

メディアデザイン学科における 「ビジュアル・プログラミングⅢ」の教育内容及び方法と実践

Contents and Methods of Training and Practices of Visual Programming III in the Department of Media Arts and Design

小松隆行*

Takayuki Komatsu

概要

本学メディアデザイン学科の科目「ビジュアル・プログラミングⅢ」の内容、方法、そして実践について報告する。プログラミング言語 Java⁽¹⁾の関連3科目の中で最後に開講され、オブジェクト指向を初めて学習するものである。初学者にとってその概念の理解がしにくく、その考え方や特徴を理解した前提でプログラミングしなければならないことから、講義と演習において様々な工夫を行っている。オブジェクト指向の理解を助けるために、Java プログラミングだけではなく、ブロックプログラミング言語でのゲーム制作を取り入れたり、講義内容と同様の動画を作成し学内教育用グループウェアで配信したりすることも行っている。

1. はじめに

本学の未来デザイン学部メディアデザイン学科では、専門教育科目系列の選択科目として2年生後期に科目「ビジュアル・プログラミングⅢ」を半期15回で開講している。この科目はメディアデザイン学科のディプロマポリシー(D) 情報技術の獲得、およびそれらの社会や自然に及ぼす影響・効果に関する理解力、さらに技術者やデザイナーとして社会に対する責任を自覚する能力(技術者倫理)、に対応している。

この科目では、オブジェクト指向の概念と Java プログラミング技法の基本を修得することが目標であり、以下の4項目が具体的な達成目標である。
①オブジェクト指向の概念を理解し、オブジェクト指向の基本であるクラスとメソッドの基本的な概念を修得し、クラスやメソッドを定義できるようになる、②Java プログラミングのための統合開発環境である Eclipse の使用方法を習得する、③Java プログラムの仕組みを理解し、一般的な逐次処理、繰り返し処理、選択処理とオブジェクト指向型プログラミングの技法を組み合わせることで自由にプログラムを作成できる、④クラスの使用法を理解し、これらを組み合わせることにより実用的かつ有用なプログラムを作成することができる、である。

2. 教育内容と実践方法

授業の方法としては、講義はパワーポイントによる説明と各自のノートPCによるプログラミング演習が主体となる。この科目に先立ち、1年生後期に「ビジュアル・プログラミングⅠ」、2年生前期に「ビジュアル・プログラミングⅡ」を開講している。「ビジュアル・プログラミングⅠ」ではプログラミングの基礎を Java で修得する。「ビジュアル・プログラミングⅡ」では Java アプレットを使ったプログラミングを修得する。本科目の「ビジュアル・プログラミングⅢ」では、オブジェクト指向の考え方とそのプログラミングを修得することになっている。しかしながら、十分に Java を修得しているとは言えない学生も多いため、プログラミングしやすくするための工夫と、オブジェクト指向を理解しやすくするための工夫を行っている。

授業内容と実践内容は、以下の通りである。第1週から第4週までは、ブロックプログラミングを行う。使用するのは、ブロックプログラミングのサイト Code Studio⁽²⁾と MIT が開発したブロックプログラミング言語 Scratch⁽³⁾である。タイピングをほとんどせずにプログラミングでき、ビジュアルな実行結果やゲーム作りなどで、楽しみながら容易にプログラミングを理解できる内容になっている。

この科目では、Scratch の教材を授業用に多数作成して演習で使用している。図1は、文章(仕様書)の内容を実現するプログラムを、完成形のプログラ

* 北海道科学大学未来デザイン学部メディアデザイン学科

ムのブロックをばらした状態から作成する, すなわち操作としてはブロックを組み上げるだけ教材の例である. 本科目を履修するまでに, Java の授業を 1 年間履修してきてはいるものの, その基本をほとんど理解していない学生も多い. プログラムの意味もあやふやな学生もいるが, Scratch での演習により, プログラムの意味とアルゴリズムの基本をようやく理解する学生もいる.

