログラミングが理解できない原因として、その授業の構成主義的な考え方が、コンストラクティビズム⁹⁾なのか、コンストラクショニズム^{10) 11)}なのかという問題でもなく、グラフィックの概念が理解できないことでもなく、もちろん、文字入力でも、プログラミング的思考の問題でもないことになる。

本研究では、学習者がプログラミングそのものの学習を速やかに行うことができることを目的にしているため、グラフィック機能においても、描画領域の設定やグラフィック・ライブラリのインストールなどの初学者に困難な操作を行うことなく、試行錯誤しながら、ペンで絵を描くように図形を描くことで、少しでも構成主義的な学習に近いプログラミング学習を可能とする機能を実装した。

Fig. 1 に示すように、プログラムから描画オブジェクトが呼び出され、コントローラが、その実行結果として座標データを生成する。出力された座標データは、プログラム・インタフェースを経由して、gnuplot に伝達され、画像ファイルが生成される。そして、gnuplot によって生成された画像ファイルを用いて、ビュー・コンポーネントが実行結果の画像を生成する。

具体的なグラフィック描画の手順としては、本オブジェクト指向実行環境のグラフィック・クラスをインポートし、そのクラスのインスタンスを生成し、生成されたインスタンスに対して、グラフィック描画を指示する。このグラフィック描画メソッドは、基本的に、LOGO のタートル・グラフィックスと同様のインタフェースであり、Table 1 に示すように、絶対座標、相対座標のいずれかで動作を指定して用いることができる。

4. 実行環境でのグラフィック出力

本研究において、構築した環境で実行した結果を、Fig. 2, Fig. 3 に示す。インターネットに接続された Web ブラウザがあれば、すぐに Java 言語の学習が始められ、さらに、グラフィック出力も可能であるので、実行環境としては、極めて非力な環境である電子書籍リーダーに付属のブラウザ環境で動作させている。

