

# プログラミング人材育成の在り方に関する調査研究 報告書

平成 27 年 6 月

 総務省

## 目次

1. 背景及び目的	1
2. 調査方法	2
3. アンケート・ヒアリング調査の実施概要	3
3.1. プログラミングに関する教育の現状・実態の調査	3
3.1.1. アンケート調査	3
3.1.2. ヒアリング調査	4
3.2. プログラミングに関する教育がもたらす効果等の調査	4
3.3. プログラミング人材に対する企業のニーズ	5
4. プログラミングに関する教育で一般的に使われるツール	6
4.1. プログラミングに関する教育に用いられる主な言語	6
4.2. プログラミングに関する教育に用いられる主なハードウェア	13
5. プログラミングに関する教育事例の取りまとめ	17
5.1. 教育関連団体	17
5.2. 受講者	22
5.3. 教育の目標・対象	24
5.4. 教育形式・教育コンテンツ	27
6. プログラミングに関する教育がもたらす効果	32
6.1. プログラミングに関する教育がもたらす効果に関する研究の状況	32
6.1.1. 子供向けのプログラミング言語の開発経緯	32
6.1.2. 文献調査結果	32
6.1.3. 有識者ヒアリング調査結果	33
6.1.4. プログラミングに関する教育に関する評価指標	34
6.2. プログラミングに関する教育がもたらす効果に関する教育事業者への調査結果	34
6.3. プログラミングに関する教育がもたらす効果	38
7. プログラミング人材に関する段階的な育成手法	41
7.1. 段階的な育成方法	41
7.1.1. 基礎的能力の発達に基づく段階的育成方法	41
7.1.2. プログラミングスキルに応じた段階的育成方法	45
7.2. 高度 ICT 人材育成につながる取組み	46
7.3. 学習ツールの特徴	48
7.3.1. プログラミング言語の特徴	48
7.3.2. ハードウェアの特徴	50
8. 民間企業が求めるプログラミング人材像	51

8.1. 調査対象の概要 .....	51
8.2. 期待する能力 .....	52
8.3. 理想とする人材像 .....	54
8.4. 人材の不足感 .....	54
<b>9. プログラミングに関する教育の整理 .....</b>	<b>56</b>
9.1. プログラミングに関する教育の目指す方向性 .....	56
9.2. プログラミングに関する教育の場 .....	57
9.3. プログラミングに関する教育の導入段階 .....	58
9.3.1. 動機付け .....	59
9.3.2. 継続的な学習 .....	60
9.3.3. 深化 .....	62
<b>10. プログラミングに関する教育の普及展開に有効な取組手法 .....</b>	<b>63</b>
10.1. 普及啓発に関する課題とその解決策 .....	63
10.1.1. 現状と課題 .....	64
10.1.2. 普及啓発における課題のまとめ .....	69
10.2. 運営方法に関する課題とその解決策 .....	70
10.2.1. 現状と課題 .....	70
10.2.2. 運営方法における課題のまとめ .....	73
10.3. 普及展開のための取組み案 .....	74
10.3.1. モチベーションの向上および効果に関する認知度の向上 .....	74
10.3.2. 地域コミュニティとの連携による地域格差の解消 .....	75
10.3.3. 共通して利用できるプラットフォームの活用 .....	76
10.3.4. その他の有効な取組み案 .....	76
<b>A. 付録 .....</b>	<b>78</b>
1. プログラミングに関する教育の現状・実態アンケート .....	78
2. 企業ニーズアンケート .....	94

## 目次

図 2-1 本調査の実施手順.....	2
図 4-1 ロゴ坊による二進木の描画プログラムとその実行結果イメージ.....	8
図 4-2 「ドリトル」による緑色の三角形を描くプログラムと実行結果イメージ.....	8
図 4-3 Viscuit の操作方法.....	9
図 4-4 Scratch の画面イメージ.....	10
図 4-5 Blockly の迷路ゲームの画面イメージ.....	10
図 4-6 スモウルビーの画面説明.....	11
図 4-7 スモウルビーの Ruby への翻訳イメージ.....	11
図 4-8 プログラミン画面イメージ.....	12
図 4-9 RaspberryPi 第二世代、Raspberry Pi 2 Model B.....	14
図 4-10 IchigoJam.....	14
図 4-11 LEGO® MINDSTORMS® Education EV3 Core Set.....	15
図 4-12 EV3 ソフトウェアによるプログラミングイメージ.....	15
図 4-13 Pepper.....	16
図 5-1 教育関係団体の経営形態.....	17
図 5-2 プログラミング教室・講座の地域別教室数.....	18
図 5-3 プログラミング教室・講座の開始時期.....	19
図 5-4 講師のプログラミングスキルと人数について.....	20
図 5-5 講師の募集方法（複数回答可）.....	21
図 5-6 プログラミング教室・講座の受講者数.....	22
図 5-7 生徒の募集方法（複数回答可）.....	23
図 5-8 教室・講座の対象年齢（複数回答可）.....	26
図 5-9 受講者の男女比.....	26
図 5-10 教室・講座の受講形式.....	27
図 5-11 教室・講座で利用されているプログラミング言語（複数回答可）.....	29
図 5-12 中心となるテキスト教材.....	29
図 5-13 オリジナルのテキスト教材のイメージ.....	30
図 5-14 使用している機器（複数回答可）.....	31
図 6-1 プログラミングに関する教育がもたらす効果.....	35
図 6-2 21 世紀型能力（出典：国立教育政策研究所試案をもとに弊社加工）.....	39
図 7-1 段階的教育および高度 ICT 人材育成につながる取組み.....	47
図 8-1 調査対象企業の事業区分（複数回答）.....	51
図 8-2 調査対象企業の従業員数.....	51
図 8-3 企業が期待する能力(N-23).....	52
図 9-1 プログラミングに関する教育の方向性.....	57
図 9-2 プログラミングに関する教育の導入段階.....	58
図 10-1 プログラミング教室・講座に対する想定される保護者の思考・行動.....	63
図 10-2 各フェーズにおける離脱率のイメージ.....	69

## 表目次

表 4-1 プログラミング教室等で利用されている言語 .....	6
表 4-2 プログラミング教室等で利用されるハードウェア .....	13
表 5-1 プログラミングに関する教育を提供する教育関連団体の分類.....	17
表 6-1 プログラミングの特性とプログラミングに関する教育の効果.....	38
表 6-2 プログラミングに関する教育がもたらす効果 .....	40
表 7-1 段階的指導の区分と典型的な指導方法 .....	43
表 7-2 「段階 2」の時期における段階的指導方法の典型.....	46
表 7-3 ヒアリングで得られた各言語への意見 .....	49
表 7-4 ヒアリングで得られた各ハードウェアへの意見.....	50
表 8-1 期待する能力に関する選択率(N=23).....	52
表 8-2 企業ニーズと教育事業者のアンケートにおける選択肢の対応関係.....	53
表 8-3 人材不足があると回答した企業の比率(N=13).....	55
表 9-1 学校への出張授業の概要 .....	59
表 9-2 ワークショップの概要 .....	59
表 9-3 スクールの概要 .....	60
表 9-4 アフタースクール、児童館でのクラブ活動の概要.....	61
表 10-1 各フェーズの課題と課題解決に参考となる取組み.....	70
表 10-2 各カテゴリにおける課題及び参考となる取組み.....	73
表 10-3 普及啓発及び教室運営における課題とその解決策.....	74

## 1. 背景及び目的

昨今、プログラミング人材育成の重要性に関して国際的な認識が高まるなか、イギリス、ギリシャ、エストニア等、初等教育の段階からプログラミングを正式教科として導入する国<sup>1</sup>が増えている。

我が国は世界最先端の ICT 技術を保有しているが、これを活かして世界をリードしていくためには、より多くの子供たちが早い段階から ICT 利活用の素地を磨くことが不可欠である。

これについては、本年6月に改訂された「世界最先端 IT 国家創造宣言」（平成26年6月24日閣議決定）において、「初等・中等教育段階におけるプログラミングに関する教育の充実に努め、IT に対する興味を育むとともに、IT を活用して多様化する課題に創造的に取り組む力を育成することが重要」とされており、また、総務省の各種会議においても、プログラミング人材育成の重要性やプログラミングに関する教育を通じた論理的思考力の向上の可能性について言及されており、総務省としてプログラミング人材の育成に取り組むこととしている。

総務省におけるプログラミング人材育成の取組とは、プログラミングに関する高度な技術者を直接的に育成しようとするものではなく、青少年（18歳以下程度を指す。以下同じ。）の発達段階に応じたプログラミングに関する教育を通じて、将来の高度 ICT 人材としての素地の構築・資質の発掘を図ろうとするものである。

このように、本件は、「世界最先端 IT 国家創造宣言」等の ICT に関する政府方針や高度 ICT 人材を育成するために実施してきた実践的 ICT 人材育成事業につながる事業と位置付けられるものである。

本事業は、教育関係団体（NPO 法人・民間教育事業者・教育関係機関（原則として学校を除く。）、学識経験者及び民間企業等への調査及び文献調査を通じ、プログラミング人材に必要な育成手法等を明らかにすることで、将来の高度 ICT 人材へとつながる青少年のプログラミング人材の育成に関する総務省としての政策展開の基礎資料とするものである。

なお、本調査研究は、株式会社三菱総合研究所に調査実施を依頼し、NPO 法人 CANVAS 等の協力を得ながら進めた。

---

<sup>1</sup> Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe, European Schoolnet, 2014

## 2. 調査方法

本調査では、下図に示す手順により、青少年のプログラミング人材育成に関する実態、効果、企業ニーズを把握し、課題や解決策および普及展開の取組手法についてまとめた。

調査は、大きく分けて（１）アンケート・ヒアリング調査および（２）分析・取りまとめ等から構成され、（１）において、教育事業者、学識経験者、ICT ベンダ・利活用企業へのアンケート・ヒアリングにより実態を把握し、（２）において、（１）の結果に基づき分析・整理を行った。

本調査の実施に関しては、株式会社三菱総合研究所が調査の遂行を行い、アンケート調査・ヒアリング調査の回答者への依頼においては、NPO 法人 CANVAS の協力を得て進めた。

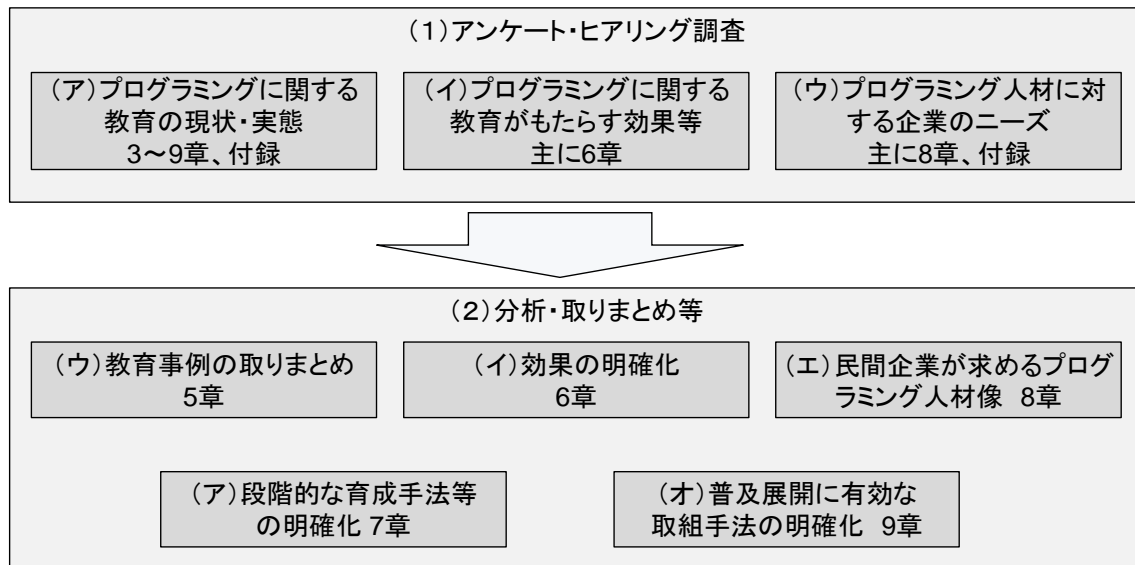


図 2-1 本調査の実施手順

「（２）（ウ）教育事例の取りまとめ」においては、主に、「（１）（ア）プログラミングに関する教育の現状・実態」に基づき、プログラミングに関する教育事例について、その現状・実態を取りまとめた。「（２）（イ）効果の明確化」においては、主に「（１）（イ）プログラミングに関する教育がもたらす効果等」をもとに、プログラミングに関する教育がもたらす効果についてまとめた。

「（２）（エ）民間企業が求めるプログラミング人材像」については、主に「（１）（ウ）プログラミング人材に対する企業のニーズ」の結果をもとに、民間企業が求めるプログラミング人材像をまとめた。

「（２）（ア）段階的な育成手法等の明確化」については、「（１）（ア）プログラミングに関する教育の現状・実態」、「（１）（イ）プログラミングに関する教育がもたらす効果等」をもとに、青少年の発達段階に応じてどのように教育を行うことが有効かについてまとめた。

以上の結果をもとに、「（２）（オ）普及展開に有効な取組手法の明確化」において、プログラミングに関する教育の認知度の向上、子供たちのモチベーションの向上等の方法論を取りまとめた。

### 3. アンケート・ヒアリング調査の実施概要

以下の3つの項目についてアンケート及びヒアリング調査を実施した。

- プログラミングに関する教育の現状・実態
- プログラミングに関する教育がもたらす効果等
- プログラミング人材に対する企業のニーズ

各項目の実施概要について、以下に記載する。

#### 3.1. プログラミングに関する教育の現状・実態の調査

プログラミングに関する教育の現状・実態については、アンケート調査、ヒアリング調査の2段階で実施した。全国の教育関係団体（NPO法人・民間教育事業者・教育関係機関）に第一次調査としてアンケートを実施し、さらに深掘り調査としてヒアリングを実施した。

##### 3.1.1. アンケート調査

アンケートでは、プログラミングに関わる教育を実施している教育関係団体計43団体を対象とし、25の有効回答を得た。

アンケート対象については、プログラミングに関わるワークショップなどのイベントの開催やプログラミングに関わる講座・スクールの提供を行っている教育関係団体を対象とし、Web上の関連文献及びホームページをキーワード検索により抽出し、対象団体を選定した。

選定にあたっては、地域の偏りがないように全国の教育関係団体を抽出するとともに、大都市だけでなく、中都市や小都市の教育関係団体も抽出した。対象43団体の属性は以下の通り。

- |                            |         |
|----------------------------|---------|
| ● 地域                       |         |
| 東日本/西日本で事業を展開              | ： 4 団体  |
| 東日本で事業を展開                  | ： 25 団体 |
| 西日本で事業を展開                  | ： 14 団体 |
| ● 都市規模                     |         |
| 大都市（人口 50 万人以上の都市）         | ： 28 団体 |
| 中都市（人口 10 万人以上 50 万人未満の都市） | ： 11 団体 |
| 小都市（人口 10 万人未満の都市）         | ： 4 団体  |



アンケート調査は、電子ファイル形式の調査票の送付により上記 43 団体に依頼を実施し、計 25 の有効回答を得た。

主な調査項目は以下の通り。

- 育成の対象となっている年齢・属性等
- 教育関係団体等における育成の目的・目標
- 教育関係団体等の指導計画・教材等
- 指導方法等のポイント
- 講師の確保・育成に関する課題・解決方法
- プログラミング講座の開講に当たっての課題・解決方法及びPRポイント
- 受講生確保の課題・解決方法
- プログラミング講座の実施によって想定される又は得られる効果
- プログラミング人材育成の普及展開に関する取組事例

### 3.1.2. ヒアリング調査

第二次調査として、アンケート対象の事業者のうち計 16 団体にヒアリング調査を実施した。ヒアリング対象の選定にあたっては、以下の観点で満遍なく選定を実施した。

- 地域  
東日本/西日本 4 団体、東日本 6 団体、西日本 6 団体
- 都市規模  
大都市 10 団体、中都市 4 団体、小都市 2 団体
- 教育対象（重複あり）  
幼児・小学生 15 団体、中学生 10 団体、高校生 3 団体
- 利用しているプログラミング言語（重複あり）  
Scratch 9 団体、Viscuit3 団体、レゴ®マインドストーム®EV3 3 団体  
Objective-C 2 団体、他 Ruby、スクイーク Etoys、Visual Basic、Java 等

### 3.2. プログラミングに関する教育がもたらす効果等の調査

プログラミングに関する教育がもたらす効果等については、教育関係団体及び学識経験者へのヒアリング調査を実施した。

プログラミングに関する教育の現状・実態の調査で実施した教育関係団体 16 社に加え、有識者 3 名に、プログラミングに関する教育がもたらす効果や、青少年の発達段階に応じたプログラミングに関する教育にふさわしいスキルとそのレベルについて、ヒアリング調査を実施した。

ヒアリング対象は以下の通り。

- 学識経験者 3 名  
NTT コミュニケーション科学基礎研究所 原田 康德 氏  
(専門分野：計算機科学 (プログラミング言語開発))  
青山学院大学・津田塾大学 非常勤講師 阿部 和広 氏  
(専門分野：計算機科学 (子供向けプログラミング))  
京都大学大学院 教育学研究科・教育学部長 子安増生 教授  
(専門分野：教育認知心理学)
  
- 教育関係団体 16 団体  
東日本/西日本 4 団体、東日本 6 団体、西日本 6 団体  
(うち、大都市 10 団体、中都市 4 団体、小都市 2 団体)

### 3.3. プログラミング人材に対する企業のニーズ

プログラミング人材に対する民間企業のニーズについて、ICT ベンダ及び ICT 利活用企業計 116 社を対象にアンケート調査を実施し、23 の有効回答を得た。

アンケート対象については、ICT ベンダ (ソフトウェアを開発する企業)、ICT ベンダ (機器システムを開発する企業)、ウェブビジネス企業 (インターネット上のウェブサービスを提供する ICT 利活用企業)、ICT 利活用企業 (ウェブビジネス企業以外の ICT 製品を提供せず、ICT を利活用する企業) の 4 者に対して実施した。アンケート対象の分類は以下の通り。

