

# オブジェクト指向プログラミング教育支援アプリケーションの開発

小清水 誓太<sup>†</sup> 相馬 侑弥<sup>†</sup> 齋藤 菜月<sup>†</sup> 高野 辰之<sup>††</sup> 小濱 隆司<sup>†††</sup> 宮川 治<sup>†††</sup>

東京電機大学情報環境学部情報環境学科<sup>†</sup> 関東学院大学工学部<sup>††</sup> 東京電機大学情報環境学部<sup>†††</sup>

## 1 はじめに

プログラミングの概念の一つであるオブジェクト指向は、モジュール化やプログラムの再利用性などを考慮したシステム設計をするうえで重要である [1]。しかし、オブジェクトやオブジェクト間の関連などの抽象的な概念を正しく捉え、理解するのは難しい。そのため、プログラミングの授業ではUML (Unified Modeling Language) を利用した説明が試みられている。UML は、クラス構造やオブジェクトの関連などをグラフィカルに表現できる。

しかし、ソースコードの静的な関連を表現するオブジェクト図では、ソースコード実行に伴うオブジェクトの生成や関連が生じる順番、そのタイミングを表現することは難しい。

そこで、本研究ではオブジェクト図の変化がソースコードの実行に伴って表現されることを目的としたアプリケーションの開発をする。

## 2 アプリケーション概要

本アプリケーションは、一斉学習で教授者がオブジェクト図とソースコードを対応させて説明する際に使用する。

本アプリケーションに表示するオブジェクト図は、実行するソースコードの一行ごとに対応して変化する。また、オブジェクトの配置をドラッグで変更することも可能である。

図1に本アプリケーション構成を示す。本アプリケーションは説明画面、リモコン、サーバの3つで構成される。説明画面は学生にソースコードとオブジェクト図を表示し、教授者がリモコンを操作することで説明画面のソースコードの実行している行などを変更することができる。また、サーバは説明画面とリモコン間の操作情報の中継やソースコードの解析などを行っている。

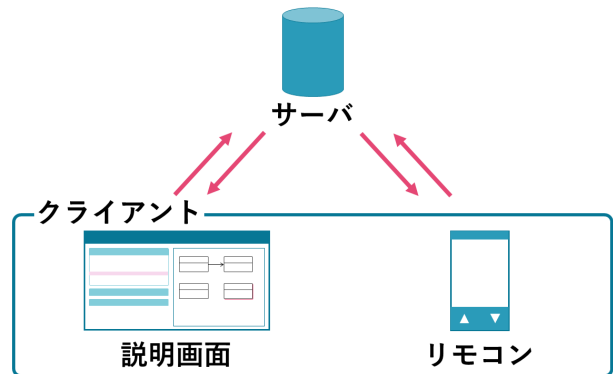


図1: アプリケーション構成

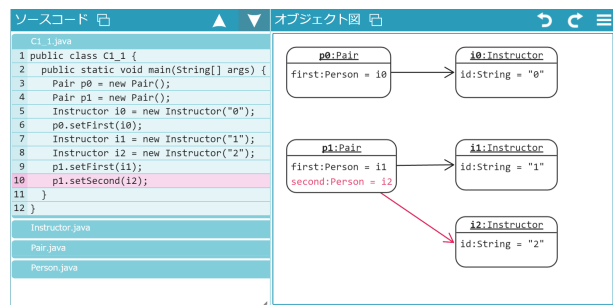


図2: 説明画面

## 3 アプリケーション構成

本アプリケーションのクライアントサイドはHTMLとJavaScriptで開発し、サーバサイドはJavaを用いて開発した。また、リモコン画面はスマートフォンによる利用を想定している。

### 3.1 説明画面

図2の説明画面は左側にソースコード、右側にオブジェクト図を表示している。また、本アプリケーションで使用されるオブジェクト図は教育上の強調や拡張を行っている。

オブジェクト図は「オブジェクト名: クラス名」で記述される上段と「属性名=値」で記述される下段で構成されるオブジェクトと、オブジェクト間の関連を表す矢印で表現される。

オブジェクトはそれぞれの位置をドラッグで変更することができ、オブジェクトの関連を表す矢印は、オ

Developing an Assist-Tool for Teaching Object-Orientated Programming  
<sup>†</sup>Seita KOSHIMIZU, Yuya SOMA, Natsuki SAITOH  
 School of Information Environment, Tokyo Denki University  
<sup>††</sup>Tatsuyuki TAKANO  
 College of Engineering, Kanto Gakuin University  
<sup>†††</sup>Takashi KOHAMA, Osamu MIYAKAWA  
 School of Information Environment, Tokyo Denki University

