

初等中等教育段階におけるプログラミング学習の必要性

久野 靖*

2014.6.1

1 はじめに

□ 情報処理学会と（高校までの）情報教育

- 情報科の最初の指導要領制定時→試作教科書による内容提案「プログラミングもこれくらい入れたらどうですか」
- 情報科がはじまった結果を見て→提言 2005「手順的な自動処理が大切。これをぜひもっとやってほしい」
- 指導要領改訂時→「もっと高度な内容と積み上げ型に」
- 今回の（来る）改訂→試作教科書 2012「選択必修修をやめて統一を」
- 疑問： こんなおとなしい提案をされていていいの？

□ 世の中の趨勢の変化

- 2005 年ころ→筧先生が「プログラミング」とか言ったら記者の方たちがよってきて詰問
- 現在→「小学生にプログラミング」ブーム（多くの活動、IT 戦略会議）
- プログラミング言語・環境も大きく変化（Viscuit、Scratch、Dolittle、アルゴリズムック）
- 今こそ「プログラミングをちゃんと入れるべきだ」と言った方がいい？

□ 本日の内容

- 「情報学」の定義 ← 情報学の参照基準
- 「コンピューティング」が重要だという話
- 小学校から高校まで系統的に ← 個人的提案
- 小学校 1 年からできるという実例

2 情報学の参照基準

□ 参照基準： 大学の学士課程において学生が身につける専門分野の内容を定める

- 文科省の依頼により学術会議で各分野ごとに策定中
- 学生が「職業人」「市民」として生きて行く基礎・基本

- 学問分野固有の特性への本質的理解+それに根ざした教育
- 「学士課程」「専門教育」だが、初等中等と関連（情報教育の親学問）
- 分野固有の能力+汎用的能力

□ なぜ情報学は必要か

- わが国のソフトウェアやシステムは遅れている
- ユーザ企業には情報システムが分かる人がいない
- 素人からの聞き取り調査に基づいて専門家が作るという体制はだめ
- 「プログラマを育成しなくては」は間違い：「プログラマをちゃんと評価でき、プログラマと話ができる人を育成しなくては」
- あるべき姿： 初等中等で全員が情報学を学び→その一部が専門家（システム開発側、ユーザ側）に

□ 情報学の定義： 「情報学は、情報によって世界に意味・価値を与え秩序をもたらすことを目的に、情報の創造・生成・収集・表現・記録・認識・分析・変換・伝達にかかわる原理と技術を探究する学問である」

- コンピュータなどの情報技術に限定されない包括的な定義
- これまで「情報教育の親学問は?」「情報科学?」「それでは狭い」のような議論が多数繰り返されて来たが→ついに決着したかも

□ 情報学に固有の（他分野と明確に区別される）知識の体系

- ア． 情報一般の原理 --- 抽象的な内容（大学レベル）
- イ． コンピュータで処理される情報の原理 --- デジタル情報+コンピューティング（アルゴリズムを用いて情報を扱うこと）
- ウ． 情報を扱う機械および機械を設計し実装するための技術 --- ハード・ソフト・ネットワーク
- エ． 情報を扱う人間と社会に関する理解 --- 「情報社会」

*筑波大学ビジネスサイエンス系

- オ． 社会において情報を扱うシステムを構築し運用するための技術・制度・組織 ---- ソフトウェア工学、HCI など

□ これを現在の情報教育（高校の情報科）と比較すると…

- アは扱われていない（もともと抽象度の高い大学レベルの内容）
- イ～オはそれぞれの入門部分がひとつおり情報科に含まれている→すばらしい！
- しかしよく見ると！ → イ～オの各所でコンピューティングの深い理解が前提となっている
- 情報科ではコンピューティングは「ごく一部」

□ まとめると： 「情報学の本質部分であり、情報教育にはほとんど含まれていない内容」＝「コンピューティング（アルゴリズムを用いて情報を扱うこと）」

- 克服すべき問題： 情報技術の専門家が時間を掛けて学んで来ていること（コンピューティング）について、ほとんど触れたことがなく、何であるかも分からない利用者によって引き起こされるさまざまな問題