- ICT ベンダ (ソフトウェアを開発する企業) : 33 社
- ICT ベンダ (機器システムを開発する企業) : 21 社
- ウェブビジネス企業 : 30 社
- ICT 利活用企業 : 32 社

アンケート調査では、プログラミングに関する教育により育成が期待される人材のニーズや現状の不足感について調査を実施した。

## 4. プログラミングに関する教育で一般的に使われるツール

プログラミングに関する教育で利用されているプログラミング言語やハードウェアについて文献調査を行った。

### 4.1. プログラミングに関する教育に用いられる主な言語

プログラミングに関する教育に用いられる主な言語は表 4-1 の通り。

表 4-1 プログラミング教室等で利用されている言語

言語	言語	概要
LOGO (ロゴ)	教育用テキスト言語	<ul style="list-style-type: none"> <li>・児童の思考能力向上を目的として開発された言語</li> <li>・命令文によって画面上の「タートル」を動かし、タートルの軌跡で線画を描く</li> </ul>
ドリトル	教育用テキスト型言語	<ul style="list-style-type: none"> <li>・オブジェクト指向言語</li> <li>・LOGO 同様にタートルを動かし、図形などを描く機能がある</li> <li>・日本語に対応</li> </ul>
Viscuit (ビスケット)	ビジュアル言語	<ul style="list-style-type: none"> <li>・手書きイラストを用いたアニメーション作成機能に特化したプログラミング言語</li> <li>・タブレットで利用可能</li> </ul>
Scratch (スクラッチ)	ビジュアル言語 (ブロック型言語)	<ul style="list-style-type: none"> <li>・ブロックの組み合わせによってプログラミングするオブジェクト指向言語</li> <li>・8才以上が主な対象</li> <li>・外部コントローラとの接続によりロボット操作等も可能</li> <li>・日本語に対応</li> </ul>
Pyonkee (ピョンキー)	ビジュアル言語 (ブロック型言語)	<ul style="list-style-type: none"> <li>・タブレット端末 (iPad) で動作する、Scratch を基にした言語</li> <li>・2014 年に合同会社ソフトウメヤにて開発</li> </ul>
Scratch Jr.	ビジュアル言語 (ブロック型言語)	<ul style="list-style-type: none"> <li>・Scratch よりもさらに低年齢 (5～7才) 向けに開発された言語</li> <li>・タブレットで利用する</li> </ul>
Blockly (ブロックリー)	ビジュアル言語 (ブロック型言語)	<ul style="list-style-type: none"> <li>・Google が提供している教育用プログラミングライブラリ。</li> <li>・導入段階用に迷路ゲーム等が用意されている</li> <li>・タブレットで利用可能</li> </ul>
Smalruby (スモウルビー)	ビジュアル言語 (ブロック型言語)	<ul style="list-style-type: none"> <li>・Ruby を基としたビジュアル言語</li> <li>・Ruby との互換性あり</li> <li>・島根県松江市の中学校の授業でとりいれられている。</li> </ul>
プログラミン	教育用ビジュアル言語 (ブロック型言語)	<ul style="list-style-type: none"> <li>・文部科学省が開発したブラウザで動作するプログラミング学習用サービス</li> <li>・タブレットで利用可能</li> </ul>
JavaScript	テキスト型言語	<ul style="list-style-type: none"> <li>・プログラミング教室・講座では WEB アプリケーション</li> </ul>

		の作成に用いられることが多い
Java	テキスト型言語	<ul style="list-style-type: none"> <li>・プログラミング教室・講座では Android アプリ作成に用いられるケースが多い</li> </ul>
Python	テキスト型言語	<ul style="list-style-type: none"> <li>・全米の大学では初心者プログラミングを教育する教材として最もカリキュラムに取り入れられている</li> <li>・オブジェクト指向型スクリプト言語</li> </ul>
C	テキスト型言語	<ul style="list-style-type: none"> <li>・現在もっとも普及しているプログラミング言語とされる</li> <li>・国際標準化機構(ISO)や日本工業規格(JIS)にも標準として採用されている</li> </ul>
Objective-C	テキスト型言語	<ul style="list-style-type: none"> <li>・OS X や iOS 向けのアプリケーションの開発に利用できる言語（従来は Objective-C に代わる言語として 2014 年 6 月、Swift が発表されている）</li> </ul>
HTML	テキスト型言語	<ul style="list-style-type: none"> <li>・ウェブ上の文書を記述するための言語</li> <li>・現在は改定版である HTML5 が多く利用されており、HTML5 では高度な Web アプリも作成可能</li> </ul>
Basic	テキスト型言語	<ul style="list-style-type: none"> <li>・50 年の歴史をもつ</li> <li>・中高の教科書にも掲載されている</li> </ul>
Arduino	テキスト型言語	<ul style="list-style-type: none"> <li>・マイコンボード Arduino の開発用言語</li> <li>・C 言語に近い</li> </ul>

## (1) LOGO (ロゴ)

LOGO は MIT 教授 Seymour A. Papert が児童の思考能力向上を目的として 1960 年代に開発された。LOGO は命令文によって画面上の「タートル」を動かし、タートルの軌跡で線画を描く。タートルグラフィックと呼ばれ、現在は LOGO 以外の言語でも使われている。

日本でも 1980 年度に学校授業で活用され、日本語版 LOGO である「ロゴ坊<sup>2</sup>」が大阪電気通信大学兼宗教授によって開発された。

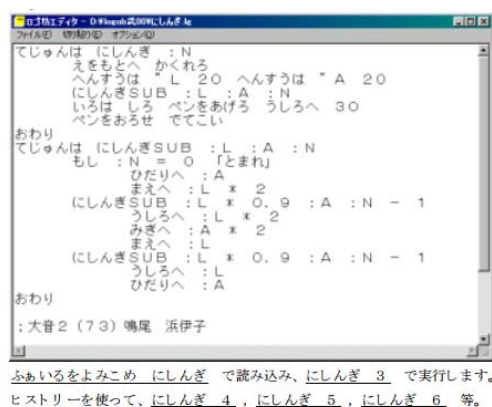


図 4-1 ログ坊による二進木の描画プログラムとその実行結果イメージ<sup>3</sup>

## (2) ドリトル

日本語でプログラミングできる、初学者向けプログラミング教育言語 LOGO 坊の後継言語として、2000 年に筑波大学久野教授、大阪電気通信大学兼宗教授が開発した。LOGO と同様に、画面上の亀を動かし、亀の軌跡で絵を描く。描画機能が強化されており、線画だけでなく、色づけや塗りつぶし等も可能である。また、描画だけでなく、ゲームを作成することもできる。

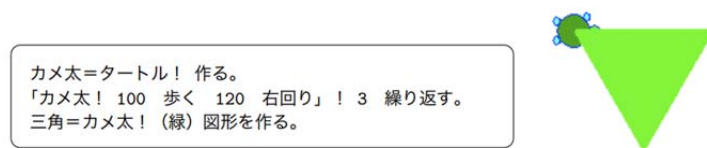


図 4-2 「ドリトル」による緑色の三角形を描くプログラムと実行結果イメージ<sup>4</sup>

<sup>2</sup> プログラミング言語「ドリトル」 > 「LOGO 情報室」 > 「ロゴ坊」(2015 年 3 月アクセス)

<http://dolittle.eplang.jp/index.php?Logo%E6%83%85%E5%A0%B1%E5%AE%A4>

<sup>3</sup> 「ロゴ坊 (LOGOB) 要項」(広瀬達雄) より (2015 年 3 月アクセス)

<http://dolittle.eplang.jp/index.php?plugin=attach&refer=Logo%E6%83%85%E5%A0%B1%E5%AE%A4&openfile=hirose01.pdf>

<sup>4</sup> 「ドリトルで学ぶ」(兼宗進) P16

<http://dolittle.eplang.jp/index.php?plugin=attach&refer=%E3%83%80%E3%82%A6%E3%83%B3%E3%83%AD%E3%83%BC%E3%83%89&openfile=book.pdf>

### (3) Viscuit (ビスケット) <sup>5</sup>

2003年2月に原田康徳氏によって教育用プログラミング言語として開発された。ブロック型ビジュアル言語とは異なり、変数や配列等の概念を持たず、手書きイラストを用いたアニメーション作成機能に特化し、絵だけでプログラムを作る。そのため、幼い子どもでもアニメーションやゲームを作成することができる。



図 4-3 Viscuit の操作方法<sup>6</sup>

### (4) Scratch (スクラッチ)

MIT メディアラボ<sup>7</sup>で Squeak Etoys (スクイーク)<sup>8</sup>をベースに開発された教育用ビジュアルプログラミング開発環境。ブロック型言語とも呼ばれ、ソースコードをテキスト入力するのではなく、命令文にあたるブロックをくみ上げることで、プログラミングを体験することができる。Scratch で作成されたゲームは、オフィシャルサイト上で世界中と共有することができる。

<sup>5</sup> ヴィジュアルプログラミング言語 Viscuit ビスケット <http://www.viscuit.com/>

<sup>6</sup> ヴィジュアルプログラミング言語 Viscuit ビスケット ビスケットの使い方 (2015年3月アクセス) <http://www.viscuit.com/howto/>

<sup>7</sup> アメリカのマサチューセッツ工科大学に設置された研究所。 <http://www.media.mit.edu/>

<sup>8</sup> Squeak Etoys とは、アラン・ケイ博士によって作られた教育用オブジェクト指向プログラミング言語

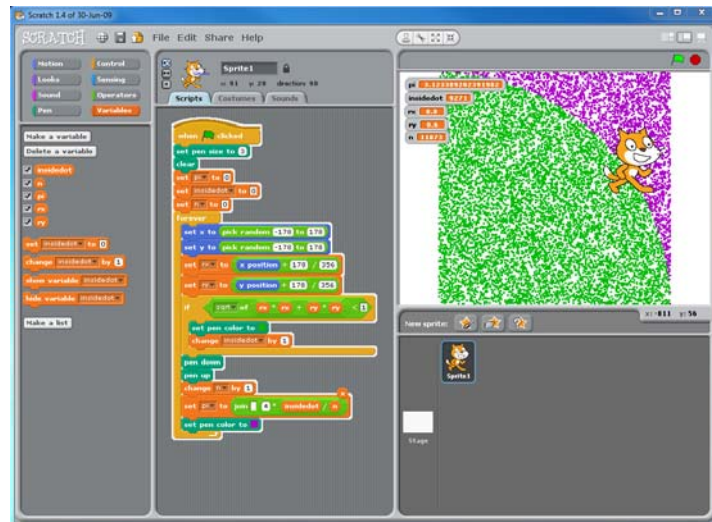


図 4-4 Scratch の画面イメージ<sup>9</sup>

(5) Blockly (ブロックリー) <sup>10</sup>

Google が作成したビジュアルプログラミング開発ライブラリ。ブラウザ上で動作するオープンソースのブロック型言語である。オフィシャルサイトには、Blockly をつけた子供向けゲームが準備されている。

ブロックリー: 迷路 1 2 3 4 5 6 7 8 9 10



図 4-5 Blockly の迷路ゲームの画面イメージ<sup>11</sup>

<sup>9</sup> <http://www.stemulate.org/2013/03/27/learn-scratch-raspberry-pi/>

<sup>10</sup> Blockly (Google Developers) <https://developers.google.com/blockly/>

<sup>11</sup> 「Google Blockly」をやってみた！グーグルが提供するビジュアルプログラミング言語 (Tech ACADEMY magazine) (2015 年 3 月アクセス) <http://techacademy.jp/magazine/1812>

(6) Smalruby (スモウルビー) <sup>12</sup>

Ruby の教育用プログラミング言語として Scratch を参考に、高尾宏治氏（ネットワーク応用通信研究所）によって開発された。島根県松江市の市立中学校では Smalruby を用いた授業が行われている。

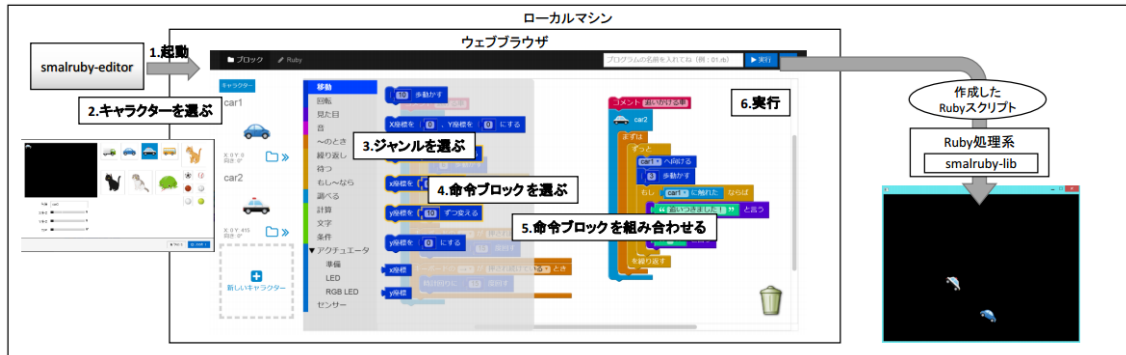


図 4-6 スモウルビーの画面説明<sup>13</sup>

Smalruby は Ruby へ変換可能である。



図 4-7 スモウルビーの Ruby への翻訳イメージ<sup>13</sup>

<sup>12</sup> Ruby プログラミング少年団 <http://smalruby.jp/>

<sup>13</sup> [http://www.ruby.or.jp/ja/news/20140404.data/2013\\_grant\\_Takao.pdf](http://www.ruby.or.jp/ja/news/20140404.data/2013_grant_Takao.pdf)



## (7) プログラミン<sup>14</sup>

文部科学省が Scratch を参考に開発した教育用プログラミング言語。ブラウザ上で動作し、作品を共有することもできる。



図 4-8 プログラミン画面イメージ<sup>15</sup>

<sup>14</sup> きみの絵を動かそう！プログラミン（文部科学省）（2015年3月アクセス）

<http://www.mext.go.jp/programin/>

<sup>15</sup> 「プログラミン」をやってみた！（TECH ACADEMY magazine）（2015年3月アクセス）

<http://techacademy.jp/magazine/801>

## 4.2. プログラミングに関する教育に用いられる主なハードウェア

プログラミングに関する教育ではPCやタブレット端末以外のハードウェアが用いられることも多い。PC、タブレット端末以外のプログラミングに関する教育に用いられる主なハードウェアは表 4-2 の通り。

表 4-2 プログラミング教室等で利用されるハードウェア

ハードウェア		概要
シングル ボードコ ンピュー タ	RaspberryPi (ラズベリー パイ)	<ul style="list-style-type: none"> <li>・小型のボードコンピュータ</li> <li>・若年層向け教育用コンピュータとして英国ラズベリーパイ財団が開発</li> <li>・USB ポート、HDMI ポート、LAN ポート、GPIO (汎用入出力) ポートといった入出力を備える</li> </ul>
	IchigoJam (イチゴジャ ム)	<ul style="list-style-type: none"> <li>・Basic のみを搭載したボードコンピュータ</li> <li>・子どもの教育用パソコンとして開発されており、LAN ポートがないためインターネットには接続できない</li> </ul>
	Arduino	<ul style="list-style-type: none"> <li>・入出力ピンを備えた小型のボードコンピュータ</li> <li>・Arduino 言語による統合開発環境 (IDE) がセットで提供されている</li> <li>・オープンソースハードウェアであり、ハードウェア設計情報が公開されている</li> </ul>
レゴマインドストーム EV3		<ul style="list-style-type: none"> <li>・教育用組み立て式ロボット</li> <li>・レゴ社と MIT が共同開発</li> <li>・工具不要でレゴブロックのように組み立てが可能</li> </ul>
Pepper (ペッパー)		<ul style="list-style-type: none"> <li>・コミュニケーション特化型ロボット</li> <li>・Softbank 社により 2014 年に発表された</li> <li>・API の公開と SDK の提供により個人のアプリ開発が可能</li> <li>・開発言語は Python、C++ 等であり、ビジュアル開発環境「Choregraphe」も提供されている</li> <li>・Pepper App Challenge を Softbank 社が開催</li> </ul>
ロボット、センサーカー等		<ul style="list-style-type: none"> <li>・マイコンをコントローラとし、教育用プログラミング言語等で制御可能なロボットやセンサーカー</li> <li>・中学校の技術の教科書等でも紹介されているものもある</li> </ul>

## (1) Raspberry Pi (ラズベリーパイ)<sup>16</sup>

教育用の小型 PC として開発され、2012 年 2 月に発売されたボードコンピュータ。安価(数千円程度)で小型だが Linux 等の OS を搭載すれば、インターネットアクセスも可能である。2015 年 2 月に第二世代が発売され、処理速度が高速化した。

日本では Google と NPO 法人 CANVAS<sup>17</sup>による取組み (PEG<sup>18</sup> : Programming Education Gathering) によって 2014 年の 1 年間、5000 個の RaspberryPi が教育現場へ配布され、RaspberryPi で Scratch を学ぶプログラミング教室・講座が開催された。また、むき出しの基盤の PC だが小型で比較的安いため、教育現場では外部コントローラを接続し、動かすといったワークショップも行われている。



図 4-9 RaspberryPi 第二世代、Raspberry Pi 2 Model B<sup>16</sup>

## (2) IchigoJam (イチゴジャム)<sup>19</sup>

教育用小型コンピュータとして 2014 年に福野泰介氏が開発<sup>20</sup>。子ども部屋に安心して置けることをコンセプトとし、インターネット接続等はできず、電源を入れると BASIC の画面がすぐに立ち上がる。RaspberryPi よりも小型で安価 (2000 円未満) となっている。



図 4-10 IchigoJam<sup>19</sup>

<sup>16</sup> Raspberry Pi (RS) (2015 年 3 月アクセス) <http://jp.rs-online.com/web/generalDisplay.html?id=raspberrypi>