プロジェクトの位置に対応して描画される。

ソースコードは実行している行の色を変えて表示される。上部に配置したボタンをクリックすると画面上の実行している行が変化し、その行に対応するオブジェクト図が表示される。この行でオブジェクトが新たに生成された場合、そのオブジェクトに「new」マークが付加されオブジェクト図の色付けにより強調される。

また、オブジェクトの属性が変化し新たに関連が発生した場合には、オブジェクト間の関連を表す矢印と、該当部分の属性を表す文字列が色付けにより強調される。

この操作により、一行ごとのオブジェクト図の表現が可能となる。

### 3.2 リモコン

図3にリモコン画面を示す。リモコン画面には、オブジェクト図を配置し、画面下部にソースコードを進めるボタンと戻すボタン(1)、画面上部に説明画面の表示の切り替えボタン(2)と操作を元に戻すボタンと操作をやり直すボタン(3)を配置している。

リモコン画面のオブジェクト図には、次の行で生成されるオブジェクトやリンクが薄く表示される。そして、教授者がソースコードを一行ごとに説明する際、その行までに生成されたオブジェクトやリンクは濃く表示される。これは、教授者にあらかじめオブジェクト図のそれぞれのオブジェクトの位置を確認させるためである。また、説明画面で表示する前にオブジェクトの位置を変更できる。

進めるボタンを押すと、次の行まで実行された状態のオブジェクト図が表示され、説明画面にも反映される。同様に、戻すボタンを押すと、前の行まで実行された状態のオブジェクト図が表示される。

画面の表示の切り替えボタンは、説明画面のオブジェクト図やソースコードの表示を全画面にすることが可能である。

操作を元に戻すボタンと操作をやり直すボタンは、オブジェクトの配置を変更する操作を元に戻したり、その操作をやり直したりすることが可能である。

### 3.3 サーバ

サーバはソースコードの保存と解析を行う。また、説明画面とリモコン画面との情報を中継し、状態の同期を行う。

現在の実装手法について説明する。説明画面には教授者がアップロードしたソースコードを利用する。アップロードしたソースコードはサーバでコンパイルをされ、コンパイルが成功したものを JPA (Java Persistence API) を利用してデータベースに保存をしている。

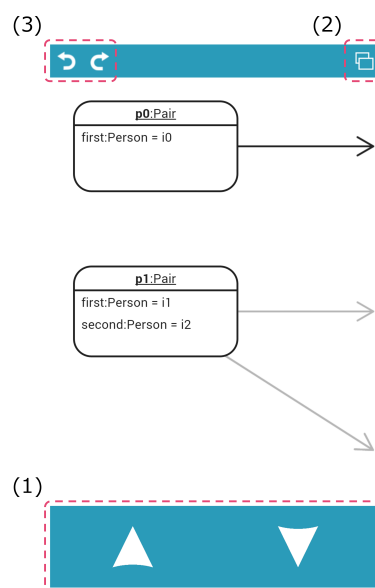


図3: リモコン画面

ソースコードの解析では JDI (Java Debug Interface) を用いて解析し、その解析結果を XML に書き出して利用する。

説明画面とリモコン画面の状態を同期するための通信に、SSE (Server Sent Events) を利用した。

## 4 考察

オブジェクト指向プログラミングにおいて、静的な関連を表現するオブジェクト図ではソースコードの実行に伴うオブジェクト生成や関連を表現するのは困難であった。本アプリケーションを使用することで、オブジェクト図を用いてソースコードのステップごとの状態を視覚的に表現することが可能となり、よりわかりやすい説明に向けたサポートが期待できる。

## 5 まとめ

今回開発したアプリケーションはソースコードからオブジェクト図を自動生成することで、ステップごとの変化を視覚的にわかりやすく示せるようになった。今後は学生がオブジェクトの生成や関連が生じる順番やタイミングを理解することができるか検証する。

## 参考文献

- [1] 松浦 佐江子: 実践的ソフトウェア開発実習によるソフトウェア工学教育, 情報処理学会論文誌, Vol.48, No.8, pp.2578-2595 (2007) .