

Dolittleのオンラインプログラミング環境の開発

本多 佑希¹ 大村 基将^{1,2} 長 慎也³ 久野 靖⁴ 並木 美太郎⁵ 兼宗 進¹

概要 : Dolittle は教育用に設計されたオブジェクト指向言語である。Dolittle では、クラスとオブジェクトの関係のような高度な概念の理解が不要なことから、初学者でもオブジェクトを扱うことができる。Doittle は Java で開発されており、さまざまな OS 上で動作するため、多くの授業で使われてきた。本研究ではタブレットを含む多くの環境で動作するように JavaScript で動作する Dolittle の開発について現状を報告する。

キーワード : Dolittle, JavaScript, オンラインプログラミング

1. はじめに

Dolittle^{1,5)} は教育用に設計された、オブジェクト指向言語である。Dolittle では、オブジェクトを扱う際にクラスとオブジェクトの関係のような概念を理解する必要が無い。そのため、初学者がオブジェクト指向を学ぶ際に適している⁶⁾。

従来の Doittle (以下, Java 版) は Java で開発されており, Windows/Mac/Linux などのさまざまな PC 上で動作する。Java を同梱しているパッケージでは, ファイルを展開するだけで管理者権限を必要とせずに使うことができる。(ブラウザ上で動作する Java アプレット版も存在するが, OS に Java がインストールされていることが前提であるため, 利用は減っていると思われる)

今回提案する JavaScript 版の Dolittle (以下, DolittleJS) はブラウザ上で動作し, ファイルの展開を含めたインストール作業が不要という特徴を持つ。また, 作成したプログラムを PC だけでなくタブレットやスマートフォン (以下, モバイルデバイス) のブラウザ上で実行することができる。(将来的にはタブレットなどのデバイス上でのプログラム記述にも発展できる可能性がある)

本研究で開発している DolittleJS は, 言語の基本仕様の実装を行い, 現在一部のグラフィックスと GUI 部品などの対応を進めているところである。今後は Java 版との互換性を保ちつつ, JavaScript との連携やモバイルデバイスに搭載されたセンサ類へのアクセスなど, 新たな可能性についても検討を進めていきたいと考えている。

2. Dolittle の設計思想

Dolittle は, ユーザがオブジェクトを意識せずにオブジェクト思考を学ぶことを目標にしている。例えば, C++ や C などを学ぶ際には, 最初は printf や cout といったオブジェクトとは無関係な出力命令から始まる。クラスを設計し, インスタンスを

¹ 大阪電気通信大学
Osaka Electoro-Communication University

² 静岡大学
Shizuoka University

³ 明星大学
Meisei University

⁴ 筑波大学
University of Tsukuba

⁵ 東京農工大学
Tokyo University of Agriculture and Technology

生成するまでには多くの学習時間を要する。

Dolittle の多くの教材⁵⁾ や論文^{2-4,7-9)} で、ユーザが最初に実行することになるプログラムの先頭は次の通りである。

かめた=タートル!作る。

このプログラムを実行すると、画面にタートルが表示される。このタートルはオブジェクトであり、ユーザは最初に実行するプログラムからオブジェクトに触れることになる。付ける名前は文献によって差異はあるが、タートルオブジェクトを生成して名前をつけることから始まる。

おおよそ、次にこのかめたに前に歩くように命令をする。

かめた=タートル!作る。

かめた!10 歩く。

この2行のプログラムで、ユーザはオブジェクトを意識することなく、かめたというオブジェクトに対してメソッドを実行するというオブジェクト指向を体験する。タートルを歩かせる際に、座標系のような知識を必要としないことも重要である。

このように、「オブジェクトを意識することなく、オブジェクト指向を学べるように設計する」ことを目標としている。

3. DolittleJS の設計

DolittleJS では、以下のことを目標として開発を進めている。

3.1 Dolittle の設計思想に基づいた設計

2章で述べたように、Dolittle は「オブジェクトを意識することなく、自然とオブジェクト指向を学べてしまう」ように設計されている。DolittleJS で提供するオブジェクトも、この Dolittle の思想に基づいたものにする必要がある。

生徒にオブジェクトという概念を理解させる必要がなく、オブジェクトに対してメソッド実行やカスケードを行うことを、自然と理解させてしまうようなオブジェクトを設計することを目標としている。

3.2 手軽に Web ブラウザから実行できる

プログラミングを学ぶにあたり、Web ブラウザから実行できることは、有用である。DolittleJS は、Web ブラウザを開くだけで手軽ですぐにプログラミングを学べることを目標に設計した。

ローカルにインストールする必要がなく Web ブラウザ上で動作する。また初等中等教育の授業では、タブレットを使用する環境が増えつつある。そのため、初中等教育でのタブレットでの動作も視野に入れ、PC だけでなくタブレットやスマートフォンなどマルチプラットフォームで快適に動作させることを目標としている。

3.3 モバイルデバイスでの活用

ネットワークに繋がっている Web ブラウザ上で動作することで使えるようになる機能が多数存在する。DolittleJS ではタブレットもターゲットにしているため、タブレットだからこそ可能な機能などの設計も行っている。