<sup>17</sup> NPO 法人 CANVAS <http://www.canvas.ws/>

<sup>18</sup> Programming Education Gathering <http://pegpeg.jp/>

<sup>19</sup> IchigoJam <http://ichigojam.net/>

<sup>20</sup> <http://fukuno.jig.jp/615>

### (3) レゴマインドストーム EV3<sup>21</sup>

プログラミングとロボットテクノロジーを体験・学習するためのツールとして開発されたプログラミングロボット教材。モーターや各種センサー（超音波センサー、カラーセンサー、ジャイロセンサー等）、CPU 内蔵のインテリジェントブロック等を組み立ててロボットを作成する。



図 4-11 LEGO® MINDSTORMS® Education EV3 Core Set<sup>22,23</sup>

作成したロボットは EV3 ソフトウェアでビジュアルにプログラミングし、制御する。



図 4-12 EV3 ソフトウェアによるプログラミングイメージ<sup>24</sup>

また、レゴマインドストームを使った世界大会（WRO : World Robot Olympiad）<sup>25</sup>があり、世界の 40 か国、5 万人の青少年が競い合っている。

<sup>21</sup> 教育版レゴ®マインドストーム EV3

<https://education.lego.com/ja-jp/preschool-and-school/secondary/mindstorms-education-ev3>

<sup>22</sup> <http://shop.legoeducation.com/gb/product/lego-mindstorms-education-ev3-core-set-45544-198/>

<sup>23</sup> <http://www.lego.com/ja-jp/mindstorms>

<sup>24</sup> プログラミング「直観的なビジュアルプログラミング」（LEGO education）（2015 年 3 月アクセス）

<https://education.lego.com/ja-jp/preschool-and-school/secondary/mindstorms-education-ev3/teaching-resources/software/programming>

<sup>25</sup> NPO 法人 WRO Japan <http://www.wroj.org/2015/>

#### (4) Pepper (ペッパー)

2014年にSoftBank社より発表された、感情認識ロボット。ロボットの開発環境が公開されているため、Pepperを制御するアプリを開発できる。開発はPython、C++等の言語で可能であり、グラフィカルにPepperの動きや音声をプログラムできる「Choregraphe (コレグラフ)」も提供されている。

また、Pepperを用いたプログラミング教室やアプリコンテスト（青少年に限らず対象は全年齢）<sup>26</sup>が開催されている。



図 4-13 Pepper<sup>27</sup>

---

<sup>26</sup> Pepper app challenge 2015（2015年3月アクセス）<http://www.softbank.jp/robot/special/app-challenge/>

<sup>27</sup> ホーム>製品情報>ロボット（SoftBank）（2015年3月アクセス）<http://www.softbank.jp/robot/products/>

## 5. プログラミングに関する教育事例の取りまとめ

アンケート及びヒアリング結果に基づき、プログラミングに関する教育事例について整理した。

### 5.1. 教育関連団体

#### (1) 経営形態・主たる事業

組織の経営形態としては、「その他」が最も多く、続いて株式会社、NPO 法人であった。教育関係団体の経営形態についてアンケート結果を図 5-1 に示す。

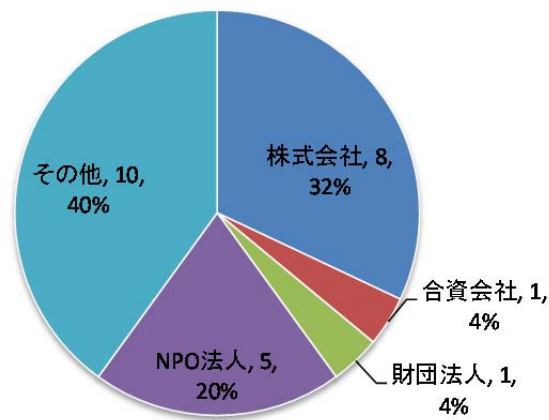


図 5-1 教育関係団体の経営形態

「その他」の多くは法人化していない組織であり、プログラミングに関する教育の多くが法人化されていない組織により非営利目的で提供されていることが読み取れる。NPO もあわせると、非営利目的の組織は半数を超え、現状では非営利の活動が中心となっていることがわかる。非営利の NPO や組織では、プログラミングに関する教育の提供を主目的とする NPO・組織に加えて、こどもの教育や子育てを主目的とした NPO・組織と、モノづくりを主目的とした NPO がプログラミングに関する教育活動を実施している。他に、企業の CSR 活動や大学の公開講座等の地域・社会貢献の一環としての活動もみられる。

営利目的でプログラミングに関する教育を提供している事業者は、アンケートでは全体の 3 分の 1 程度であった。プログラミング教室を主な事業とする事業者のほか、パソコンスクール、学習塾、教育 ICT 事業者などが子ども向けプログラミング教室を展開するケースがあり、多様な事業者がプログラミングに関する教育を展開している。

教育関連団体の組織の分類と主な教育関連団体の例を表 5-1 に示す。

表 5-1 プログラミングに関する教育を提供する教育関連団体の分類

教育関連団体の分類		主な教育関係団体
営利	プログラミング教育専業事業者	CA Tech Kids、Life is Tech!、ロボット科学教育 Crefus、TENTO 等

	ICT 企業・教育 ICT 企業	IDEA Makers、NEL&M 等	
	幼児教室・学習塾	Qremo、にいがたプログラミング塾、ロボ 団、ラーニングセンター新浦安等	
	パソコンスクール	札幌プログラミングスクール、ThinkDo、 パソコン教室 PLADY、科学技術館パソ コン教室等	
非営利	NPO、法人化し ていない組織・ 個人	プログラミング教育 の提供を主目的と した NPO・組織	CoderDojo、Ruby プログラミング少年団、 Little Coder Mie、プログラミングクラブネ ットワーク (PCN) 等
		こどもの教育、子育て を主目的とした NPO・組織	CANVAS、子ども文化コミュニティ、新 座子育てネットワーク、みどりっ子クラブ 等
		モノづくりを主目的 とした NPO・組織	Maker Lab Nagoya、新発田科学技術教育ネ ットワーク等
	ICT 企業	CSR 活動	SCSK、日本 HP、NTT データ等
	大学・高専	公開講座等、地域・社 会貢献の一環	東京大学、東京工科大学、埼玉大学、久留 米工業大学等

## (2) 地域

プログラミング教室・講座の地域別教室数のアンケート結果を図 5-2 に示す。

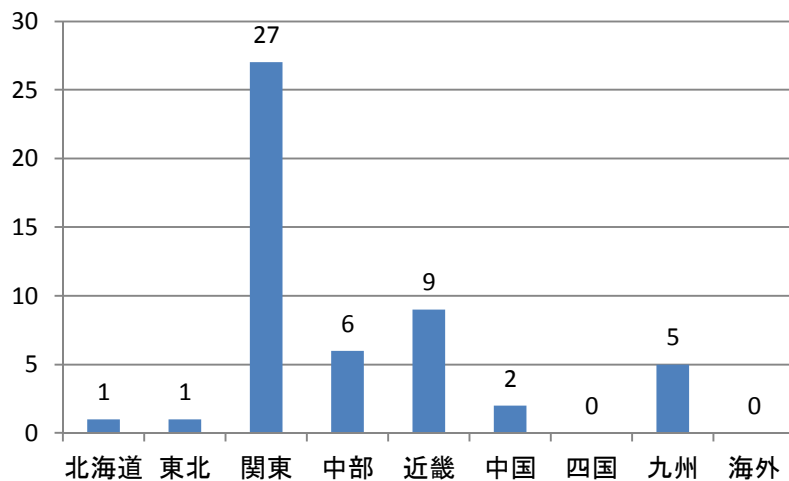


図 5-2 プログラミング教室・講座の地域別教室数

プログラミング教室・講座の開催地の多くは関東に集中している。関東の多くは都内であり、近畿地方に含まれる大阪と比べても東京とは大きな差が出ている。また、都市の規模でみた場合、多くの教室・講座が大都市での開催であり、中都市・小都市は事例も少なく、プログラミングに関する教育の認知度が現状では低いと考えられる。

### (3) 開始時期

図 5-3 に示すように、約半数の団体が 2014 年以降にプログラミング教室・講座を開始しており、2013 年開始も含めると約 6 割の団体が該当する。1999 年から 2012 年はほぼ 1 年に 1 団体の増加ペースであるが、2013 年以降にプログラミング教室・講座を開催する団体が急増している。

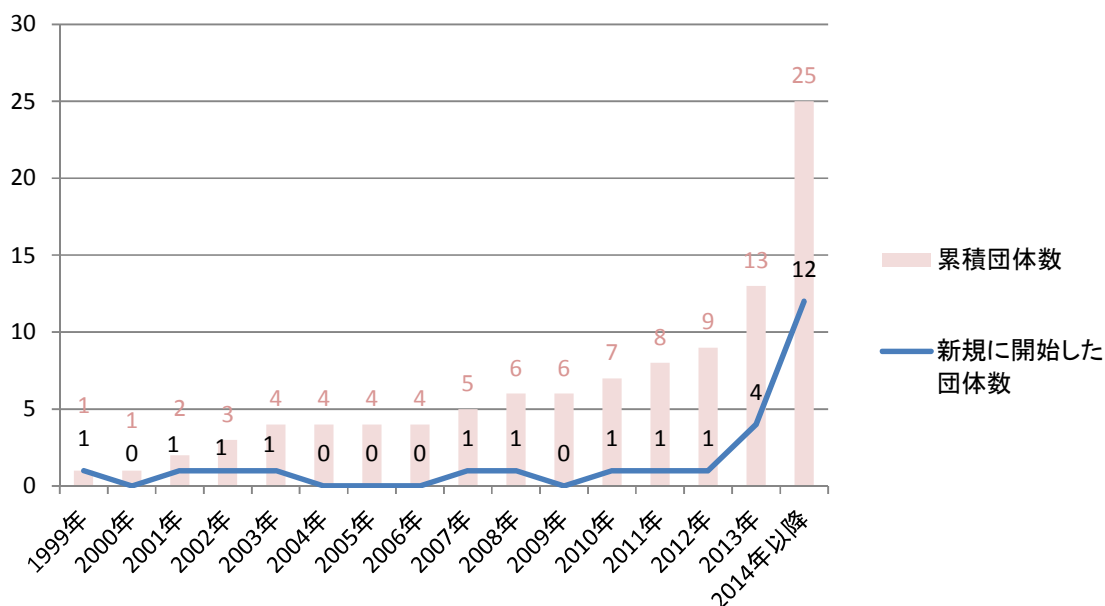


図 5-3 プログラミング教室・講座の開始時期

ヒアリングにおいても、この 2 年間ほどで参入事業者が急増しているとの指摘が多く、現状は多くの事業者が始めたばかりで手探りで進めている状態であるともいえる。

### (4) 講師属性・講師数

教育関係団体の講師のスキルについては、高度なプログラミングスキルを持つ講師が多く、講師数については、半数以上の団体が 2 人以下としている。また、学生アルバイトを雇用している団体は約 2 割 (5/23) であった。講師のプログラミングスキル及び講師の人数についてのアンケート結果を図 5-4 に示す。



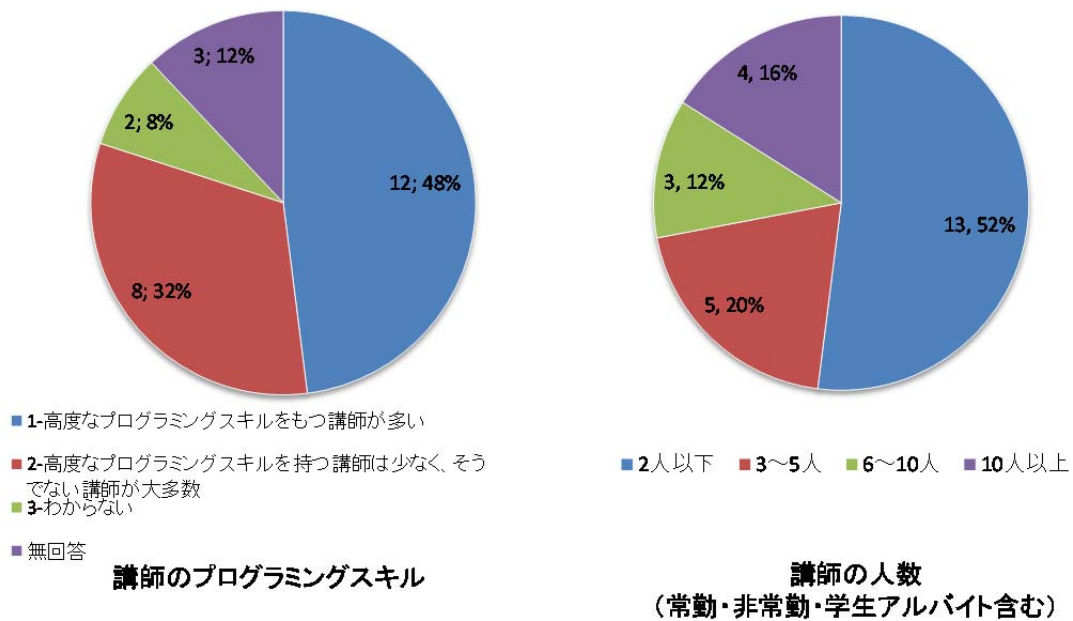


図 5-4 講師のプログラミングスキルと人数について

講師の属性や講師数について、ヒアリングで以下のような意見が挙げられた。

- プログラミング教育にはスキルが必要であり優秀な人材が必要。コンテンツもトレンドの移り変わりに沿って刷新する必要がある。アルバイトもプログラミングができて、コミュニケーション力もある学生を雇用している。
- エラーがでたときの対応などはスキルが必要。アルバイトも理系の大学学部生・院生が中心でほとんどがプログラミング経験者。
- 基本的な指導はプログラミング経験がほとんどない講師が行っているが、スキルの高い指導者の継続的な活用が課題。現在は、不定期に大学の先生などと呼んでいる。
- 受講者のプログラミングの技術レベルが上がると、専門家がいないと対応が難しくなる。

ヒアリングにおいても、講師にはプログラミングを教えるうえで一定以上のプログラミングスキルが必要であるとの意見が多く挙げられ、現状では、プログラミングスキルを有する講師が少人数を対象に実施している事例が多いといえる。また、プログラミング能力以外にも、教えるためのコミュニケーション能力が必要とされており、アルバイトを雇用する場合にも、そういった高い能力を持つ学生を雇用している事例が多い。

一方、プログラミングに関する教育の提供を主目的としていないNPOや放課後教室での実施事例もみられ、そういった事例ではプログラミングが非専門の講師・ボランティアにより実施されている。こういった団体では、開始前の準備や、継続的に行い受講者のスキルが上がってきたときの講師のスキル不足が課題として挙げられており、何れかの段階で専門家の支援を受けているケースも多い。

## (5) 講師の育成・募集

前述したように多くの事業者が高度なプログラミングスキルをもった講師 1、2 名という体制で実施しており、育成・募集ともに行っていないというケースや友人・知人の紹介に頼っているケースが多い。講師の育成・募集についてのアンケート結果を図 5-5 に示す。

募集している場合の募集手段としては、友人・知人の紹介、ウェブサイトが多く、積極的に募集していないのが現状であるといえる。

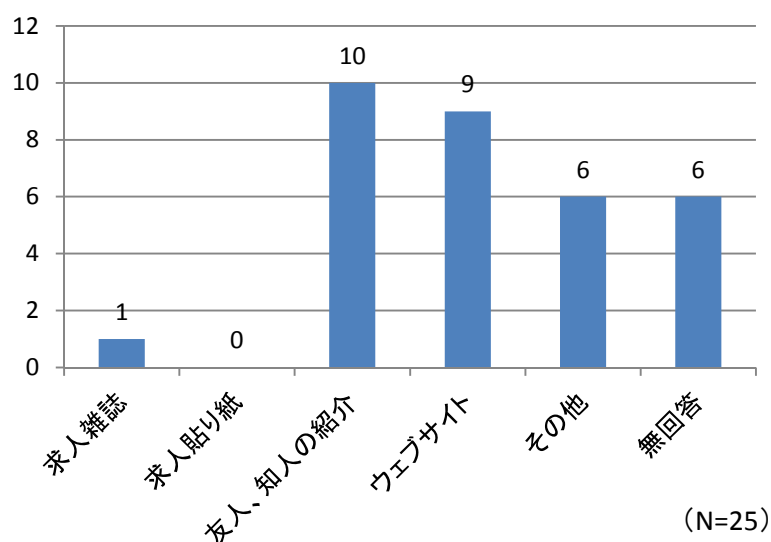


図 5-5 講師の募集方法（複数回答可）

ヒアリングでは、育成・募集するほどニーズが顕在化しておらず、その予算もないとの意見もある一方で、特に地方においては講師不足が課題となっているとの意見もあった。講師不足の場合の取り組みとしては、地域のプログラマーの発掘や、シニア・主婦などの活用とそのような講師を支援する取り組みが挙げられた。

ヒアリングで得られた意見を以下に示す。

- 講師については民間エンジニアや大学と連携するネットワークをもっているため、間に合っている。
- 地方では講師の確保が課題。プログラミング教育に関心を持つ高校専門学校や大学の学生をバイトとして雇いたいと考えている。
- ボランティアをつのるようにしている。ワークショップに参加した子供の父親が手伝ってくれているケースもある。
- 地域のプログラマーを発掘できるようなものがあればよい。やろうとなったときに地域に見当たらないため、やることができない。
- シニアにもプログラミング経験者がいる。今後はいわゆる現役世代以外が講師役をしてもよいと考えている。
- シニアが講師役となるプログラミング教室の開催は医療費削減、子育て支援、ICT 人材育成にもなってよいのではないかと。シニアの非 ICT 技術者でも教えられるような教材を作っている。