3 コンピューティングの重要性

□ 世界的に「小学校からのコンピューティング学習」ブーム

- 米国： オバマ大統領やビル・ゲイツ氏などの呼びかけ、多数の活動
- 英国： 2014年からの新たなカリキュラムで「コンピューティング」→全児童・生徒のプログラミング学習を義務づけ→単にプログラミングできる、ではなく抽象化、モデル化、計算的思考などきちんと考えることを重視

□ なぜコンピューティングが重要なのか？

- 情報技術を担う将来の人材を今から育成していかなければ（職業ニーズ）
- 情報技術に無知なまま使うのは危険である（技術教育的視点）
- 情報技術に無知な人たちにはまともな情報システムの発注ができない
- 情報技術に無知な人たちには革新的な情報システムを構想できない
- 新しいタイプの考え方（計算的思考）を身につけさせたい

□ なぜコンピューティングが重要なのか？（つづき）

- 「仕事にこれが必要だと思ったらさっさと作る人」vs「手作業を続けたり、どうしてもなくなったら誰かに発注してすったもんだする人」では絶対に勝負にならない

- アイデアを思い付いたらその人が作るのが一番よい（他人に説明していると本質が損なわれたり時間が掛かりすぎたりする）

- 現に企業の情報システムについても「内製化」を進めようとする企業が出て来ている

- 多重下請けによる低生産性という日本の悪癖はいつまでもは続けられない（そんなお金はなくなるから）

□ 久野の独断でまとめると、めざすことは、誰もが高校までで…

- 「コンピュータは結局こういうものだ」と納得すること（今は個別のアプリが出るたびにすごいと思ってしまう）

- 「必要なツールは自分で作れるものだ」と納得すること（実際に作れる人は半分くらいでもいいから）

- 「こういう作業はこうすればコンピュータにやらせられる」という感覚を持つこと（そうすれば説明でき、発注者になれる）

- 「実際にソフトウェアを作るということはこういうプロセス」という感覚を持つこと（発注者になるためにもこれは必要）

4 小学校から高校までの系統的教育

□ 現在の「情報活用能力」ではコンピューティングが出てこない

- 「情報の科学的理解」を「コンピューティングの理解と活用能力」のようなものに変更すべきでは

- 「科学的理解」の中にある「手段の選択・評価」「問題解決」はこれからは「実践力」の一部として扱うのがいいのでは

- 以下、そのような考えで久野の独断で各学校段階の「案」を示す

□ 全体的な方針として…

- 小学校： 発達段階として可能な範囲で十分な経験と準備をおこなう

- 中学校： 現在の高校の「情報の科学」+ α 程度

- 高校： 「実際に作れる」「理論面の理解」「開発作業の理解」を目標とする

□ 小学校

- 低学年：適切なプログラミング環境を使用して、簡単な自動処理の手順を記述させ、そのふるまいを観察させるとともに、手順を修正すると動作も変化することを理解させ、計画した動作に対してこれを実現する記述を計画し実現させる。
- 中学年：適切なプログラミング環境を使用して、自動処理の手順記述中に条件判断が含まれるようなものを記述され、条件判断の成否により動作が変化することを観察するとともに、場合分けを持つ動作を計画し、それを実現する手順を実現させる。
- 高学年：適切なプログラミング環境の使用を前提として、ある程度込み入った動作をおこなう作品を計画し、繰り返しや条件判断などを組み合わせて、計画した動作を実現させる。

□ 中学校

- コンピュータの本質的な仕組みや、その万能性、性能向上の意義について理解し、コンピューティングとデジタル情報やコンピュータの関わりについて説明できる。
- コンピュータを利用した計測・制御の基本的な仕組みを知り、さまざまな組み込みシステムの利用場面や使用されている理由を理解する。
- アルゴリズムやデータ構造の概念を理解した上で、プログラミング言語を用いて、自分が構想した処理を適切なアルゴリズムやデータ構造を用いて実現できる。