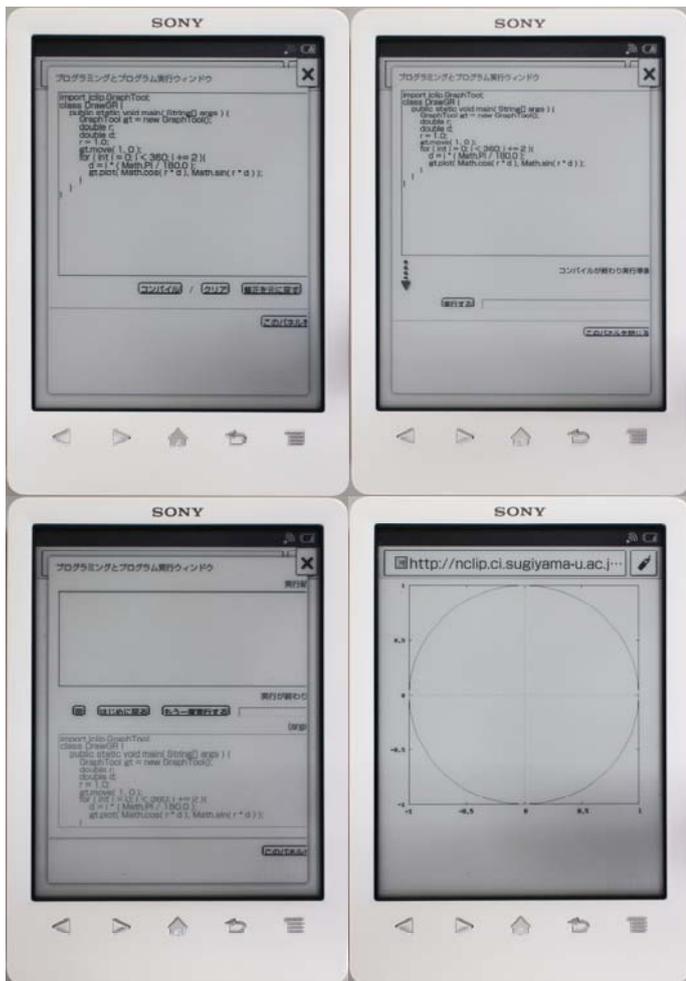


Fig. 2 グラフィック出力の実行, プログラムを入力し, コンパイルを行い, エラー等がなければ, 図ボタンをクリックすると図形が描画される

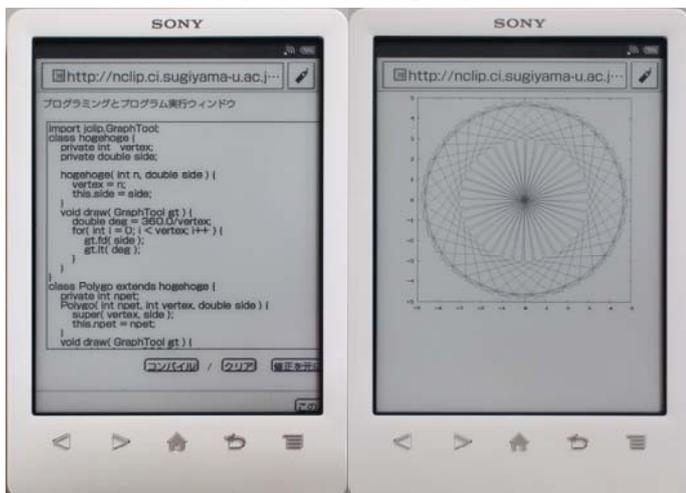


Fig. 3 辺の長さ 3.0 の正方形を 36 個, 1 度ずつ回転させながら描画した例

5. 考察

これまで述べてきたように, プログラミングをコンピュータ上で行う場合には, 実行環境を整える必要があるが, 現実には, 様々な設定が必要であり, 初学者にとっては, 不成功に終わることも多くプログラミング学習以前の問題を解決できずにプログラミング学習を行うことができない結果となる。本研究では, これ

らの問題を解決するとともに, グラフィック出力機能を追加し, 従来のプログラミング学習に, 構成主義的な学びも実現できる非常に軽量な実行環境を構築し, 大学でのプログラミング演習の授業やプログラミングのワークショップ等において, 数十人のユーザで同時に Java 言語のプログラミングを体験してもらったが, 問題もなく学習をすることが可能であった。現在, クラウドに構築された本実行環境は, インターネットに公開しており, 不特定多数のユーザが利用できる状況にしている。現時点では, セキュリティ面やシステムのレスポンスも含めて問題なく動作しているので, 本研究で構築した仕組みは, プログラミングを行う環境として十分活用できるものと考えられる。

6. おわりに

次期学習指導要領では, プログラミング的思考力へのステップとして, コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動が必要とされ, 探究的な学習の過程に適切に位置付くようにするなどの具体的な記述がされている。プログラミングにつながる論理的思考を行うためには, 必ずしも, コンピュータでのプログラミング環境が必要ではないが, 初期設定等を行うことなしに, コンパイラ型のプログラミング学習が可能となる本研究によって構築された実行環境が, プログラミング的思考の学びを支援できることを願っている。

謝辞

本研究の一部は, JSPS 科研費 24501069, 25381242, 15K00935 及び 16K04748 の助成を受けたものである。

参考文献

- (1) 秋田純一, 集積回路のコモディティ化の意義とそれに向けた「誰でもチップ試作」の試み, 電気情報通信学会誌, Vol.99, No.9, 2016, pp.911-915.
- (2) 小学校学習指導要領, 文部科学省, 2017.
- (3) 中学校学習指導要領, 文部科学省, 2017.
- (4) 情報ワーキンググループにおける審議の取りまとめについて (報告), 中央教育審議会初等中等教育分科会教育課程部会 情報ワーキンググループ, 2017.
- (5) プログラミング教育実践ガイド (文部科学省), 一般社団法人ラーン・フォー・ジャパン, 平成 27 年 3 月 26 日.
- (6) 世界最先端 IT 国家創造宣言 (改訂), 内閣府, 平成 28 年 5 月.
- (7) プログラミング人材育成の在り方に関する調査研究報告書, 総務省, 2016.
- (8) 鳥居隆司, 田村謙次, 安藤明伸, 杵淵信, 森夏節, 川崎直哉, 大岩幸太郎, 中野健秀, 藤尾聡子, 古金谷博, ブラウザベースのオブジェクト指向言語実行環境, PC カンファレンス論文集, 2016, pp.7-10.
- (9) Ernst von Glasersfeld, Radical Constructivism – A Way of Knowing and Learning, The Falmer Press, 1995.
- (10) 世界制作の方法, Nelson Goodman (著), 菅野盾樹 (翻訳), ちくま学芸文庫, 2008.
- (11) Kafai, Y., Resnick, M., Constructionism in Practice-Designing, Thinking and Learning in A Digital World, LEA, 1996.