- プログラミング教室にスケールメリットが働かないのであれば、家庭の主婦が講師になってちょっと開けるような広がり方があればよいのではないか。

## 5.2. 受講者

### (1) 受講者数

受講者数についてのアンケート結果を図 5-6 に示す。

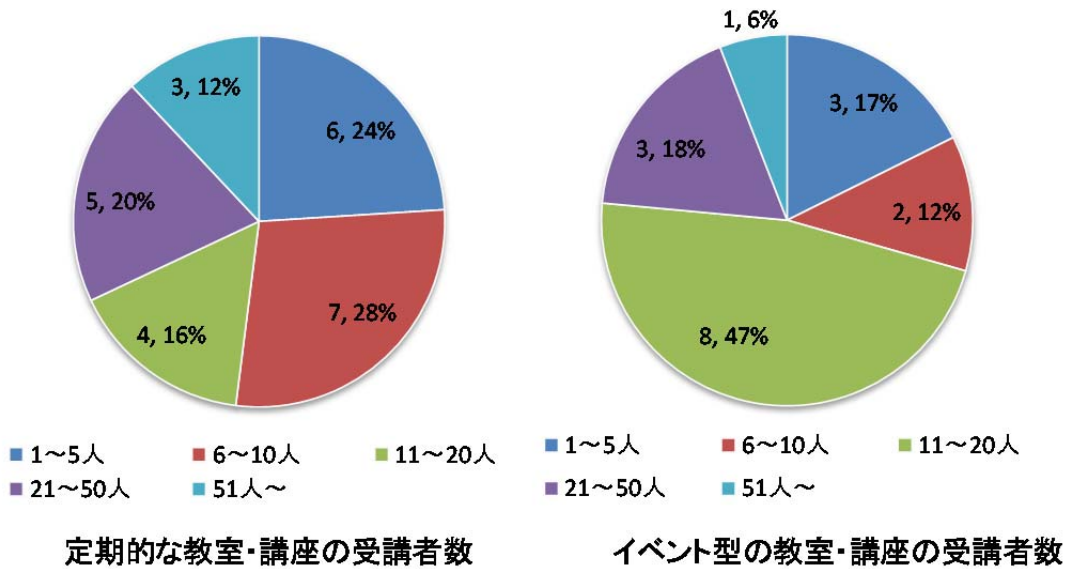


図 5-6 プログラミング教室・講座の受講者数

受講者数については、定期的に受講する教室・講座については過半数が 1~10 人であり、50 人以上の受講者を有する教室・講座は 3 教室・講座（3 団体）のみであった。50 人以上の受講者を有する団体では、ビジネスとしてプログラミングに関する教育が成立しているものの、多くの団体では継続的な受講者数は比較的少数に留まっている。

一方、イベント型の教室・講座については、11~20 人という受講者数が最も多く、約半数を占めるといった結果であった。

地域という観点では、関東以外の中都市や小都市では 10 人以下で開催しているケースが多く、20 人以上の受講者を有する教室・講座の半数以上が都内での開催であった。

### (2) 受講者の募集

生徒の募集方法についてのアンケート結果を図 5-7 に示す。

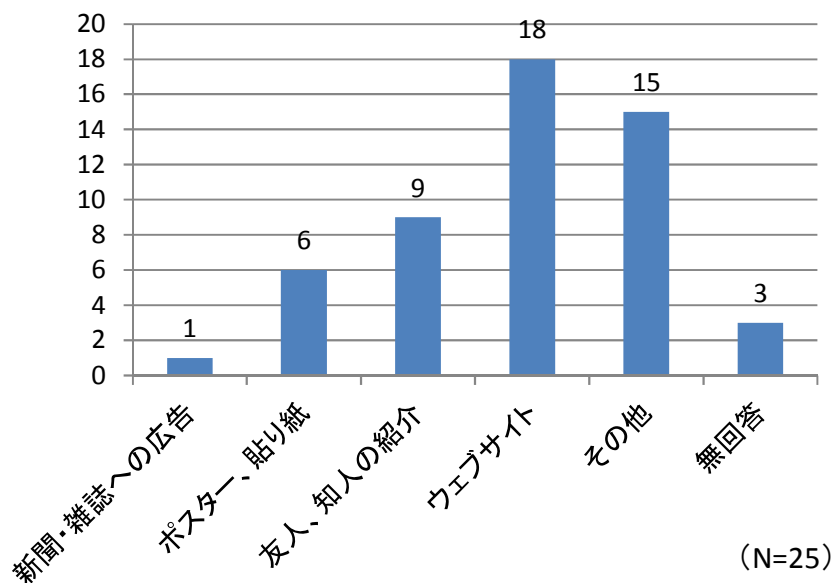


図 5-7 生徒の募集方法（複数回答可）

多くの団体がウェブサイトを利用して募集を実施している。その他では、SNS やポスティング、チラシの配布、市報などが募集手段として挙げられた。新聞・雑誌への広告などは現状ではほとんどなく、大きな宣伝は実施していないのが現状であるといえる。

アンケート及びヒアリングでは、ウェブサイトのみの募集でも一定の人数が集まっているという意見もある一方で、子供や親への訴求力や認知度の不足から、受講者の確保を課題とする団体も多かった。特に、関東以外の中都市や小都市の事例では受講者の募集に苦労しているケースが多く、地方でのプログラミングに関する教育の認知度不足が課題として多く挙げられた。また、全国で教室・講座を開講している団体からも、東京とそれ以外の地域での開催では集客に大きな差が出ているとの声もあり、東京とそれ以外の地域では顕在化しているニーズに大きな差があると考えられる。

他には、講師や会場のキャパシティの面から受講者の募集を敢えて積極的に実施していないケースもみられた。

また、受講者数の維持という観点では、継続して通う教室・講座の運営者から、プログラミング教室で長期に継続して通ってもらうことの難しさも課題として挙げられた。例えば、Scratch では半年程度で一通り基本的な事項を教えている場合が多く、そのコースが終了した後、さらに継続してもらうための工夫が必要とされている。

ヒアリング及びアンケートで挙げられた意見を以下に示す。

受講者の募集に課題を感じていないケース

- 世間がプログラミングの重要性に気づき始めており、「小学校 プログラミングスクール」等の検索でウェブサイトへ到達する方が増えている。
- 募集は Web のみで毎回 15 名程度。それでもそれなりに集まっている。
- 定員に対して希望者が多すぎ対応できない状況は課題である。

受講者の募集に課題を感じているケース

- 募集は市政だよりに掲載すると参加者が多い。各学校へチラシを置くこともあるがなか

なか集まらない。子供や親への訴求力が地方では低い。地域特性によると思うが、東京のようにプログラミング講座がすぐに埋まる状況ではない。

- 受講生の確保は課題。NHK のニュース等で取り上げられたあとは満員近くに申込みがあるが、チラシ配布だけではそこまで大きくない。
- 現状は費用の面から経済的に余裕のある家庭の子供の受講が多数であり、イベントに参加して興味があっても場所の面から遠くて通えない子もいる。
- プログラミング教育の市場は拡大しているように見えるが、実態は良く分からない。生徒の募集は、九州で反応が良く、大阪ではそれほどでもない。
- イベント 2 回の受講生はそれぞれ 2 人で、あとはその保護者。今後も開催していく予定だが、募集方法が課題。
- 九州エリアではプログラミング教育に対する認知度が極めて低く、ニーズが顕在化しない。むしろ子供を ICT 機器から遠ざけようとする動きのほうが強い。
- 地方では保護者のプログラミング教育に対する意識が低く、小学生にプログラムを体験させたいと思っている保護者はきわめて少ない。
- 活動資金も限られており宣伝できないので、受講生がなかなか集まらない。
- 有料のワークショップはかなり認知度もあがり、参加者が増えてきたが、教室の受講生の確保は非常に難しい状況。
- 多くの人に開催情報を知っていただきたいが、小学校・中学校での告知は難しいところがある。Web で告知しているが、情報拡散するために広告費用がかかるため、多くの方に開催情報を展開できない。
- 教育委員会等も通じて広報しているが、地方ではまだまだプログラミングに興味のある子どもたちに開催の情報を届けるのが難しい。募集期間を長めにとれば、定員には達する。
- まだ地域において認知度が高くない（情報発信が不十分）
- 地域においてプログラミングで学ぶことそのものへの認知が広がっていないことが課題。

#### 積極的に募集を実施していないケース

- 募集は Web サイトのみ。現状ではキャパシティの問題もあり、プログラミング教育をやっているということをあまり宣伝していない。
- キャパシティ（講師数）の問題で、教室への参加者募集はしていない。

### 5.3. 教育の目標・対象

#### (1) 人材育成の目標像

人材育成の目標については、課題解決力や創造力の育成、自己表現力・個性の育成などが掲げられているケースと、プログラミングを通じたコンピュータ・ICT の理解が掲げられているケースが多い。前者では特に課題解決、創造性といったキーワードが多く使われている。

プログラミング教室・講座で育成する人材の目標像について、アンケート及びヒアリン

グでの回答を以下に示す。

#### 課題解決力、創造力

- Plan→Do→Check&Action を意識して論理的に課題 (Mission) の解決を進めることのできる児童生徒、チームメイトと学びあい協力することで課題解決を粘りつよく実践できる児童生徒、国際大会で自分たちの技 (工夫したところ) を英語で発表したり、外国の友達と交流できる児童生徒の育成。
- 設立の目的を国際性、創造性の育成としており、そういった人材の育成を目指している。
- コミュニケーション能力や創造性豊かに育つために、広い意味でのメディアリテラシーを身に付けてほしい。
- 目指す人材像は、創造する人、オリジナリティの高い人である。
- 子供たちがプログラミングの楽しさを体感し、楽しみながらプログラミングすることで、論理的思考力、問題解決力、そして創造力を育み、これからの時代を“生き抜く”力を身につけて欲しい。
- 流れ図やシステム図でも物事を整理し俯瞰的に見て問題解決や創造物を作ることが出来る戦略性のあるクリエイティブな人間。
- さまざまなワークショップを通して子どもたちの創造性や表現力、コミュニケーション力をはぐくむことを目指しています。
- 主体性や感性、創造力、コミュニケーション力、メディアリテラシー能力の向上と育成をめざしている。
- 課題を解決できる人間になってほしい。課題解決ができるようになるには、特に目の前にある課題を解決するというのが大きい。

#### 自己表現力

- プログラミングを小・中学生の自己表現手段と考えている。企業、社会が求める問題解決能力、コミュニケーション能力を身に着けてゆく基盤となる要素に内面の成長が必要と考える。
- 外でよく遊び、自分を取り巻く世界をよく観察し、そこで見たこと・感じたことをプログラミングで表現をしてほしい。そうすることで、自分の考えやアイデアを表現できる力、論理的にものを考え構築する力を養うことができます。
- 将来、プログラマーになるということではなく、最新の ICT 技術を活用したものづくりによって個性を伸ばす教育を提供したいということを目指しています。

#### コンピュータ・ICT の理解

- プログラミングでゲームを作るだけでなく、ゲーム機の仕組みまで理解することを目的としている。
- コンピュータやアプリなどを必要に応じて創造することができる、コンピュータやインターネットなどのメディアに対する正しい知識を持ち、バランス良く活用できる、プログラミングを楽しむことができる、人材の育成が目標。
- ICT に関する総合的な理解とリテラシーを最重要にしている。
- 遊びとしてプログラミングを楽しめる人材育成。
- プログラムに興味を持って貰い、自己学習をしていけるような人材。

#### 地域に貢献できる人材

- 地域で子供が育つことを目標としている。
- 「やってみる」「触れてみる」という新しい体験・機会を提供する場を地域につくることが目標です。そしてその場から得たものを地域への還元していけるような人になってもらえることを期待しています。

## (2) 育成の対象となっている年齢・属性等

教室・講座の対象年齢についてのアンケート結果を図 5-8、受講者の男女比を図 5-9 に示す。

小学校 4 年生～6 年生から中学生を対象とした事例が最も多く、小学校 1 年生～3 年生、中学生が続いて多い。受講者の男女比については、多くの事例で男子の比率が高いという結果であった。また、比率については、男子比率が 8 割を超えるという回答も約 4 割(11/25)あった。

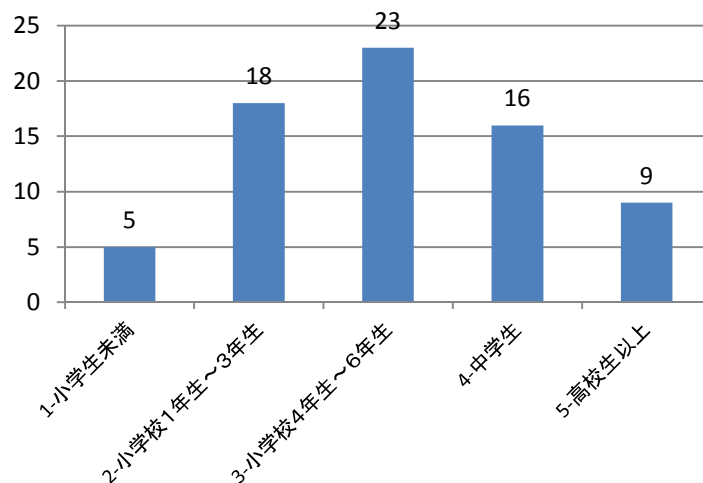


図 5-8 教室・講座の対象年齢（複数回答可）

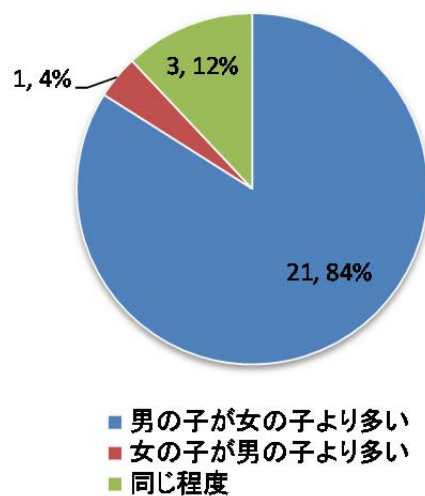


図 5-9 受講者の男女比

対象年齢については、ヒアリングにおいて以下のような意見が挙げられた。

- 小学校低学年は教えるのが難しいということもある。初めてパソコンを触ったという子も多く、導入指導も大変。
- 中学生は部活動と学習塾でスケジュールが埋められている。中学生向けに教室を展開するのは難しい。
- 小学校低学年だと抽象的な概念を理解できない。
- プログラミングを教える上で、関数名やパラメータが英語であることは小学生に教える上での課題。キーボード入力にも苦労することが多く、スペルミスなどが大きなビハインドになる。

小学校低学年以下については、パソコンの操作面（マウス操作やキーボード入力）の課題や、プログラムの関数や説明が英語であること、抽象的な概念を理解できないことなどから、指導の難易度は高くなるという意見が挙げられた。一方、中学生以上については、部活や他の習い事に時間を割かれるという意見が複数あった。これらの要員から、現状では小学校高学年が主な対象となっていると考えられる。

#### 5.4. 教育形式・教育コンテンツ

##### (1) 実施形態・受講形式

実施形態及び受講形式についてのアンケート結果を図 5-10 に示す。

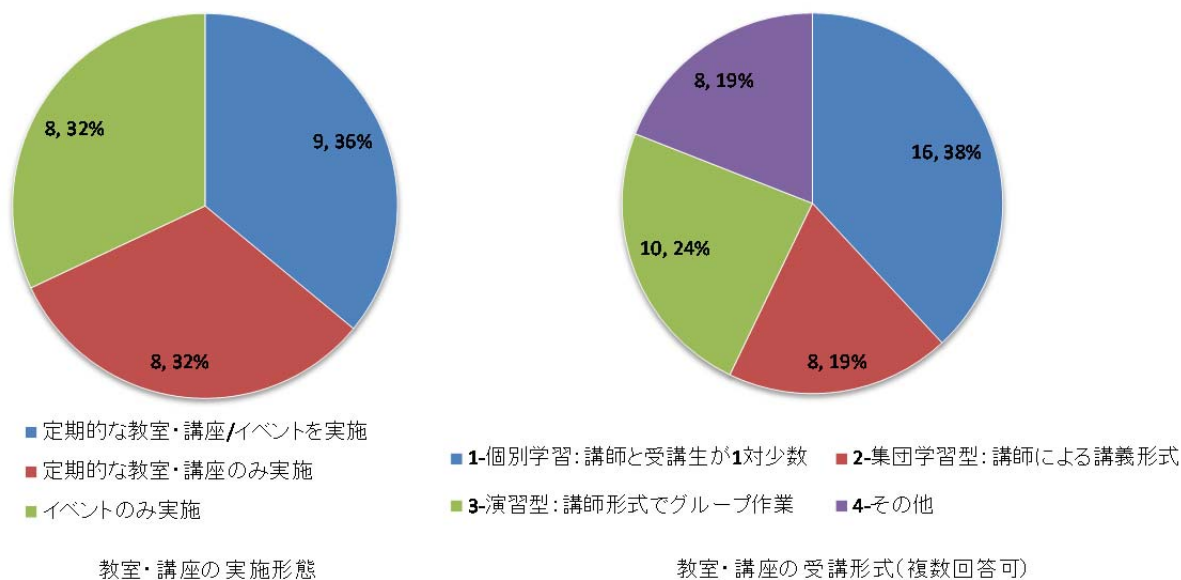


図 5-10 教室・講座の受講形式

図 5-10 に示す通り、定期的な教室・講座型のプログラミング教室・講座とイベント型のプログラミング教室・講座が同程度に開催されており、定期的な教室・講座型のみを実施している団体やイベント型のみを実施している団体も多い。



受講形式については、最も多い形式は個別学習型（講師と受講生が 1 対少数）であり、少数の受講生に対して講師がつく形式が多い。集団学習型（講師による講義形式）、演習型（講師形式でグループ作業）についても、事例の半数（9/18）は講師 1 人に対して受講者数は 5 名以下であり、講師 1 人に対して受講者数 8 名以下まで含めると約 9 割であった。

ヒアリングにおいても、Viscuit を用いたプログラミング教室・講座では講師 1 人当たり 10 人以上の受講者を担当する事例もあったものの、多くのケースでは講師 1 人に対し受講者 4 人程度であり、プログラミング教室・講座では 3~4 人に 1 人の形で講師が必要との意見も複数あった。