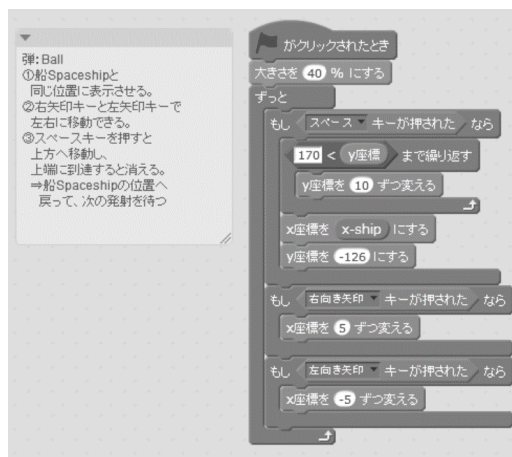


図 1 仕様文書からの Scratch プログラム作成の例

また, 「ビジュアル・プログラミングⅠ」と「ビジュアル・プログラミングⅡ」では, フローチャートを学んでいないので, 図 2 に示すようにフローチャートを提示して, それを基にしたブロックプログラムを作成する教材も作成し演習させたり, 図 3 のように Java の基本的な命令を提示して, それと同じ機能のブロックプログラミングを行わせる教材も作成し, 演習させたりしている.