□ 高校

- プログラミング言語を用いて、(自分個人用、集団内などレベルの差はあっても)実際に役に立つソフトウェアを記述できる。
- 問題解決に必要なアルゴリズムを考えられるとともに、計算量や計算可能性などの概念を理解し、アルゴリズムの実用性を判断できる。
- 問題を解決するためのソフトウェアが持つべき要件を整理でき、そのようなソフトウェアを設計・構築するプロセスが理解できる。

□ まとめ： 今後は「小学校から高校まで、系統的かつ段階的に」コンピューティングを学ぶようなカリキュラム構成が理想では？

- 実際に理想が実現するかはまた別の問題として…

5 小学生対象のプログラミング教育

□ 「小学校1年からコンピューティングなんてできるの？」への解答

- 久野が前から知っていて、論文化のためにお手伝いした事例
 - 「みどりっ子クラブ」： 墨田区立緑小学校の課外活動
 - 勝沼さん： コンピュータの専門家ではない普通の人
 - 原田さん（ビスケット作者）の助けを借りて活動を開始
 - 原田さんは通常はいない。子どもたちの教え合いでうまく行っている
- ビスケット --- 画面上に配置した絵の書き換え規則によるプログラミングシステム

- キーボード入力が不要、制御構造が不要 → 発達段階的に小さい子でも OK
- 実際「ビスケット塾」では未就学児童も対象に開催している
- 規則は単純だがさまざまな「技」を組み合わせて高度な動きが可能
- 「ゲーム」も作れる→子どものモチベーション的にも有効

□ みどりっ子クラブの運用上の特徴

- ビスケットシステムのさまざまな工夫
- 「ノートに設計を書く」→「それに従って作る」
- 「ビスケット検定」で「技」を意識→その「技」を知っている子は知らない子に教えられる
- カリキュラム的に言えば「思ったように動かす」「交互」「ぶつかる等の条件による分岐」などが対応
- 運営者はコーディネーター+大人としての見守り(モラル)
- アクティブラーニング、教え合い→楽しければ学ぶ

□ 少人数に対するインタビュー評価

- 「絵を描いて動かすことは楽しい」
- 「できることはコンピュータとスマホで変わらない」
- 「ゲーム機は中身を作れないけれどコンピュータでは作れる」
- 「教えることは楽しい」 ←教え合いが重要

□ まとめ： 小学校対象に目標を設けてプログラミング体験は可能

- 最初は「計画して、その計画のように動く」
- 次は「枝分かれのような、条件に従う変化」
- ビスケットでかなり高度なものまで作れるが、どこから先はテキスト言語に移行するのがよいと考える

□ おまけ： 英国のカリキュラム「Computing」

- 小学校低学年： アルゴリズムとは何か、簡単なプログラムの構築+デバッグ
 - 小学校高学年： 物理システムの制御やシミュレーション、接続/選択/反復、変数、複数アルゴリズムの比較
 - 中学校： 状態やふるまいのモデル化、アルゴリズムの有用性、データ構造、手続きによる抽象化
 - 高校： 「より高度」
- これはかなり欲張りなんでは？ 本当にできるんだろうか？

6 まとめ

- 情報教育の親学問としての「情報学」→コンピューティングが中核
- 今後の日本では社会の構成員全員にコンピューティングの素養を求めたい
- 小学校から高校まで段階的に学んで行くという提案
- 小学校で実践している例 ← 英国のほど過激ではない

参考文献

- [1] England Department of Education, The national curriculum in England --- Framework document, september 2013.
<https://www.gov.uk/government/collections/national-curriculum>
- [2] 萩谷, 情報学を定義する --- 情報学分野の参照基準, 情報処理学会誌, vol. 55, no. 7, 2014 掲載予定.
- [3] 原田, 勝沼, 久野, 公立小学校の課外活動における非専門家によるプログラミング教育, 情報処理学会論文誌, 掲載予定.