

高大連携によるプログラミング学習カリキュラムの実践

坂田 圭司

東海大学情報教育センター

kgsakata@tokai-u.jp

遠藤 陵二

東海大学付属浦安高等学校

ryoji@tokai.ac.jp

ソフトウェア開発力を応用段階まで身に付けるには、知識量と実践時間の積み重ねが必要となる。大学からプログラミング学習を開始する場合には、授業のみで2年次までに基礎学習を完了させるのは困難であり、在学中に実践的なソフトウェア開発を経験する時間が少なくなる。

この状況を改善するには、初等中等教育においてプログラミング学習を行い、大学と連携して学習の継続性を持たせる必要がある。本学では、高等学校三年生における選択情報において、大学と連携したプログラミング学習を実施してその成果を2011年度から本学会で報告してきた。

本研究では、高等学校でのカリキュラム実践の結果と、大学でのプログラミング入門授業での活用について報告する。

1. はじめに

大学においてプログラミングの基礎を取り扱う授業では、プログラミング言語の文法を段階的に学ぶ内容が多い。このようなカリキュラムでは、週1コマの授業(1セメスタ15回)で、文法の基礎的な内容をカバーするのが限界である。ソフトウェア開発力を卒業研究や社会に出た後で通用する応用段階まで身に付けるには、さらに知識量と実践時間の積み重ねが必要となる。

このため、東海大学情報教育センターでは、2014年度からプログラミング系科目のカリキュラム改定を実施して、入門・基礎・応用の三段階として、さらに上位にソフトウェア開発の実践科目を用意した。特にプログラミング入門は、大学初年次を対象に、プログラミング言語に依存しないソフトウェア開発スキルを向上させることを目的として、ビジュアルプログラミング環境やアンプラグドによる学習を主とする内容にした。しかし、プログラミング入門を3セメスタ以降で受講する学生も多く、実践的な授業や自主的なソフトウェア開発を行うようになるのは、三年次後半からというパターンになる(表1)。

表1 プログラミング入門履修者数

学年	履修者数
1年次	17
2年次	26
3年次	8
4年次以上	8

(2015年度春学期・火曜4限実施)

情報分野においてソフトウェアの重要度が高まる中で、この分野への就職を希望する学生は3年次夏休みでのインターンが重要なポイントとなる。

したがって3年次春学期までに実践的なソフトウェア開発を経験しておく必要があり、学習開始時期の遅れは就職活動や研究活動に悪影響を与える。この問題の解決策として高大連携によるプログラミング学習を行い、なるべく早い時期から経験値を向上させることは必須と考える。

東海大学付属浦安高等学校では、三年生の選択情報において、東海大学と連携してプログラミング学習を中心とした授業を実施している。

この授業では、アルゴリズムの理解に注力して、2011年度からScratchをプログラム開発環境に取り入れて、学習効果を上げてきた。⁽¹⁾ 2012年度には、計測制御の理解を深めるために、グループワークによるロボットのプログラミング実習を行い、試行錯誤によって問題解決力を向上させる試みを行った。⁽²⁾ 2013年度からのカリキュラムでは、計測制御の学習内容の充実と、Processingによるプログラム記述の学習内容を追加して学習効果の改善を図った。⁽³⁾ これらの成果は、前述の「プログラミング入門」科目のカリキュラム作成の元になっている。

本研究では高大連携を深め、大学入学前および大学初年次でのプログラミング学習において、カリキュラムの改善と実践を行うことを目的とする。

2. 選択情報における授業カリキュラム

2.1 大学での学習状況

プログラミング入門開始にあたって、履修者のプログラミング学習経験を調査した。2015年春学期における調査結果を表2に示す。高校での教科情報でプログラミングを経験した者が少ない現状が分かる。また履修者には2年生以上が多いのに経験者数が少ないことから、プログラミング学習開始時期が遅いことが確認できた。

表 2 プログラミング経験者数

学年	回答数
習ったことがない	39
中学校以前から経験あり	1
高校で経験あり	4
大学で経験あり	9

(2015 年度春学期・火曜 4 限実施, 有効回答数 53)

2.2 カリキュラム内容の検討

選択情報およびプログラミング入門では, プログラムのしくみを理解し, アルゴリズムやオブジェクト指向を試行錯誤によって身に付けることを目的とした.

今回のカリキュラムの冒頭では, プログラムの「状態」と「操作」を理解して組み立てるために, VISCUIT (ビスケット) を導入した. これは手続き型プログラミングに慣れる前に, プログラム作成のアプローチにおいて柔軟な思考を養うことも目的としている.

また制御構文の理解では, Google Blockly を用いて, 条件内での目標達成を学びつつ, 順次・選択・反復処理を身に付けられる内容に, 授業後半ではプログラミング言語によるソフトウェア開発を学ぶ内容とした.

表 3 実施カリキュラム

項目	時間
VISCUIT によるプログラミング実習	2
プログラムのしくみ, Scratch の基本	2
Blockly による制御構文の理解	1
Scratch によるプログラミング基礎	3
Processing プログラミング	4
計測制御グループ学習	8
C#プログラミング	10
GUI アプリケーション作成実習	6

3. 実施内容

VISCUIT, Scratch, 計測制御プログラマ, ビューポートビルダーP を用いたプログラミング環境を活用することで, 文法にとらわれない目標となる学習内容に注力することが可能になる.

実際のソフトウェア開発では, プログラミング言語を用いるのが一般的であるため, その理解と実践を学習する必要がある. Processing を用いることで入門段階から, 本格的な言語による学習の間に繋ぐ有効な手段となる. また応用的なプログラミング学習でも活用可能であり, 特にグラフィックスとイベント処理の連携は, シンプルかつ高

度な機能を持ち, データの可視化をプログラミングで行う実習に向いている. 加えて, 組込みプログラミングで広く使われている Arduino との連携, 開発環境の互換性によって, Processing 自体を実践的な学習で使い続けることができる.

今回の授業の対象は, 東海大学付属浦安高等学校の 3 年生選択情報科目 (週 1 回 2 時間連続開講). 授業は東海大学の教員 1 名と付属高等学校の教員 2 名で行った.

前述の内容追加に加え, グループ学習開始時にはマシュマロチャレンジを取り入れて, 効果的なグループワークを可能にしている.

4. 期待される効果

プログラミング学習において, 高大連携を進めることは, 社会に通用するプログラミングスキルを実質的に身に付けるためには重要である. プログラミング学習は, 他の内容以上に体系的かつ連続性のあるカリキュラムが必要となる. 本研究では, 「情報の科学」からの継続性を保ちつつ, 本格的な言語によるプログラミングに繋げ, 自分の手でソフトウェア開発できる基礎力の充実を期待できる.

参考文献

- (1) 坂田圭司, 遠藤陵二: ソフトウェア開発における基礎学習の実践と評価, 第 5 回日本情報科教育学会全国大会 (2012).
- (2) 坂田圭司, 遠藤陵二: 高大連携によるプログラミング学習の実践, 第 4 回日本情報科教育学会全国大会 (2011).
- (3) 坂田圭司, 遠藤陵二: プログラミング学習における高大連携, 第 6 回日本情報科教育学会全国大会 (2013).
- (4) Jerry Lee, Jr. Ford: Scratch Programming for Teens, Course Technology Ptr(2008)