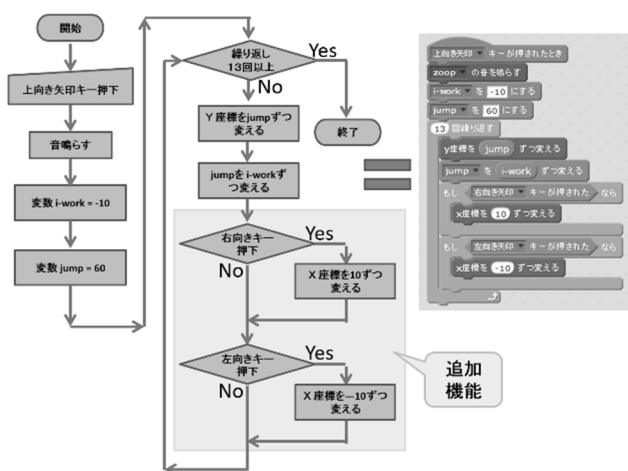


図 2 フローチャートによる Scratch の処理の解説

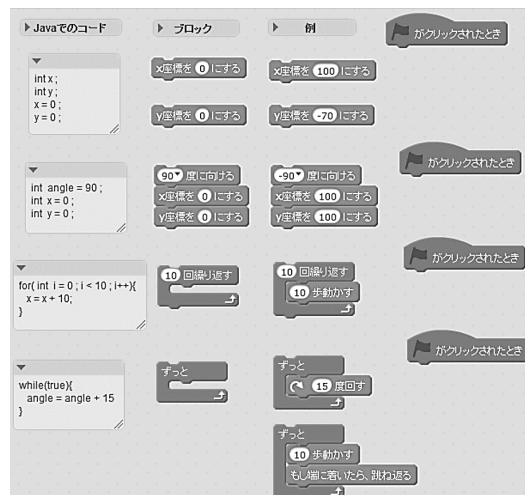


図 3 Java と Scratch を対応させた演習教材

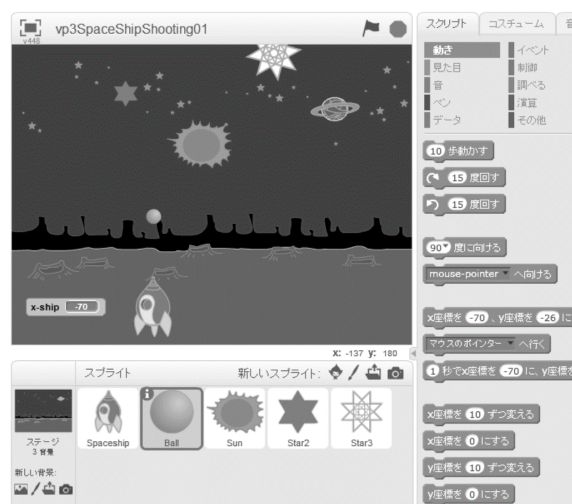


図 4 演習用 Scratch 教材：シューティングゲーム

図 4 はシューティングゲームの例である. 素材の配置や一部の動きなどの基本的な部分は作成して配布する. 要求される機能や学生各自の独自部分を追加する課題となっている. ブロックプログラミングという簡易なインターフェースの言語であるが, オブジェクト指向的な考え方の例えを説明するのに適していると考えられる. すなわち, 素材 (スプライト) はクラスやインスタンスとして解釈できると解釈できる. この考えを利用して, カプセル化などを説明すると, オブジェクト指向の概念や考え方の理解の助けになる. Scratch を例にして解説しながら Java のオブジェクト指向プログラミングの演習を進めてゆく (図 5).

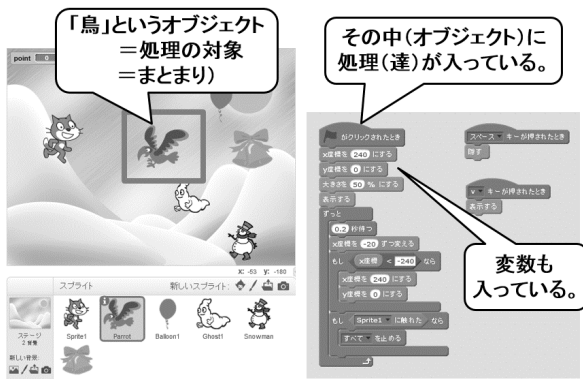


図5 Scratchを使ったオブジェクト指向の考え方

Java のオブジェクト指向プログラミングに先立ち、少しだけ Java の基本を復習させる (図 6)。作るべきプログラムロジックの仕様をコメントとして入れ、例となる部分やテンプレートなどを用意した 1 つのプログラムとして提供し、復習しやすくしている。

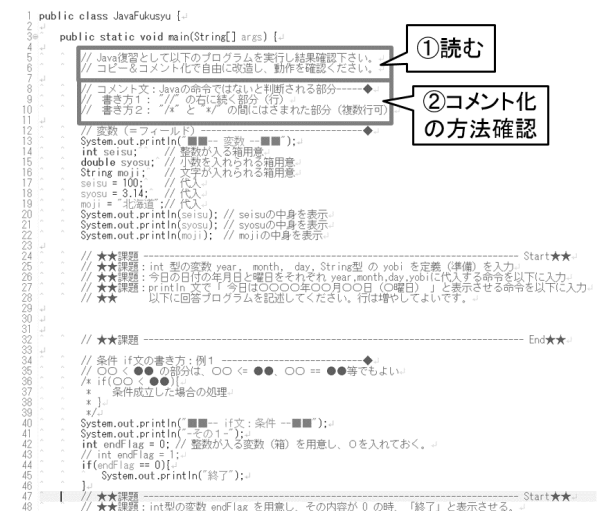


図6 復習用の課題 Java プログラム

教科書として「やさしい Java 第 6 版」⁽⁴⁾ を使用しているが、例題や問題のプログラム作成は、図 7 のような Java の基本的な命令のテンプレートをひとつひとつクラスにしたパッケージを用意し、これらを統合開発環境の Eclipse にインポートさせ、これらを必要に応じてコピー＆ペーストしてプログラム作成できるようにしている。これにより、Java の命令の書き方の記憶が曖昧である学生や、タイピングが苦手な学生でも効率よくプログラム作成できる。