タブレット上で動作することにより、タブレットに備え付けられたセンサにアクセスすることが可能となった。センサにより値を計測し、それを読み取ることによってプログラムの動作を変えることができる。

4. DolittleJS の実装

4.1 実行モデル

Dolittle の言語処理系を JavaScript で実装するにあたり、本研究では Dolittle で書かれたプログラムを JavaScript にコンパイルし、JavaScript として実行する方式を採用した。図1に、DolittleJS の言語処理系を示す。プログラムは全て Web ブラウザ内で実行される。Run ボタンが押されると、テキストエリアに入力された Dolittle のプログラムが JavaScript のプログラムにコンパイルされる。コンパイルされた JavaScript のプログラムが、DolittleJS の実行画面で実行される。

4.2 開発状況

現在は言語の基本仕様の実装が完了し、タートルグラフィックス、GUI 部品などの機能の実装

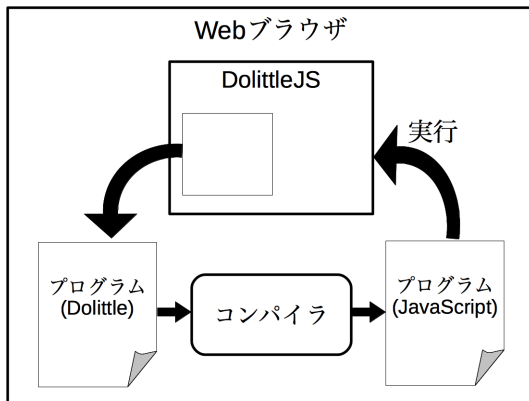


図 1 DolittleJS の実行モデル

```

    かめた=タートル!作る。
    時計=タイマー!作る。
    時計!「かめた!10 右回り」実行。
    時計!待つ。
    かめた!100 歩く。
  
```

図 3 待つメソッドの使用例

```

    かめた=タートル!作る。
    時計=タイマー!作る。
    時計!「かめた!10 右回り」実行。
    時計!「かめた!100 歩く」終了時実行。
  
```

図 4 終了時実行の使用例



図 2 DolittleJS の実行画面

を進めている。実行環境としては、Mac OSX の GoogleChrome 及び Firefox, iPhone の safari での動作を確認している。

図 2 に、開発中の DolittleJS の画面を示す。プログラムは、画面左側のテキストエリアに入力する。Run ボタンを押すと、テキストエリア内に入力されたプログラムが実行され、画面右側に実行結果が表示される。エラー表示を含む開発環境については今後も検討を行っていく予定である。

現在対応しているオブジェクトは「数値」「文字列」「真偽値」「配列」「タイマー」「ブロック」などの基本オブジェクトの基本機能と、「タートルグラフィックス」「ボタンなどの GUI」などである。

Java 版の機能のうち、音楽演奏, Dolittle 間の P2P 通信, ロボット等の外部機器との通信機能をどこまで実現できるかは今後も検討を進める予定である。

4.3 互換性の課題

DolittleJS では基本機能については Java 版との互換性を重視しているが、一部に現時点で互換性を実現できていない部分が存在する。ここではタイマーオブジェクトを例に説明する。

Java 版のタイマーオブジェクトは Java のスレッドの機能で実現されていたが、JavaScript にはスレッドが存在しないため、完全な互換性を実現することは容易ではないことがわかっている。

タイマーオブジェクトの待つメソッドは、タイマーの実行が終わるまで実行を止めるというメソッドである。しかし、JavaScript の setInterval 関数でこれを実現することは、非常に困難である。そのため、完全な再現をするのではなく、待つメソッドを使用した際と同等の動作をすることができる別のメソッドを提供することで、対応を行った。

終了時実行メソッドは、timer の実行が終わった際に実行される関数を登録するメソッドである。この終了時実行メソッドを使用して、タイマーオブジェクトの待つメソッドと同等の動作を実現した。図 3 に、Dolittle の本⁵⁾で紹介されている待つメソッドの使用例を示す。このプログラムはタートルが右に回るタイマーの実行が終わるのを待ってから、歩くプログラムである。終了時実行メソッドを使用すると、このプログラムは図 4 のように書くことができる。待つメソッドの対応はコストが大きいので、このように同等の動作をするメソッドを提供することによって対応した。

4.4 開発を進めている機能

現在開発を進めている機能と、今後開発したい機能を報告する。

4.4.1 Web サービスとの連携

DolittleJS は、ネットワーク上の WebAPI にアクセスして天気を取得したり、地図データを取得したりということが可能である。これにより、天気によって結果が変わるプログラムを作成することができるようになる。DolittleJS では、この WebAPI にアクセスして Web サイトからデータを取得する。

4.4.2 Web ページの作成

一般的に、動的に結果が変わる Web ページを作成するには HTML+JavaScript の 2 つを学ぶ必要があり、学習コストが大きい。しかし DolittleJS では、Dolittle の学習のみで動的な Web ページを作成することもできるようになる。