プログラミング教室・講座では、受講生に対して講師による手厚いサポートが必要であり、4 人程度の少人数に 1 人の講師が必要となっているといえる。

ヒアリングで挙げられた意見を以下に示す。

- 4~5 名に 1 人は講師が必要。
- プログラミング教育は人数が必要。コンテンツも時代のトレンドの移り変わりに沿って刷新する必要がある。
- プログラミング教育はスケールメリットが出ない。人を育てるのにも時間がかかってしまう。
- どの教室でも生徒 10 名に講師 3,4 名の体制。
- サポーターをいっぱいつけて、時間もたっぷり与えて実施している。受講者 15 人にサポーター 3 人と先生 1 人という体制。
- 生徒 3 人にメンター 1 人の割合である。

## (2) プログラミング言語

プログラミング教室・講座で利用されているプログラミング言語としては、Scratch が最も多く、続いて JavaScript や Java、レゴマインドストーム EV3 という結果になった。特にイベント型では Scratch を利用しているケースが多く、短時間での学習に適していると考えられる。一方、定期的な教室・講座では、JavaScript や Java、レゴマインドストーム EV3 も多く、より多様なプログラミング言語を用いて指導が行われている。なお、JavaScript では Web アプリ、Java では Android アプリを作っているケースが多い。そういった言語を対象とする教室・講座はソースコードの穴を埋める形式で教えるなどの工夫が行われていることが多く、Scratch の受講終了生の次のステップとして行われているケースもあった。

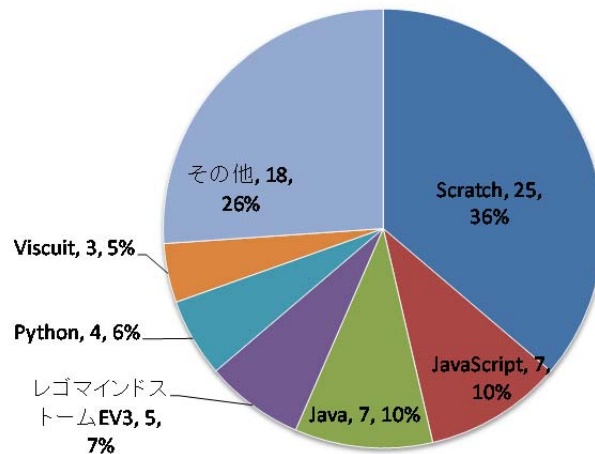


図 5-11 教室・講座で利用されているプログラミング言語（複数回答可）

### (3) テキスト教材

多くのプログラミング教室・講座ではオリジナルのテキスト教材を中心に指導が行われている。中心となるテキスト教材についてのアンケート結果を図 5-12 に示す。

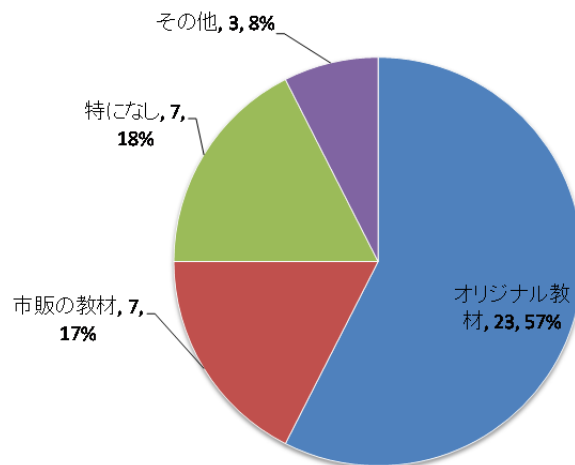


図 5-12 中心となるテキスト教材

過半数の教室・講座でオリジナルのテキスト教材を利用しており、市販のテキスト教材を利用しているケースは 2 割未満に留まっている。実際に青少年に教えることを目的としたような市販の教材がないことが原因として考えられる。一方、市販の教材を中心に教えている 7 事例のうち 6 事例が Scratch を利用する教室・講座であり、Scratch については市販の教材となる書籍が存在し、活用されていることがわかる。ヒアリングにおいても、Scratch は書籍が販売されているものの、他のプログラミング言語については小中学生などの指導に適した書籍がないという意見が複数あった。

オリジナルのテキスト教材については、図や写真、キャラクターなどを多用し、漢字・アルファベットにはふりがなを振るなど、子どもが抵抗なく取り組めるように工夫されて

いることが多く、コンピュータの操作やプログラミングの方法についても写真付きでわかりやすく示すなどの工夫がされている。また、テキスト型言語を用いる場合には、ソースコードの穴を埋める形式とするなどの工夫もみられた。オリジナルのテキスト教材のイメージを図 5-13 に示す。

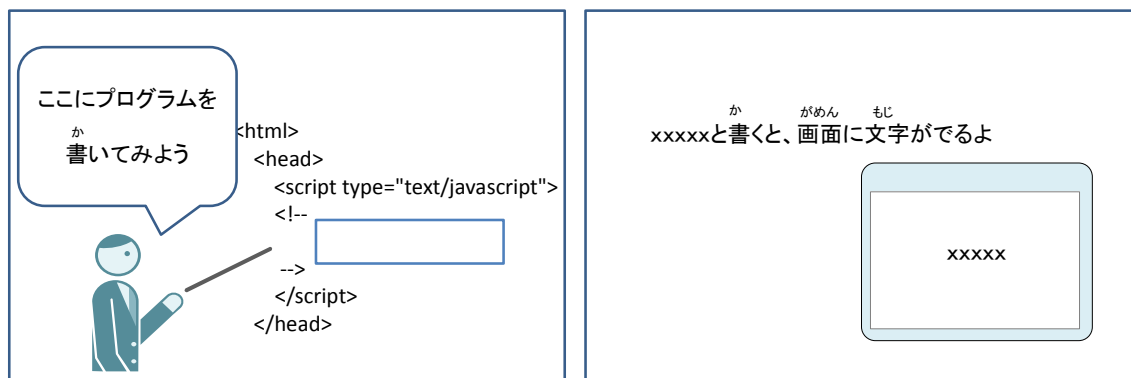


図 5-13 オリジナルのテキスト教材のイメージ

ヒアリングで挙げられた意見を以下に示す。

- レゴマインドストーム EV3 の教材は自分で作成している。EV3 自体は基本的には 10 歳から 21 歳が対象となっていて、小学生に教えるためにはわかりやすい教材をつくってあげる必要がある。
- 教材はすべて自分で勉強して作成している。
- Scratch は教材も書店で本が売っており、講師のコミュニティもあり、優れたツール。
- 子どもの取組みを見ながら、教材とノウハウを併せて展開するための教材を開発したいと考えている。

#### (4) 機器

機器については、多くのプログラミング教室・講座でパソコンを利用しているものの、RaspberryPi やタブレット、ロボットキット（レゴマインドストーム EV3 含む）等を利用している団体も一定数存在する。

使用している機器についてのアンケート結果を図 5-14 に示す。「その他」では、Arduino などのマイコンボードという回答も複数あった。

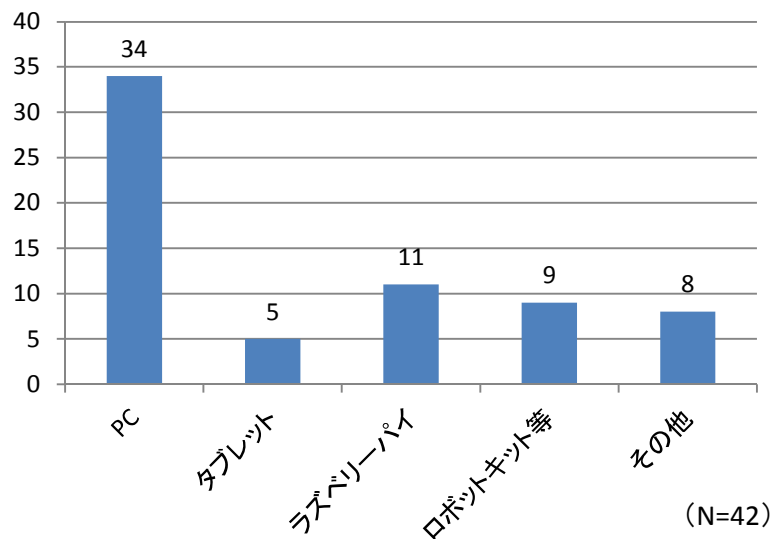


図 5-14 使用している機器（複数回答可）

## 6. プログラミングに関する教育がもたらす効果

プログラミングに関する教育がもたらす効果について、文献調査及び有識者ヒアリング、教育事業者ヒアリングの結果に基づき整理した。

### 6.1. プログラミングに関する教育がもたらす効果に関する研究の状況

#### 6.1.1. 子供向けのプログラミング言語の開発経緯

子供を対象とした学習ツールとしてのプログラミング言語の起源としては、Seymour Papert が開発した LOGO がある。LOGO では、「問題を解決する手段やプログラミング言語の習得ではなく、プログラミング（あるいはデバッグ）という行為そのものが子供たちの心的枠組み（schema）の構築につながるというアイデア」<sup>28</sup>が設計思想となっている。これは、「発達心理学者 Jean Piaget の構成主義（constructivism）を発展させた構築主義（constructionism）」<sup>28</sup>として知られており、人工物を構築するという作業が学習にとって重要という考え方に基づいている。Papert は、コンピュータを活用した学習環境の研究者でもあり、子供たちの動機づけにつながる学習ツールとしてプログラミングを捉えていたと考えられる。

また、「プログラミングを通して、子供たちが数学や科学を自ら学ぶことを目指した Papert」<sup>28</sup>に対して、Scratch を開発した「Resnick は、それらを含んだうえで、プログラミングを、単なる職業訓練ではなく表現の新しい形式や学びの新しい文脈と定義している」<sup>28</sup>。つまり、Resnick は表現力、創造力を高める学習ツールと捉えていると考えられる。

このように、子供を対象としたプログラミング言語は発達心理学の考え方を取り入れた学習ツールとして開発されてきた。

#### 6.1.2. 文献調査結果

プログラミングに関する教育がもたらす効果に関する論文は、定性的な評価に留まるものが多く、一部の論文では「課題解決力」<sup>29</sup>、「創造力」<sup>30</sup>、「コンピュータの原理に対する理解」<sup>31</sup>に効果があったという指摘がある。

なお、ヨーロッパの 20 か国のうち 12 か国は、既に学校の教育課程にプログラミング・コーディングが含まれており、7 か国が計画中である。学校の教育課程にコーディング／プログラミングを導入した根拠として、「論理的思考力の育成」、「コーディング・プログラミングスキルの育成」を挙げる国は 15 か国、「問題解決能力の育成」は 14 か国、「ICT 業界での雇用されるために必要な能力の育成」、「高等教育におけるコンピュータ科学専攻者の増

<sup>28</sup> 情報処理 Vol.56 No.4 「子どもの創造的活動と ICT 活用」、阿部和広、2015、p.350-354

<sup>29</sup> Electrical and Control Engineering (ICECE), 2011 International Conference “The learning effect of visualized programming learning on 6th graders'problem solving and logical reasoning abilities”、Ah-Fur Lai, Shu-Ming Yang、2011、p.6940-6944

<sup>30</sup> “The Effect of Logo Programming Language for Creativity and Problem Solving”、BENS PARADAMEAN, EVELIN, HONNI、2011、p.151-156

<sup>31</sup> 情報処理学会論文誌 Vol.44 「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」、兼宗進ほか、2003、p.58-71

加」は10か国、「他のキーコンピテンシーの育成」は8か国となっている<sup>32</sup>。

### 6.1.3. 有識者ヒアリング調査結果

有識者ヒアリングにおける意見を整理すると以下のようなになる。

プログラミングはその学習ツールとしての特性により、作業に対する結果がすぐにわかるため、子供の内発的な動機づけに優れているとの指摘、また、集中力や忍耐力が育まれるという指摘があった。また、コンピュータや情報科学の原理の理解につながることで、自ら課題を発見し、取り組み、解決するという問題解決能力、試行錯誤をする力、表現力や創造性など、21世紀型能力<sup>33</sup>が身につく、という指摘もあった。

一方で、プログラミングを学ぶことの効果については、プログラミングのどのような特性が教育効果に働くのか、プログラミングの効果ではなく、プログラミングに用いられるツールや教材の特性による特有の影響ではないのか等の検証が必要との指摘があった。

#### プログラミングは子供への動機づけ効果が高い

- プログラミングでは自分の作ったものに対する結果がすぐにわかるため、内発的な動機づけに優れている。

#### 集中力、忍耐力の向上

- インタープリター型言語では、反応がすぐに返ってくるため、取組に対する集中力、忍耐力などが身につく。

#### 情報科学の原理の理解、リテラシーの向上

- コンピュータや情報科学の原理の理解につながる。
- 将来、ICTを利活用していくための素養・リテラシーが身につく。

#### 21世紀型能力の向上

- 自ら課題を発見し、取り組み、解決するという課題解決能力、試行錯誤をする力など、21世紀型スキルに近い能力が身につく。
- 合理性や論理的認知を促進する。
- 表現力や想像力、創造性が向上する。

#### その他

- チームでの作業を行う場合には、協調性が育まれる。
- 積極性が向上する。
- プログラミングを学ぶ際には、学校での他教科の学習理解度とは異なる理解度や反応を見せる児童生徒がいる（プログラミングに適性を見せる生徒がいる）ため、児童生徒の活躍の可能性が広がる。

<sup>32</sup> "Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe", European Schoolnet, 2014, p.10

<sup>33</sup> 21世紀型能力とは、国立教育政策研究所が試案として出している能力・資質で、「生きる力」としての知・徳・体を構成する資質・能力から、教科・領域横断的に学習することが求められる能力を資質・能力として抽出し、それらを「基礎」「思考」「実践」の観点で再構成した日本型資質・能力の枠組みである。

#### 6.1.4. プログラミングに関する教育に関する評価指標

プログラミングに関する教育の効果の測定方法の事例については、日本では「授業ごとのアンケート、定期試験（客観テスト）、理解度アンケート、生徒作品の分析」<sup>34</sup>によって評価をしている例があった。また、プログラミング学習を目的にするのではなく、教科学習の理解を深めるための手段としてプログラミングを実施している場合（品川区京陽小学校）<sup>35</sup>には、教科学習の達成度により効果の測定を行っていた。また、有識者からの主な意見は以下のとおりであった。

- 現状では、海外においても、国内においてもプログラミングに関する教育の評価についての検討や研究はほとんど行われていない。
- プログラミングに関する教育の評価をするためには、教育の目的（例：「プログラミングを学ぶ」のか「プログラミングで学ぶ」のか、教育内容等）に応じた評価指標の設定が必要である。
- 論理的思考については、他の教育分野においても評価手法が確立しているとはいえない。例えば、論理的思考について、丁寧に評価するためには、評価手法として作業プロセスの観察が必要になるだろう。

一方で、教育事業者からは、以下のように評価指標の検討を期待する意見もあった。

- プログラミングに関する教育の効果について、多くの人にとっての共通理解となるようなメッセージを作り上げていく必要がある。
- 子どもにとってどのような点が効果的なのかについて、統一的なメッセージを出すための評価指標が必要なのではないか。

これらのことを踏まえると、プログラミングに関する教育の評価手法については、現状では十分に検討されていないと考えられる。一方で、プログラミングに関する教育の適切な評価を行うためには、プログラミングに関する教育の目的や育成する能力の定義を明確にする必要があるという有識者の意見が強かった。教育の目的や育成する能力の定義を明確にしたうえで、評価指標や客観テストを作成し、有識者や関係者の間で検討し、実証を経て最適化していくことが期待される。

## 6.2. プログラミングに関する教育がもたらす効果に関する教育事業者への調査結果

プログラミングに関する教育を実施する教育事業者に対するアンケート結果<sup>36</sup>では、プログラミング教室・講座によって得られる・伸びると思われる能力や意欲について、選択肢

<sup>34</sup> 情報処理学会論文誌 Vol.44「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」、兼宗進ほか、2003、p.58-71

<sup>35</sup> 品川区立京陽小学校第5学年国語科学習指導案、peg programming education gathering、<http://pegpeg.jp/pdf/keiyoschool5.pdf>、（最終アクセス＝2015年3月）

<sup>36</sup> 今回調査対象とした教育事業者は一人一台での学習が主であり、チームで一つのアプリケーション開発を行う形態の教育は実施していないため、リーダー育成に関する効果については言及がなかった。

にあげたいずれの能力や意欲についても効果ありとする事業者が多い。また、相対的には、「新しいものを生み出す創造的な力」を効果ありと回答する事業者が最も多く、次いで、「自ら学ぼうとする意欲」、「ものづくりの意欲」、「ものごとをやり遂げる粘り強さ」の順で、効果ありとの回答が多くなっている。

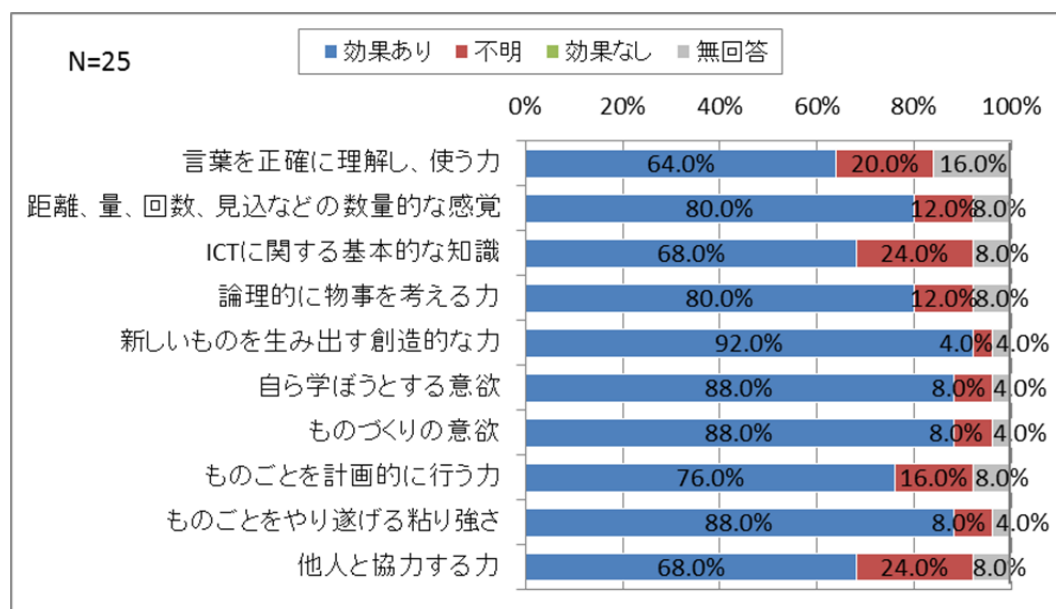


図 6-1 プログラミングに関する教育がもたらす効果

また、各能力等を伸ばすための学習段階として効果的だと感じられる年代（「就学前」、「小学校低学年」、「小学校高学年」、「中学校」、「高等学校」の5区分）について、各年代を対象とした講座・教室を実施している事業者に質問した。その結果、就学前では「新しいものを生み出す創造的な力」、「自ら学ぼうとする意欲」に効果が見られ、小学校低学年では「新しいものを生み出す創造的な力」、「自ら学ぼうとする意欲」、「ものづくりの意欲」に効果が見られた。また、小学校高学年～中学校では、これらに加えて、「距離、量、回数、見込などの数量的な感覚」「論理的に物事を考える力」「ものごとを計画的に行う力」「ものごとをやり遂げる粘り強さ」「他人と協力する力」に効果がみられた。