また、初学者の域を出ない学生も多いことから、プログラミングはアジャイル的なコーディングをさせるようにしている。図 8 に示すように、基本と

なる簡単なプログラムに、機能を一つ一つ追加するなどして徐々に最終目的のプログラムにしてゆく形式である。

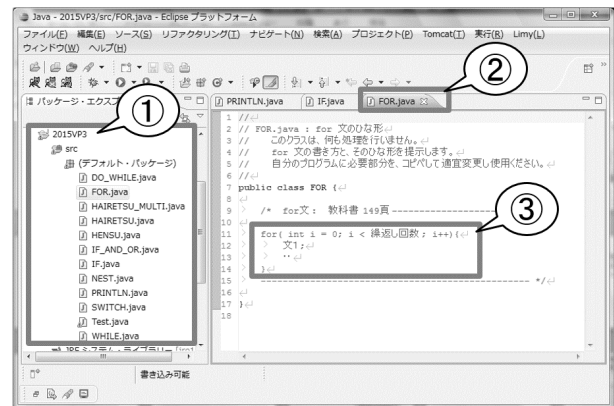


図7 Eclipse での命令毎のテンプレートクラス

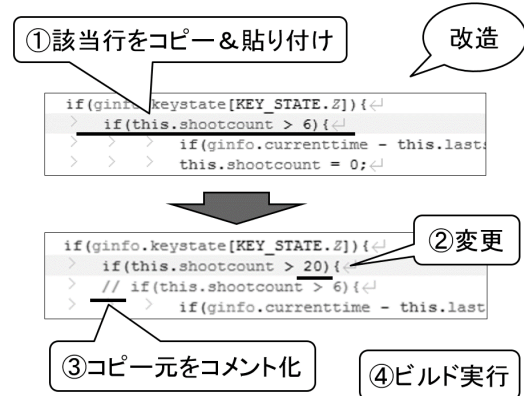


図8 アジャイル的コーディング手順

この過程における改造では、改造行の命令をコピーしてもう一行作り、それを改造するようにさせている。コピー元の命令行はコメント化して残しておくようにする。こうすることでバグが発生した際も、元の状態 (正常に動作する状態) まで容易に戻ることができる。この方法により、学生 90 人超のプログラミング演習授業を、TA 無しかつ教員 1 人でもスムーズに進めることが可能となっている。

毎回の授業では、授業中の演習結果のプログラムソースと実行結果を画面コピーしたレポート課題 (図 9) を提出させて確認し、不備は Moodle システムのコメント機能を使いアドバイスしている。これに評価点をつけて、最終的な評価に加味している。オブジェクト指向を理解しやすくするために、Java 教材の題材は、学生に馴染みのある有名なロボットアニメの設定を真似ている (図 10)。日常生活での経験を例えにしたプログラムと UML のクラス図などを作り、理解しやすくもしている (図 11, 図 12)。

【課題 1:P275Sample4.java ⇒P290Sample8.java まで反映させたもの】

```
class Car {
    public static int sum = 0;

    private int num;
    private double gas;
    // 以下にコンストラクタを定義してください。 p275
    public Car() {
        num = 0;
        gas = 0.0;
        sum++;
        System.out.println("車を作成しました。");
    }

    public void setCar(int n, double g) {
        num = n;
        gas = g;
        System.out.println("ナンバーを" + num + "にガソリンを"
            + gas + "にしました。");
    }
}
```

図 9 毎回の課題レポート例 (Word 文書)