DolittleJS では、HTML のタグを出力するオブジェクトによって Web ページを作成することができる。また、DolittleJS は JavaScript として動作することから、HTML と JavaScript の機能を連携し、動的に結果が変わる Web ページを作ることもできる。次に、テキストエリアを生成する例を示す。

テキストエリア！作る。

このテキストエリアオブジェクトに対して、高さや幅などの設定を行う。次に、高さと幅を設定する例を示す。

入力＝テキストエリア！作る。

入力！15 幅 3 高さ。

4.4.3 モバイルデバイスのセンサ等へのアクセス

DolittleJS では、タブレットを視野に入れていることからタブレットの加速度センサ (以下、傾きセンサ) を利用して傾きを取ることができる。これにより、傾きを計測することが可能になる。傾きセンサは、イベントハンドラのような動作を行う。生徒にイベントハンドラについて理解させるのは困難であるため、意識せずに容易に値を取得できるよう、設計を行う必要がある。傾きセンサは、作るメソッドにより有効になる。

傾きセンサ！作る。

値を取得するためには、次のようにイベントハンドラを定義する必要がある。

傾き＝傾きセンサ！作る。

傾き：動作＝「| x y | ...」。

ボタンの動作定義文と文法は同じであるが、ボタンの場合はユーザが自ら押した場合の動作だったのに対して、傾きセンサの場合はセンサが関数を呼び続けることになる。このように、ボタンの動作と傾きセンサの動作は、文法は同じだが動作は異なる。この機能をどう生徒に理解しやすいものにするか、議論が必要である。

また、タブレット、スマートフォンの GPS 機能により、現在地を取得することができる。GPS は、作るメソッドを実行されることにより有効化される。

GPS！作る。

GPS オブジェクトにより、現在地の緯度と経度を取得することができる。

現在地＝GPS！作る。

現在地！緯度？。

現在地！経度？。

このように、GPS オブジェクトから緯度と経度を取得することができる。

ただし、GPS 機能を有効化する際にはポップアップで位置情報の通信を許可するかどうかのポップアップが表示される。授業で使用する際、生徒が Web サービスのポップアップを許可することに慣れてしまうのは問題がある。そのため、今後 GPS オブジェクトはどのように扱うか、議論が必要である。

4.5 自己拡張

現在 DolittleJS のオブジェクトは、JavaScript で書かれている。予め用意された Dolittle のプログラムを、ライブラリとして読み込んで実行することができない構造になっている。今後はこの問題の改良を行いたい。ライブラリとしてプログラムを提供したいことも多く、提供されている Dolittle のライブラリを読みたい場面も予想される。そのため、Dolittle で書かれたプログラムを予め読み込み、有効化する機能の実装を検討している。

5. おわりに

実行環境が不要な, Web ブラウザ上で手軽に Dolittle に触れることができるオンラインプログラミング環境を開発した. タブレットやスマートフォンを対象とした機能を追加することにより, 初中等の授業での活躍が期待される. 本研究の Dolittle の開発が進むと, より Dolittle は授業利用に適した言語になる. 今後も Dolittle の改良に努めたい.

謝辞 本研究は, 科学研究費補助金(基盤研究(C) 25350214)の補助を受けています.

参考文献

- [1] プログラミング言語「ドリトル」.
<http://dolittle.eplang.jp/>
- [2] 兼宗進, 御手洗理英, 中谷多哉子, 福井真吾, 久野靖: 学校教育用オブジェクト指向言語「ドリトル」の設計と実装, 情報処理学会論文誌, Vol.42, No.SIG11, pp.78-90 (2001).
- [3] 中谷多哉子, 兼宗進, 御手洗理英, 福井真吾, 久野靖: オブジェクトストーム: オブジェクト指向言語による初中等プログラミング教育の提案, 情報処理学会論文誌, Vol.42, No.6, pp.1610-1624 (2002).
- [4] 兼宗進, 中谷多哉子, 御手洗理英, 福井真吾, 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.SIG13, pp.58-71 (2003).
- [5] 兼宗進, 久野靖: ドリトルで学ぶプログラミング [第2版] (2011).
- [6] 兼宗進, 久野靖: プロトタイプ階層を持つ教育用オブジェクト指向言語「ドリトル」, コンピュータソフトウェア, Vol.28, No.1, pp.43-48 (2011).
- [7] 西野洋介, 兼宗進, 早川栄一: 高等学校産業科における3学年同時のドリトルによるプログラミング実習. 情報教育シンポジウム 2010 論文集, 2010 巻, 6 号, pp.23-28 (2010).
- [8] 兼宗進, 中谷多哉子, 井戸坂幸男, 御手洗理英, 福井真吾, 久野靖: 教育用オブジェクト指向言語「ドリトル」による授業実施とその評価, 情報教育シンポジウム 2002 論文集, 2002 巻, 12 号, pp.229-236 (2002).
- [9] 兼宗進, 久野靖: 未来のプログラム好きを育てる, 第51回プログラミングシンポジウム予稿集, 2010 巻, pp.65-72 (2010).