小学校低学年までは意欲や創造力、小学校高学年以降は、意欲や創造力に加えて、数量的感覚、論理的思考力、計画性や粘り強さなど、幅広い能力の向上に効果的だと考えられている。

教育事業者からのアンケート及びヒアリングにおいて、プログラミングにおける教育が効果的と考えられる点についての情報収集を行った。主な意見は以下のとおりである。

#### a.創造力の向上

プログラミングでは自由に作品を制作する課題を与えることが多いため、子供たちの作品制作を通じて、作りたいものを実現するという創造力が向上するという意見が多く得られた。

- つくるものを統制せず、作りたいものを実現まで導くことで創造性を発揮できる。
- 創造力がつく。作りたいものを作る、楽しむ、さらに作るという好循環が生まれる。



- 同じテーマの中で他者との発想の違いを感じ、その上でアイデアを相乗的に膨らませていく面白さや楽しさを体験できるため、子供の「共に創る力」を育む効果大きい。
- 小学校3年生から6年生までの間にプログラミングを経験させて、ゼロからものを作り出す面白さを実感させることで創造性が伸びる。受講生徒の年齢が上がるにつれて、ゼロからものを作り出すという創造性は弱くなり、その場における正解を推測して答えを提示しようとする傾向が強まる。

#### b.学習意欲の向上

プログラミングは制作物の実行と結果の確認ができることから、その活動自体が子供たちにとって楽しいものであるため、子供は積極的に活動に取り組むこと、数学的な知識を体感しながら学ぶことから、学習意欲が向上するとの意見が得られた。

また、デバッグやトライアンドエラーを繰り返して作品を完成させるというプログラミングの作業特性から子供たちがプログラミングの様々なタイミングで気づきを得るため、子供たちの忍耐力、集中力が持続し、学習意欲が向上するとの意見があった。

- ビジュアル言語では、未就学児や小学校低学年でも、2次元座標や角速度（回転）の概念がわかり、自分で思い通りにプログラミングする。プログラミングを利用することで、教科書を機械的に覚えるのではなく、自分で動かして納得する、という学びができる。
- プログラミング自体に楽しさがあるため、学習意欲の向上が認められ、自ら取り組もうとする意欲が向上する。結果として、受験や試験のための勉強ではなく、課題解決や楽しむための学びとなり、学習の意欲が根本的に変革できる。
- うまく動かない際のデバッグやしくみが分かった時の腑に落ちた感じなど、参加者それぞれのタイミングで色々な気づきや成長がある。
- トライアンドエラーを繰り返すことにより、忍耐力、集中力がつく。宿題だと30分も集中できない子供が3時間集中できている。

#### c.課題解決能力（論理的な思考力、計画性等）の向上

フローチャートや仕様書の作成とそれらに基づきプログラミングするという過程で、俯瞰的に考えたり、時系列で考えたり、仕組みを考えるなどの合理的、論理的な思考が必要となるため、論理的な思考力や課題解決能力が向上するとの意見が得られた。

- 可視化に優れ、見通しが立てやすいロボットを通して、「失敗をして、見通しをたててやりなおす」ことによって、論理的思考力を学ぶ。
- 流れ図を用いることで、俯瞰的に時系列で物事を見ることができるようになる。
- 手続き型言語を扱うため、順番の重要性が伝わる。
- 論理的に物事を考える力が育つ。
- 作りたいもの（仕様書）を具体的に再現する手段を考えることで、論理的な思考力の向上に繋がる。
- フローチャートを書けるようになるため、論理的思考力が伸びる。
- 課題発見力や課題解決力が身につく。

#### d.表現力の向上

仕様書の作成やプログラミングの過程を通じて指導者や受講生同士で相談が必要になる。子供が自分の考えを説明するという活動を通じて、表現力の向上が図られるとの意見があ

った。また、完成した作品の発表を通じて、子供のプレゼンテーション能力が図られるという意見が得られた。

- 頭の中にあるイメージを紙に描くことで、仕様書ができ、それを元に作成すると、表現力の向上につながる。
- やりたいこと、知りたいこと、考えていることなどを、言葉で相手に伝える能力も身につく。
- 作品の発表の場を通じて、子供たちの表現力、プレゼンテーション能力が向上する。

#### e.行動力の向上

プログラミングの過程で子供たち同士で相談したり、教えあい、学びあうことを通じて、目的達成のために前に進む主体的な行動力が身につくとの指摘があった。

- 子供たち同士で教え合うことを身に付け、受け身ではなく、主体的になり、行動力に繋がる。

#### f.情報活用能力、プログラミングスキル、英語力の向上

プログラミングにおいて、テキスト型言語を取り扱う場合には、アルファベットや英単語に触れる機会があるため、英語力に好影響があるとの意見があった。

また、プログラミングの過程で、不明点を調べたり、既存のライブラリを利用するなど、作業を行う際にインターネットによる情報検索を行うため、情報検索能力の向上など、情報活用能力が向上するとの指摘があった。

また、プログラミングスキルそのものが向上するという意見があった。

- テキスト型言語を使っているため、ある程度は英語力にも良い影響がある。
- 作業の過程や作品公開を行うことを通じて、情報モラルが身につく。
- プログラミングスキルと言語力や検索力（情報利活用力）が向上する。
- アルファベットや英語に慣れる効果がある。

#### g.理科好きの育成

プログラミングは理科好きを育成する効果があるとの指摘があった。

- 子供が科学を学ぶツールとしてプログラミングを活用できるようになる。
- 理科が好きになる子供が増える。

#### h.卒業生の活躍

プログラミングを学んだ生徒がNPOの立ち上げ等で活躍し、また、理系の学校への進学実績があることなどの指摘があった。

- 卒業生がプログラミングに関する教育のNPOを立ち上げたり、ワークショップを主催している。
- スクラッチカンファレンスに出席したり、スクラッチデイ東京のミーティングに参加するなど、主体的に活動できる人材が育っている。
- 卒業生が高等専門学校に進学したり、理系の進学校に進学するなど、高度な人材が育成できている。

#### i.その他

プログラミングの作業過程において、短時間でトライアンドエラーを行いやすいという

特性があるため、繰り返し取り組むことが容易であり、これまでに挙げたような教育効果が生じやすいという意見があった。

また、プログラミングは芸術やスポーツと同様の活動だと捉えるべきであり、天才的なプログラマーを育成するためには、若年層から学ぶことが必要になるのではないかとの指摘があった。

- プログラミングは、トライアンドエラーが早いため、成長速度が速い。
- 音楽等と同様に、天才的な域に到達するには子供のころから学ぶ必要がある。

各教育事業者が実施する事業の目的によって、目指す人材像や感じている効果は異なるものの、プログラミングの有する様々な特性に応じた効果が得られていた。プログラミングの特性別にプログラミングに関する教育の効果を整理すると、以下のとおりである。

表 6-1 プログラミングの特性とプログラミングに関する教育の効果

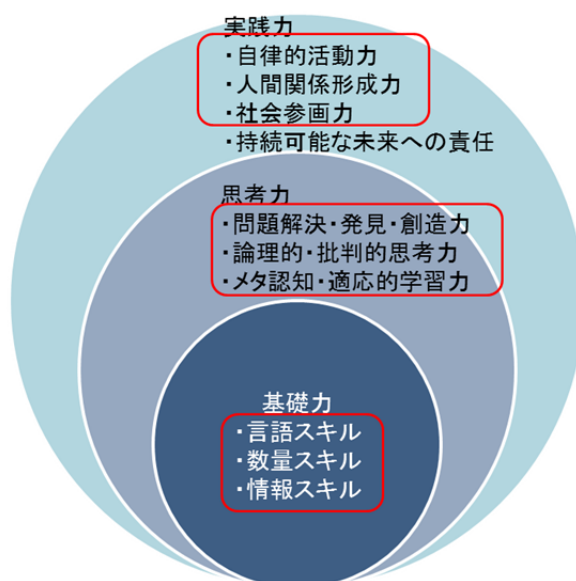
プログラミングの特性	プログラミングに関する教育の効果
ICT を利用する	<ul style="list-style-type: none"> <li>・プログラミング能力の向上</li> <li>・検索力等の情報利活用力の向上</li> <li>・コンピュータに関する知識の理解</li> </ul>
フローチャートや設計図、仕様を考える	<ul style="list-style-type: none"> <li>・論理的思考力の向上</li> </ul>
インタープリター型言語においては、試行錯誤が容易である（結果がすぐにわかり、改善点が明確である）	<ul style="list-style-type: none"> <li>・課題解決能力の向上</li> </ul>
モノづくりである	<ul style="list-style-type: none"> <li>・完成させる、やりきる力（忍耐力、集中力）の向上</li> </ul>
プログラミング自体及び成果（作品）が子どもにとって魅力的である	<ul style="list-style-type: none"> <li>・プログラミングや関連スキル（情報利活用力、語学力等）に対する学習意欲、関心、動機の向上</li> </ul>
チームでの活動を設定できる	<ul style="list-style-type: none"> <li>・主体的な行動力の向上</li> <li>・自己・他者肯定感の向上</li> <li>・発表力の向上</li> <li>・チームでの活動力の向上</li> </ul>
プログラミングはツールであり、多様な目的で活用できる	<ul style="list-style-type: none"> <li>・より多くの子どもへの関心・意欲、動機付けに有効</li> <li>・創造力の向上</li> <li>・表現力（デザイン力）あるいは論理性の向上</li> </ul>
その他	<ul style="list-style-type: none"> <li>・早期教育による天才的プログラマーの育成可能性向上</li> </ul>

### 6.3. プログラミングに関する教育がもたらす効果

現在、実施されているプログラミング教室・講座は、プログラミング体験自体を目的としていたり、プログラミングを楽しみ、創造力を養うことを目的としていることが多かった。一方で、有識者や教育事業者からは、プログラミングに関する教育の効果はあるかもしれないが、それをプログラミングの目的とすべきではないとの指摘が多くあった。効果を得ることを目的とすると、プログラミングが他の教育手法よりも効果的であるかどうか

を問われることになり、結果として、プログラミングに関する教育の推進を阻害することを危惧する指摘であった。

諸外国におけるプログラミング／コーディング教育においては、主に高度 ICT 人材育成を目的とする「ICT 業界での雇用されるために必要な能力の育成」、「高等教育におけるコンピュータ科学専攻者の増加」よりも、「論理的思考力の育成」、「コーディング・プログラミングスキルの育成」、「問題解決能力の育成」など、主に、21 世紀型能力の育成を目的としている国が多いことが明らかとなった。実際に、下図のように 21 世紀型能力で示されている能力と、有識者や教育事業者に対するヒアリングで得られたプログラミングに関する教育の効果を比較すると、21 世紀型能力として示された能力のうち、一定の範囲をカバーしていることが確かめられた。



※赤枠はプログラミングに関する教育で効果があるとされたスキル

図 6-2 21 世紀型能力<sup>37</sup>（出典：国立教育政策研究所試案をもとに弊社加工）

また、プログラミングに関する教育がもたらす効果について、学説または有識者の意見と教育事業者の見解の対応関係を表 6-2 に整理した。同表から、学説または有識者の意見と教育事業者の見解は、概ね整合していることが確認できる。

このようなプログラミングを通じた人材育成をより効果的なものとするためには、目的に照らしてプログラミングのどのような特性を活用するのかを検討し、適した教育手法、ツール、教育プログラムを選択していくことが必要と考えられる。

<sup>37</sup> 21 世紀型能力は、①思考力を中核とし、それを支える②基礎力と、使い方を方向づける③実践力の三層構造から成り立つ。

表 6-2 プログラミングに関する教育がもたらす効果

学説または有識者の意見	事業者の感じる効果
創造力 <sup>38</sup> の向上	<ul style="list-style-type: none"> <li>・プログラミングでは自由に作品を制作する課題を与えることが多いため、子供たちの作品制作を通じて、作りたいものを実現するという創造力が向上する。</li> <li>・ものを作り出す面白さを実感させることで創造性が伸びる。</li> </ul>
課題解決力 <sup>39</sup> の向上	<ul style="list-style-type: none"> <li>・プログラミングでは、結果がすぐにわかり、改善点が明確であることなどから、トライアンドエラーを繰り返しやすく、課題発見力、解決力が身につく。</li> <li>・プログラミングを完成させるという目的達成のために前に進む主体的な行動力が身につく、完成させ、最後までやりきる力が身につく。</li> </ul>
表現力 <sup>40</sup> の向上	<ul style="list-style-type: none"> <li>・プログラミングの過程で、プログラムの構想を書いたり、受講者同士で教えあい、学びあいをしたり、作品を発表したりすることによって、表現力が向上する。</li> </ul>
合理性、論理的思考力 <sup>41</sup> の向上	<ul style="list-style-type: none"> <li>・フローチャートやプログラムの構想の作成とそれらに基づきプログラミングするという過程で、俯瞰的に考えたり、順序立てて考えたり、仕組みを考えるなどの合理的、論理的な思考が必要となるため、論理的な思考力が向上する。</li> </ul>
意欲の向上（内発的な動機づけ効果） <sup>42</sup>	<ul style="list-style-type: none"> <li>・プログラミングという活動自体が子供たちにとって楽しいものであるため、子供は積極的に活動に取り組む。</li> <li>・デバッグやトライアンドエラーを繰り返して作品を完成させるというプログラミングの作業特性から、子供たちがプログラミングの様々なタイミングで気づきを得るため、子供たちの忍耐力、集中力が持続し、学習意欲が維持、向上される。</li> </ul>
コーディング・プログラミングスキル <sup>40</sup> の向上	<ul style="list-style-type: none"> <li>・テキスト型言語を用いて高度なプログラミングを行ったり、大人と同様の習熟を見せるなど、子供であってもプログラミングスキルが向上する。</li> </ul>
コンピュータの原理に関する理解 <sup>43</sup>	<ul style="list-style-type: none"> <li>・プログラミングの過程で、不明点を調べたり、既存のライブラリを利用するなどの作業を行う際にインターネットによる情報検索を行うため、情報検索能力の向上など、情報活用能力が向上する。</li> <li>・プログラミングの活動を通じて、コンピュータの性質・原理に対する理解が深まる。</li> </ul>

<sup>38</sup> Scratch 開発者の Resnick や論文“The Effect of Logo Programming Language for Creativity and Problem Solving”にて指摘。

<sup>39</sup> “The learning effect of visualized programming learning on 6th graders'problem solving and logical reasoning abilities”にて指摘。

<sup>40</sup> Scratch 開発者の Resnick が指摘。

<sup>41</sup> “Computing our future Computer programming and coding - Priorities, school curricula and initiatives across Europe”にて指摘。

<sup>42</sup> Logo 開発者の Papert が指摘。

<sup>43</sup> 「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」にて指摘。

## 7. プログラミング人材に関する段階的な育成手法

プログラミング人材に関する段階的な育成手法について、有識者ヒアリング、教育事業者ヒアリングおよび文献調査の結果に基づき整理した。

### 7.1. 段階的な育成方法

#### 7.1.1. 基礎的能力の発達に基づく段階的育成方法

青少年のプログラミング人材育成においては、青少年の認知能力などの基礎的な能力の発達に従って無理の無い段階的育成を行うことが重要である。

プログラミングにおいて必要とされる主要な能力の一つとして、論理的思考力、抽象的思考力が挙げられる。下記に示す有識者の意見および学説から、論理的思考、抽象的思考が身に付く前は、IF-THEN 型の論理構成は難しいため、論理的な組み上げを必要とするプログラミングは難しく、具体的な視覚的な操作による育成が妥当と考えられる。

(ヒアリング結果、文献情報より)

- IF-THEN 型の論理構成は 9 歳 10 歳まで理解できない。また、論理的思考は、11 歳以降でなければ理解するのは難しい。
- 論理的な考え方、抽象的な能力が限られているので、それが備わってからやるべきである。Scratch の主なターゲットは 8 歳以上である。
- 11,12 才以上から、抽象的一般的な形で論理形式的に考えることができるようになる形式的操作期 (operational stage) としている (Jean Piaget<sup>44</sup>)。
- フローチャートの学習は小学校 5 年生以上になってからでないと難しい。よほど優秀な場合は小学校 4 年生でもできる場合もあるが、小学校 3 年生以下にはまず不可能である。

論理的な思考力を身につける年齢には個人差はあるが、上記の有識者の意見では、9 才頃以降が多数を占める。ただし、海外においては、イギリスなど一部の国において、5 才から義務教育でプログラミングが取り入れられ、たとえば、5 才では、ボタン操作による動きを操作する Roamer<sup>45</sup>を用いた教育や 6~7 才では簡単な Scratch を用いた教育など、簡単な論理を必要とする教育が行われている<sup>46</sup>。このことから、簡単な論理を必要とする教育は、9 才以前に開始することも考えられるが、6 章に示す通り、現時点では、効果に関する評価データは十分には得られていないため、本格的な論理構成を必要とする教育は、9 才頃以降とし、今後、教育の実践と評価を通じて開始時期を調整する必要があると考えられる。