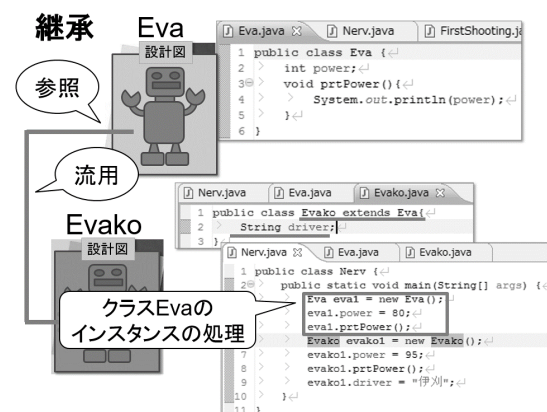


図 10 Java のオブジェクト指向プログラム教材の例

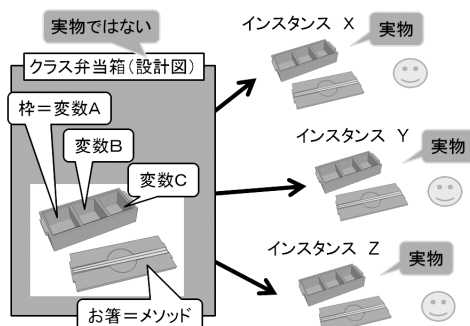


図 11 クラスとインスタンスの解説例：例え話

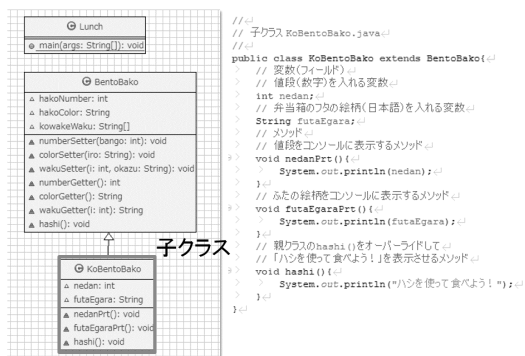


図 12 例え話の UML クラス図とプログラム例

また、動画配信システム ALF Learning⁽⁵⁾を用いて、授業で講義したスライドをそのまま使用した、授業毎、かつトピックス毎の解説ナレーション付き映像 (mp4 形式) を別途制作して多数公開しており、現在 15 個の動画ファイルを登録している。動画の

総合計時間は約 2 時間 32 分である。図 13 は登録一覧と映像例である。これらは、授業支援システムの Moodle 内でも閲覧できるようにしている。

ID	ファイル名	登録者	再生時間	状態	公開設定
133	VP3 : 0 4 : 条件 IF文とは?	小松 隆行	00:07:23	変換済	公開
132	VP3 : 2 0 : 抽象クラスとインターフェース	小松 隆行	00:21:20	変換済	公開
124	VP3 : 1 7 : コンストラクタとクラス変数・クラスメソッド	小松 隆行	00:23:05	変換済	公開
127	VP3 : 1 9 : 継承とオーバーライド	小松 隆行	00:16:29	変換済	公開
122	VP3 : 1 5 : アクセス制御とオーバーロード	小松 隆行	00:18:03	変換済	公開
123	VP3 : 1 0 : 基本事項の説明	小松 隆行	00:02:51	変換済	公開
121	VP3 : 1 4 : クラスとインスタンス：例え話	小松 隆行	00:03:32	変換済	公開



図 13 ALF Learning での動画登録一覧と内容例

3. 課題と評価について

この科目は講義と演習が中心の授業であるが、成績は提出課題 (複数回 10 点)、課題提出 (中間 15 点)、課題提出 (最終 40 点)、期末定期試験 (35 点) を総合して評価する。中間課題は、Scratch プログラムの作成と Java の基本機能を用いたプログラム作成である。最終課題は、オブジェクト指向の基本機能をすべて含んだ Java プログラムの作成である。両方の課題とも、授業での演習教材の内容の延長として機能追加や改造を行う形にしている。また定期試験期間に筆記試験を行い、専門用語や基本事項、プログラミング能力の確認と評価も行っている。

4. まとめ

本報告では、メディアデザイン学科における「ビジュアル・プログラミングⅢ」の教育内容及び方法と実践について報告した。学生の理解度の向上のため、さらなる工夫とその実践が必要と考えられる。

参考文献

- (1) Oracle : 2017 年 5 月 24 日,
<https://www.oracle.com/java/>.
- (2) Code Studio: 2017 年 5 月 24 日,
<https://studio.code.org/>.
- (3) Scratch : 2017 年 5 月 24 日,
<https://scratch.mit.edu/>.
- (4) 高橋麻奈 : やさしい Java 第 6 版,
SB クリエイティブ, 2016.
- (5) ALF Learning システム, 2017 年 5 月 24 日,
<http://alflearning-cms.hus.ac.jp/>.