論理的思考力が身につく 9 才頃以降になると、プログラムの条件分岐や繰り返しなどの制御に係わる構成が必要となるプログラミングを指導することが妥当と考えられる。

ただし、下記のヒアリングにおける意見などから、テキスト型言語は、実現したいものを記述するために、文法やタイピングなどの習得が問題となり、使えるようになるまでに

<sup>44</sup> 発達論の研究者 J.ピアジェ (Jean Piaget) (2015 年 3 月アクセス)

<http://hermes321.com/developmental-theory/researcher/jean-piaget/>

<sup>45</sup> ボタン操作で、動作を指示する亀型のおもちゃ。

<sup>46</sup> 海外調査 2014 ~英国・エストニア~ 報告書 (一般財団法人 国際 IT 財団、2015 年 2 月)

時間がかかるため、ブロックの組合せなど視覚的な支援を伴う論理構成が可能な Scratch などを用いることが有効であると考えられる。

(ヒアリング結果、文献情報より)

- ビジュアルプログラミング言語を用いることにより、早い段階で、どのようなプログラムを作るか(What)を考えてからプログラムを作成することができるようになる。テキストベースの言語の場合、タイピングや文法など(How)が問題となり、思いついたものをプログラムとして落とし込めるようになるまでに時間がかかる。
- テキスト型言語では、文法上のミス等コーディングが難しい。自宅に帰って子供だけで作成することが難しかった。
- 文法エラーの生じないビジュアルプログラミング（文字入力とプログラミングの間に本質的な関係はなく、文字入力に習熟していない段階での苦痛を軽減する）
- タイピング能力が無ければ iPhone アプリ開発、Web アプリ開発コースは難しいため、タイピングを自分でできる子に限っている。

このようなことから、テキスト型プログラミング言語の文法やアルファベットの利用等を考えると、そのような文字の利用に習熟してからテキスト型言語を利用することが好ましいと考えられる。ただし、文字の利用の習熟は、個人差や経験による差が大きく、比較的早い年齢から開始することにより、年齢が低いうちに習熟し、テキスト型言語のステップアップを早期に進められる場合もある。たとえば、ある民間事業者では、小学1年生～6年生に Scratch ゲーム開発コースを提供し、小学3年生～6年生に iPhone アプリ開発コースを提供するなど、テキスト型言語を教える対象年齢に差が見られる。

論理的思考力を身につけ、テキスト型言語にも習熟した後は、年齢にあまり関係なく、プログラミングの知識とスキルに応じて指導内容をステップアップすることが可能となる。

以上の点を整理すると、段階的な育成方法に影響を与える能力および知識等の主要要素としては、以下のような点に整理することができる。

- 論理的思考力、抽象的思考力等の認知能力
- アルファベット等の文字利用の習熟
- プログラミングに関するスキル・知識

これらに基づき、段階的な指導方法について区分を行い、それぞれについてヒアリングおよびアンケート結果に基づき、典型的な指導方法や主な教材とツールを整理すると以下の通りである。なお、下記の典型的な指導方法は、相互に組合せて実施されるものもあり、排他的な分類ではない。

表 7-1 段階的指導の区分と典型的な指導方法

段階区分	典型的な指導方法（特徴）	主な教材・ツール
(1) 論理的思考が 身に付く前 (3才頃から9 才頃)	(a)プログラミング体験型 グラフィックス操作を試しながら、その結果の動きを 確認し体験を通じてプログラミングに親しむ。	Viscuit（タブレット） 等
	(b)創造性育成型 簡単な操作で利用できるビジュアルプログラミング環 境を自由に使い、アニメーションの創作などを通じて、 創造性を養う。遊びや表現を通して楽しみながら学ぶ、 個性を伸ばす。	Viscuit（タブレット） 等
(2) 論理的思考が できる時期 (9才頃以降)	(a)自主性育成型 論理的構成を必要とするビジュアルプログラミング環 境を用いて、自主性を重視し、指導者は基礎的なサポ ートのみ行い、自由にプログラムを作成させる。	Scratch 等ビジュアル 言語、EV3 レゴマイ ンドストーム等
	(b)スキル習得型 Scratch 等を用いて(1)教材学習、(2)オリジナル開発、(3) 発表会などのステップアップによりビジュアルプログ ラミングを学ぶ。スクール型、複数回のイベント型な どがある。カリキュラムと教材を体系化（オリジナル テキスト、市販テキスト）している。	Scratch 等ビジュアル 言語、
	(c)課題設定・解決力育成型 作りたいモノや目標を最初に具体化し、その後、Scratch などのツールを用いて実現することにより、問題設定 力と段取り力を身につける。	Scratch 等ビジュアル 言語、設計図・企画 シート
	(d)協調性・表現力育成型 子供の交流、子供同士の共同作業を重視し、ファシリ テーターはサポート中心。成果の発表により協調性や 表現力を育成する。	Scratch 等ビジュアル 言語、Viscuit
	(e)忍耐力・解決力育成型 問題に突き当たっても、子供が自ら試行錯誤により解 決策を見出せるようなサポートをすることで、忍耐力、 解決力を養う。	Scratch 等ビジュアル 言語、EV3 レゴマイ ンドストーム等



(3) テキスト 型言語が 使える時期	<u>(a)スキル習得型</u> (1)基礎概念の習得、 (2)全体構成と実践 (3)オリジナルアプリ開発 の3ステップにより指導する。	Java、Objective-C、 HTML5 等テキスト 言語
	<u>(b)モチベーション向上型</u> 指導員（メンター）とのコミュニケーションや、アプリ・リリース、アプリ甲子園など外部とのつながりを通じて、モチベーションを高める。	Java、Objective-C 等 テキスト言語
	<u>(c)課題設定・解決力育成型</u> どのようなプログラムを作るか(What)を考慮してから、どのようにプログラミングするか(How)を考慮のようにする。	テキスト言語

(3)においては、テキスト型プログラミング以外にも、Microsoft Robotics Studio などビジュアルプログラミングの方向に発展させることも考えられ、(3)以降については、本人の志向や関心に応じて多様な選択肢が提供されることが望ましい。

教育に必要な指導員の人数は、論理的思考が身につく前の段階で、Viscuit を用いた方法では、簡単な操作を教えた後、指導員は生徒 10 人に 1 人程度でも可能なケースもあったのに対し、論理的な思考を身につけた後の段階で、Scratch などによる論理構成を伴う教育においては、生徒 4 人に講師 1 人といった違いが確認された。

とくに小・中学生の子どものモチベーションを高める方法として、調査で得られた以下のような工夫も参考となる。

<u>動きを伴う教材の活用</u> <ul style="list-style-type: none"> <li>・ Scratch のあとにロボット、LED ライトの点灯等コンピュータ画面だけでなく、何かを物理的に動かすカリキュラムを検討している。</li> <li>・ PC 内だけのプログラムにとどまらず、さわられる物、見られる物を利用したプログラム作りを意識している。</li> </ul> <u>モチベーションを引き出す指導</u> <ul style="list-style-type: none"> <li>・ 自己学習を促すために、時間内では少し欲求不満気味させ終わらせる。</li> <li>・ より子ども目線に立ち、「こんなものを作ってみたい！」といった純粋な気持ちを大切に して、没頭したくなるような好奇心であふれた空間を用意している。</li> <li>・ 子どもたちの中で学びがその先の生活や社会と結びつくように、体系立った学問から学ぶよりも、ワクワクするようなモノづくりの中で気づいたら学んでいるような、“遊ぶように学べる”新しい教育の形を目指している。</li> </ul>
---

また、育成方法に関する考え方として、調査で得られた以下の意見は参考になる。

#### 個性の尊重と創造性の育成

・画一的な価値観よりも、多様性や自分らしさを尊重する。周囲と比較するのではなく、自分の個性や興味を大切にして、新しくオリジナルのモノを作る力を養っていく。作るペースも、作る作品の内容も、子どもが自由に取り組めるように設計することで、個々の才能や創造性を最大限に引き出し「出る杭が伸びる」ような環境を追求していく。

・自ら考え、作り、伝える力を育む

#### 自ら考え、成果を伝える力を育むこと

・自ら主体的に考え、アイデアを形にして、それを周囲に伝えていける力を子どもたち自身に身につけることを目指している。そのために、手段を学ぶだけでなく、新しいものを自分で生み出し、形にするための試行錯誤を経て、最後にそれを周囲に発信する経験まで行う。その経験を通じて、子どもたちが将来未知の問題に直面した時に、自らの力で乗り越えられるような力を育てていく。

5章に示した通り、生徒の男女比は、男子の多いクラスが8割以上であるが、ヒアリング結果からは男女による関心の違いも以下のように確認できた。

- ・男子は（プログラムの）ルールにこだわり、女子はキャラクターやデザインをつくりこむ。
- ・男子は動きにこだわり、女子は絵を描きたがる。

このようなことから、男女の教育においては、課題設定や教材の選択肢において、それぞれの関心の高いものを選択できるメニューを用意することも有効と考えられる。

### 7.1.2. プログラミングスキルに応じた段階的育成方法

論理的思考・抽象的思考などの認知能力が一定レベルに達すると（中学生以上）、年齢に関係なく、プログラミングスキルなど習得知識に応じて、段階的に指導が行われる。

そのような場合の教育コースは、以下のようなステップアップによる指導が典型的である。

- (1)基礎概念の習得
- (2)全体構成の理解と実践
- (3)オリジナル創作

ヒアリング等から得られた情報では、教育コースおよび教材を以下のように体系的に整備し、順序立てて教育している典型は表 7-2 のようになる。

表 7-2 「段階 2」の時期における段階的指導方法の典型

	指導内容	コースの例
(1) 基礎概念の習得	(1) アプリの体験	<u>サンプルアプリ（時計アプリ等）の体験</u> サンプルアプリに自分の好きな画像やデザインを使って、自分用のアプリを作る。
	(2) 変数の習得	<u>カウンターアプリ（変数の理解）</u> ボタンを押すと数値が増減するカウンターアプリ、電卓アプリなど身近なものがどういう仕組みで動いているか理解しながら、アルゴリズムを学ぶ。
	(3) アプリのカスタマイズ	<u>カスタムアプリの開発（自分のアイデアを盛り込んだアプリ開発）</u> アプリサンプルコードを自分のアイデアを盛り込んだオリジナルアプリにカスタマイズする。
(2) 理解と実践力 全体構成の	(1) アプリの構成	<u>アプリ全体の構成（参加回数 2～4 回以上）</u> 要素を組合せて全体のアプリを構成する。 iPhone 用の Xcode、Objective-C を用いて開発する。
	(2) アプリ構成の実践	<u>利用経験のあるアプリの構成（参加回数 2～4 回以上）</u> お絵かきアプリ、Twitter アプリなど、利用経験のあるアプリを、部品を活用しながら構成する。
(3) オリジナルアプリの作成	オリジナル作成	<u>ゼロからオリジナルアプリの開発（参加回数 3 回以上）</u> 開発したいアプリの企画から考え、その後、設計、実装、デバッグまでスタッフのサービスを受けながら行う。

これらのコースは、iPhone, Android, Windows8 などプラットフォームに応じたアプリプログラミング、デジタルアート、WEB 製作、ゲーム開発など目的に応じてそれぞれ体系化している例もある。

論理的思考力およびテキスト・文法力に習熟した高学年における教育においては、関心の違いが大きくなるため、一律の公教育ではなく、関心の違いに応じて、生徒に適した教育内容によりセグメント化しするとよいとの考え方があがる。

## 7.2. 高度 ICT 人材育成につながる取組み

本節では、段階的育成方法の次のステップとして、参考情報として、高度 ICT 人材に向けた育成方法について特徴的な取組みを紹介する。

スキルや知識に応じた段階的育成方法としては、通学形式の指導・オンラインコースというような形態でステップアップする事例が複数あった（図 7-1）。さらに、大学 AO 入試やコンテストの開催、アプリのストアリリース、企業就職マッチング等がある（図 7-1）。「大学 AO 入試」や「コンテスト」は、青少年のモチベーションを高め、競い合うことでス

スキルを高める。また、リリース<sup>47</sup>できるようなアプリを作ることを目標とすることによって、モチベーションや向上心が高められると考えられる。更には優秀な学生をパートナー企業へ就職するための橋渡しをする取組みを行っている教育関連団体もある。

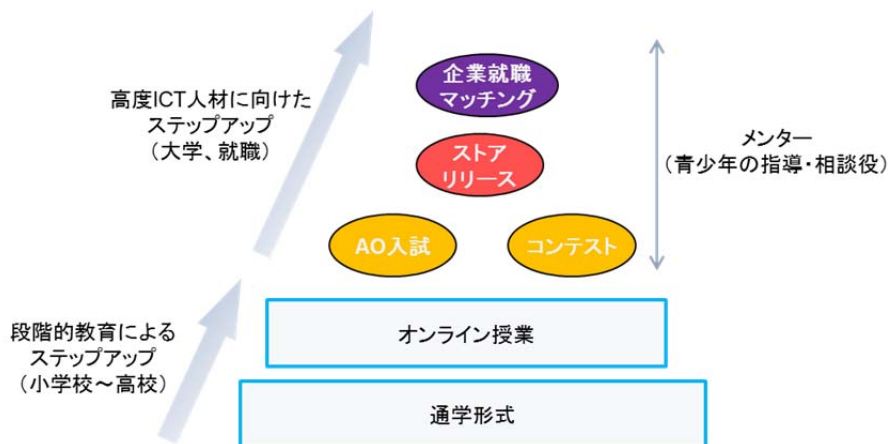


図 7-1 段階的教育および高度 ICT 人材育成につながる取組み

ヒアリングから得られた以下のような意見から、プログラミング教室受講生だけでなく指導者側の高専生・大学生等も高度 ICT 人材や地方人材育成、地方産業活性化に繋がる可能性を持つ。

- ・ 地元の高校生、高専生、大学生が指導者（講師、メンター、アシスタント等）になっている（多数）
- ・ アプリハッカソンを開催し、地元の高度な IT 技術者と大学生がコミュニケーションできる場となった。地域の企業に就職する機会になる。

さらに、民間教育事業者によっては、能力が高く、個性が光る大学生をメンターとして選抜し、数ヶ月以上におよぶ教育を行うことにより講師の質を高めている場合もある。このような取組みはプログラミング人材育成の質を高める取組みにもなる。これらによって育成されたメンターが、図 7-1 のような取組みを通じて活躍する例があり、高度 ICT 人材につながるキャリアパスとして成功モデルとなっている。

<sup>47</sup> 作成したアプリを、インターネット上で、一般の人が利用できるように公開すること。

### 7.3. 学習ツールの特徴

プログラミングに関する教育の学習ツールとして、プログラミング言語とハードウェア（操作端末類）がある。これらの概要については4章に記載されている。ここでは各ツールをプログラミング人材育成目的で使用している事業者へのヒアリング結果をもとに、長所や改善点についてまとめた。

#### 7.3.1. プログラミング言語の特徴

事業者は教育の目的に応じて、複数の言語の指導方法を検討し、選択しているため、特徴を網羅するものではないが、アンケート、ヒアリングにおける意見に基づき各プログラミング言語の強みと弱みを表7-3に整理した。

特に小中学生を対象とした教育目的で使用される言語としてはScratchを選択している事業者が最も多かった（P29 図5-11）。Scratchは教育言語として開発されているため、小学生でも比較的容易にプログラミングを楽しめる工夫がされている。また、言語の開発だけでなく、Scratchで作成したゲームを誰でも簡単に世界中で共有することのできるコミュニティサイトの評判も高かった。一方で、テキストプログラミングへ繋げるにはギャップがあるとする意見もあった。

表 7-3 ヒアリングで得られた各言語への意見

言語	区分 ※1	ブ ※2	強み	弱み (課題等)
Viscuit	V	○	<ul style="list-style-type: none"> <li>・自作イラストによるアニメーションやゲーム作成などが容易</li> <li>・幼児から遊ぶことができる</li> </ul>	<ul style="list-style-type: none"> <li>・テキストプログラミングに発展させる際にギャップがある。</li> </ul>
Scratch	V	○	<ul style="list-style-type: none"> <li>・文法ミスを抑えるためにブロック言語になっているが、高度な論理制御等も可能</li> <li>・世界中に利用者がいるため、コンテンツや情報が豊富</li> <li>・作品の共有が容易</li> <li>・市販のテキスト教材がある</li> <li>・低年齢向けの Scratch Jr.が提供されている</li> </ul>	<ul style="list-style-type: none"> <li>・ある程度の制御や論理の理解が必要</li> <li>・テキスト型言語への変換ができない</li> </ul>
Smalruby	V		<ul style="list-style-type: none"> <li>・Ruby を子供向けにブロック言語とし、文法ミスを防止</li> <li>・Ruby への変換が可能</li> </ul>	<ul style="list-style-type: none"> <li>・Scratch 等に比べ利用者は多くない</li> <li>・イラストの取り込みはできない</li> </ul>
プログラミン	V	○	<ul style="list-style-type: none"> <li>・ブラウザで動作するため導入し易い</li> </ul>	<ul style="list-style-type: none"> <li>・作品共有サイトが十分管理されておらず、多様な意見が混在している</li> </ul>
レゴマインド ストーム EV3 (言語部)	V		<ul style="list-style-type: none"> <li>・ロボット操作を伴うため、子どもの関心を得やすく、可視化に優れる</li> </ul>	<ul style="list-style-type: none"> <li>・導入するには機器が高価</li> <li>・10～21 歳が主な対象となっているため、低学年向きの教材を作成する必要がある</li> </ul>
Basic	T		<ul style="list-style-type: none"> <li>・安価な PC (IchigoJam 等) 上でも軽快に動作する</li> <li>・歴史が長いため扱えるシニア層にも指導できる人材が豊富</li> </ul>	<ul style="list-style-type: none"> <li>・テキスト型言語ならではの文法ミスのフォローが必要</li> <li>・機能の進化が少ない</li> </ul>
Objective-C (Swift)	T		<ul style="list-style-type: none"> <li>・iPhone アプリを開発できるため、モチベーションに繋がる</li> </ul>	<ul style="list-style-type: none"> <li>・開発するには Mac が必要</li> </ul>

※1 プログラミング言語の区分 (V: ビジュアル型・ブロック型プログラミング言語、T: テキスト型プログラミング言語)

※2 ブラウザ上での動作

### 7.3.2. ハードウェアの特徴

PCと比較すると、タブレット端末については直観的な操作で低学年にも使いやすいという特徴が挙げられた。また、RaspberryPiやIchigoJamについてはPCと比べて安価という特徴が挙げられ、イベント型の教室・講座で導入しやすい機器と言える。アンケート、ヒアリング結果を中心に、各ハードウェアに対する意見についてまとめると以下の通りである。

表 7-4 ヒアリングで得られた各ハードウェアへの意見

ハードウェア	強み	弱み
PC	<ul style="list-style-type: none"> <li>汎用性の高いツール</li> <li>画面サイズが最も広い</li> <li>多くの言語の開発環境に対応</li> </ul>	<ul style="list-style-type: none"> <li>高価</li> <li>セッティングが難しい</li> <li>キーボードやマウス入力ができる必要がある</li> </ul>
タブレット端末	<ul style="list-style-type: none"> <li>直観的な操作</li> <li>低学年の子どもでも使いやすい</li> </ul>	<ul style="list-style-type: none"> <li>画面が狭い</li> <li>比較的高価</li> </ul>
シングルボードコンピュータ	<ul style="list-style-type: none"> <li>安価（本体のみなら数千円程度）</li> <li>子どもが直接基盤を触りながら入力や出力を考えられる</li> </ul>	
RaspberryPi	<ul style="list-style-type: none"> <li>PCに近い性能を持っているが、PCに比べて安価</li> </ul>	<ul style="list-style-type: none"> <li>一般的なPCよりも性能は劣る</li> <li>モニター、キーボードが別途必要</li> </ul>
IchigoJam	<ul style="list-style-type: none"> <li>非常に安価</li> <li>起動が早い・機能が最小限なので子どもが利用してもリスクが低い</li> </ul>	<ul style="list-style-type: none"> <li>メモリは少ない</li> <li>Basicによる操作のみ</li> <li>モニター、キーボードが別途必要。ただしモニターは白黒でもよいため比較的安く揃えられる</li> </ul>
ロボット、センサーカー等	<ul style="list-style-type: none"> <li>画面外の物理的な動作を体験できる</li> </ul>	<ul style="list-style-type: none"> <li>プログラミング的要素は少ない</li> <li>組立に時間がかかる</li> </ul>

## 8. 民間企業が求めるプログラミング人材像

企業が求めるプログラミング人材像について、ベンダおよび ICT 利活用企業の幅広い分野の企業を対象にアンケート調査を実施し、得られた回答に基づきまとめる。

### 8.1. 調査対象の概要

アンケートは 116 社に依頼し、有効回答数は 23 社（有効回答率 20%）であった。

有効回答企業の事業分野は、複数回答による延べ企業数で集計すると、以下の通り、ソフトウェアベンダー、機器ベンダ、ウェブビジネス企業、ICT 利活用企業に分けられ、有効回答は、ICT 利活用企業、ソフトウェアベンダーの割合がやや多かった。

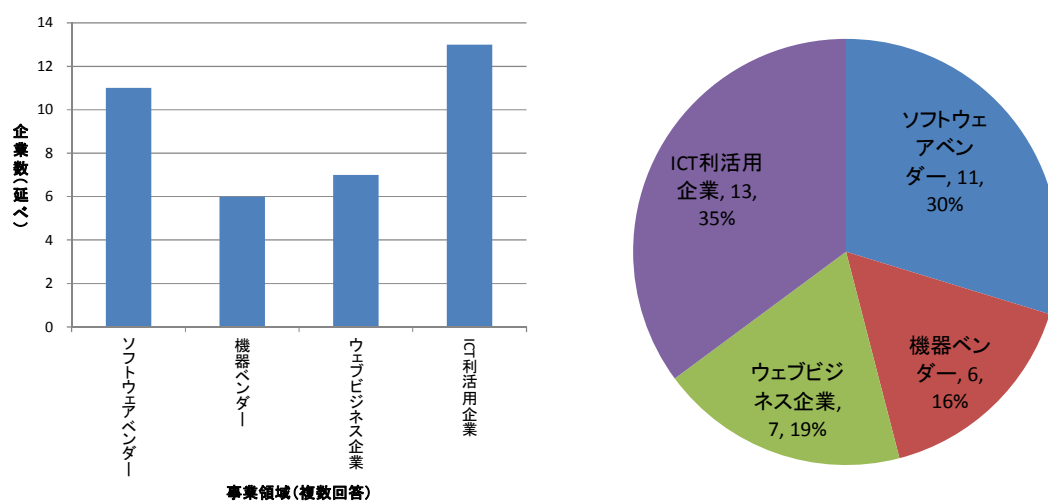


図 8-1 調査対象企業の事業区分 (複数回答)

調査対象の企業規模を従業員数でみた場合、下記の通り分布の幅が大きい。

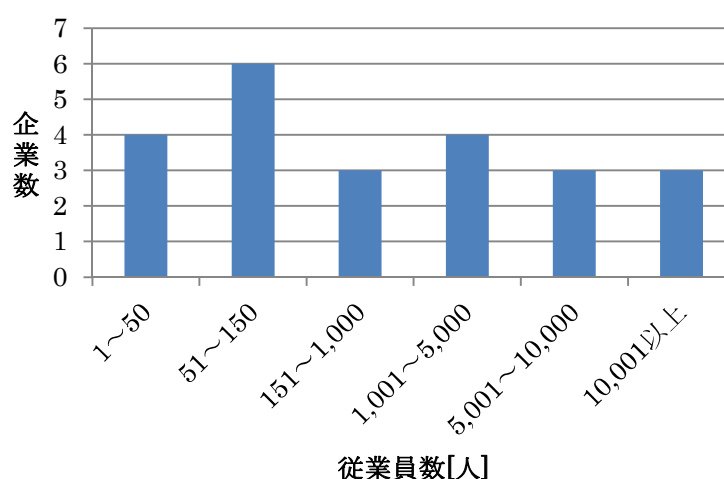


図 8-2 調査対象企業の従業員数



## 8.2. 期待する能力

アンケート結果から、民間企業が期待する人材の能力を集計した結果は以下の通りである。

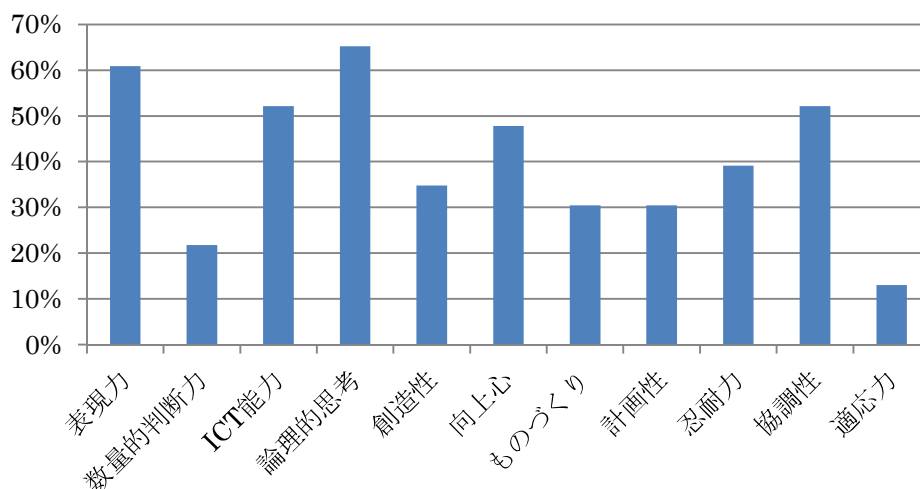


図 8-3 企業が期待する能力(N=23)

表 8-1 期待する能力に関する選択率(N=23)

能力等		選択率
基礎力	表現力・コミュニケーション力	61%
	数量的な判断力	22%
	ICT・情報を扱う能力	52%
思考力	論理的・批判的思考力	65%
	創造性・独創性	35%
	向上心・学習意欲	48%
	ものづくりに対する好奇心	30%
実践力	計画性・管理能力	30%
	忍耐力・根気	39%
	協調性・連帯感	52%
	適応力・社会参画欲	12%

有効回答数が少ないため、統計的な優位性を示すことは困難であるが、集計結果に基づき、特に値の高い項目に注目し傾向性についてまとめる。

比較的期待の高い能力として、「表現力・コミュニケーション力」、「論理的・批判的思考力」、「協調性・連帯感」、「ICT能力」が挙げられる。表現力・コミュニケーション力や協調性・連帯感は、組織として活動する際に期待される能力として重要である理解できる。

一方で、回答数の少ない能力としては、適応力・社会参画欲、数量的な判断力、ものづくりに対する好奇心、計画性・管理能力が挙げられる。

6章の教育事業者アンケートにおいては、教育がもたらす能力として、同様の選択肢による質問を行っているが、教育事業者の観点から以下の通り選択肢の表現を修正し調査を行っている。

表 8-2 企業ニーズと教育事業者のアンケートにおける選択肢の対応関係

区分	企業ニーズアンケートの選択肢	教育事業者アンケートの選択肢
基礎力	表現力・コミュニケーション力	言葉を正確に理解し、使う力
	数量的な判断力	距離、量、回数、見込などの数量的な感覚
	ICT・情報を扱う能力	ICTに関する基本的な知識
思考力	論理的・批判的思考力	論理的に物事を考える力
	創造性・独創性	新しいものを生み出す創造的な力
	向上心・学習意欲	自ら学ぼうとする意欲
	ものづくりに対する好奇心	ものづくりの意欲
実践力	計画性・管理能力	ものごとを計画的に行う力
	忍耐力・根気	ものごとをやり遂げる粘り強さ
	協調性・連帯感	他人と協力する力
	適応力・社会参画欲	—

6章で示した通り、回答結果は、就学前～小学校低学年においては、「創造的な力」「学ぼうとする意欲」「モノづくりの意欲」などが上位に挙げられ、小学校高学年～高校生の時期には、「距離、量、回数、見込などの数量的な感覚」「ものごとを計画的に行う力」「論理的に物事を考える力」などが上位に上げられた。これらは企業ニーズの上位と一致するものが少ない。標本数が少ないため、統計的には明確な結論は出せないが、企業が求める「表現力・コミュニケーション力」、「協調性・連帯感」について差が大きい。

乖離する理由としては、回答者の想定する対象年齢にずれがあることが予想される。教育事業者アンケートにおいては、小学校から高校へ年齢が上がるにしたがい、「表現力・コミュニケーション力」、「協調性・連帯感」は上昇する傾向にはあるが、企業が期待するように上位には至っていない。この結果をもとに想定される取組みの方向性としては、小学校、中学校では、これまでのようにそれぞれの年齢に適した能力の育成を行いつつ、中学、高校へと社会人に近づくにつれて、育成する能力の重点を企業ニーズに挙げられる表現力・コミュニケーション力、協調性・連帯感に移していくことが考えられる。

自由回答の中で期待する能力として特徴的な意見として以下のようなものがあった。

- 創作物が社会に与える影響を想像する力（ICT利活用企業）
- 失敗しても諦めずやり抜く力（機器ベンダー）
- 要求の分析力（ソフトウェアベンダー、ICT利活用企業）

これらについては、プログラミングとの関連性が低いものもあるが、「やり抜く力」、「要求の分析力」などについては、プログラミングに係る教育の中で育成することも可能と考えられるため、このような観点も考慮した教育が期待される。

### 8.3. 理想とする人材像

プログラミングに関する教育を通じて育成される人材の理想像については、自由回答の結果をもとに共通する観点から整理すると主に以下のようにまとめられる。

#### 理想とする人材像の整理

- 自主性・行動力の高い人材（3社）  
自ら考え、創作したり、行動を起こすことを楽しく思う人材。新たな創作物による社会への影響を考察できる人材など。
- 独創性が高く、イノベーティブな人材（3社）  
独創性が高く、イノベーションを起こせる人材、また製品などの企画力の高い人材
- 論理的思考力の高い人材（6社）  
論理力や基礎力が高く、仮説立てて検証を行うなどの能力の高い人材
- コミュニケーション力・協調力の高い人材（4社）  
コミュニケーションを通じた要求分析や、協調によるプロジェクト遂行の出来る人材。多様な価値観を認め合い、良好な人間関係を構築できる人材。顧客と真剣に向き合える人材。

以上の人材像は、期待する能力の調査結果と比較した場合、論理的・批判的思考力、コミュニケーション力、協調性・連帯感等の能力を期待するという点と関連性が高いが、理想とする人材像の自由回答においては、表現力に関する期待は明確に挙げている企業は見られなかった。また、創造性・独創性に関しては、期待する能力に関する選択式の設問と、理想とする人材像の自由記述の間で、前者は低く、後者は高かった点に違いが見られた。

なお、これらの結果については調査件数が少ないため、統計的に有意な結論を導くことは難しい。

### 8.4. 人材の不足感

人材の不足感については、他の設問に比べて有効回答数が少なく、全回答者 23 件のうち本設問に回答があったのは 13 件のみであった。データのバラつきに対して、回答数が非常に少ないため、人材の不足数に関して、統計的に有意な推計をすることは困難である。

IPA による IT 人材白書では、約 7000 社依頼、回答率 15%程度であるが、人数ベースの不足数の設問はなく、人材の不足数は推計していない。IT 人材白書では、不足感のある企業の比率を集計している。

以上のことから、本調査では、参考値として、有効回答数のうち、不足数等を挙げた企業の比率(1人以上不足と答えた企業の比率)と、その不足数の総数を集計することとする。

人材の不足感については、技術系、利活用系に関して、下表に示す人材区分について調査した結果、人材区分ごとの不足数（今現在、貴社における「新卒採用」以外の採用や外部の人材活用を考えている／考えてもよいと思われる人材数）、環境の変化時の採用数（向こう1年間で、景気の上向き感が強まった場合、自社の経営環境が整った場合、欲しい人

材が簡単に採用できるようになった場合などに、追加で採用・活用したい／してもよいと思われる人材数)が1名以上の比率を求めたものが下記である。

表 8-3 人材不足があると回答した企業の比率(N=13)

人材※		人材不足とする企業の比率	環境の変化時の採用を検討する企業の比率
主に技術系	プロジェクトマネージャ	56%	57%
	システム・コンサルタント	63%	50%
	設計・アーキテクト	71%	60%
	プログラミング技術者	75%	67%
主に利活用系	システム管理・運用者	20%	0%
	企画・要求分析者	43%	33%
その他		50%	33%

不足感のある企業の比率で見た場合、技術系の人材について高く、特に、プログラミング技術者(75%)、設計・アーキテクト(71%)が高かった<sup>48</sup>。

人材の不足数に関する人数ベースの集計結果においては、人材区別の不足数、環境の変化時の採用数は、特定の企業における人数の全体に占める割合が高く、回答数に対して、バラつきが大きく、人材区分ごとの不足数について、統計的に有意な推計を行うことは困難である。

したがって、ここでは、あくまでも参考値として、技術系、利活用系の2区分について在籍者数に対する不足数の比率をもとめ、IT人材白書に示される国内の関連人材総数にかけることで、不足数を求めるものとする。

技術系の不足数の比率は1.4%、利活用形の不足数は3.2%であった。IT人材白書においては、IT企業(IT提供側)のIT人材数は、819,000人、ユーザ企業(IT利活用側)のIT人材数は276,000人と推計されている。したがって、技術系人材の不足数は、11,000人、利活用系人材の不足数は、88,000人と推計される。ただし、本推計は、限られたサンプル数からの求めた参考値であることに注意が必要である。

<sup>48</sup> 独立行政法人情報処理推進機構の調査によるIT人材白書2014においては、IT企業がIT人材を拡大する上で重視する職種として、上位には、プロジェクトマネージャ(70.5%)、アプリ系技術者(55.4%)、インフラ系技術者(26.3%)が挙げられている。

## 9. プログラミングに関する教育の整理

プログラミングに関する教育の目指す方向性や教育の場、教育の導入段階について、5章～7章の調査結果をもとに整理した。

### 9.1. プログラミングに関する教育の目指す方向性

プログラミングに関する教育の方向性としては、プログラマー育成以外にも幅広い人材育成に関わる方向性があると考えられる。プログラミングに関する教育を実施する事業者では人材育成の目標として、「5.3 教育の目標・対象」に示す通り、「課題解決力・創造力」、「自己表現力」、「コンピュータ・ICT の理解」、「地域に貢献できる人材」が挙げられていた。また、教育の効果についても「6 プログラミングに関する教育がもたらす効果」の通り、コンピュータに関する知識だけでなく、21 世紀型スキル全般の習得についても効果が大きいことが示されている。

総合すると、プログラミングに関する教育の方向性としては主に

- ① 21 世紀型スキルの習得  
批判力、問題解決力、コミュニケーション力、コラボレーション力、ICT リテラシー等、21 世紀の知識基盤社会で求められる能力の習得。
- ② ICT を生み出す人材の育成  
ICT インフラの構築やソフトウェア開発等によって ICT を支え、新しい ICT を生み出す人材の育成。従来のプログラマーではなく、製品・サービスへの実装までを意識した力も獲得する必要がある。
- ③ ICT を幅広い産業へ使う人材の育成  
ICT を使い、イノベーションを起こす人材の育成。コンピュータサイエンスの知識と情報モラルに裏打ちされた、ICT 活用の実践力も獲得する必要がある。

と整理することができる。(図 9-1)。①21 世紀型スキルの習得が ICT を生み出す人材、使う人材の基礎能力である等、これらの方向性は互いに影響し合うため不可分のものではあるが、教育を整理する上ではどの側面に軸足を置いた教育かを意識することが重要である。

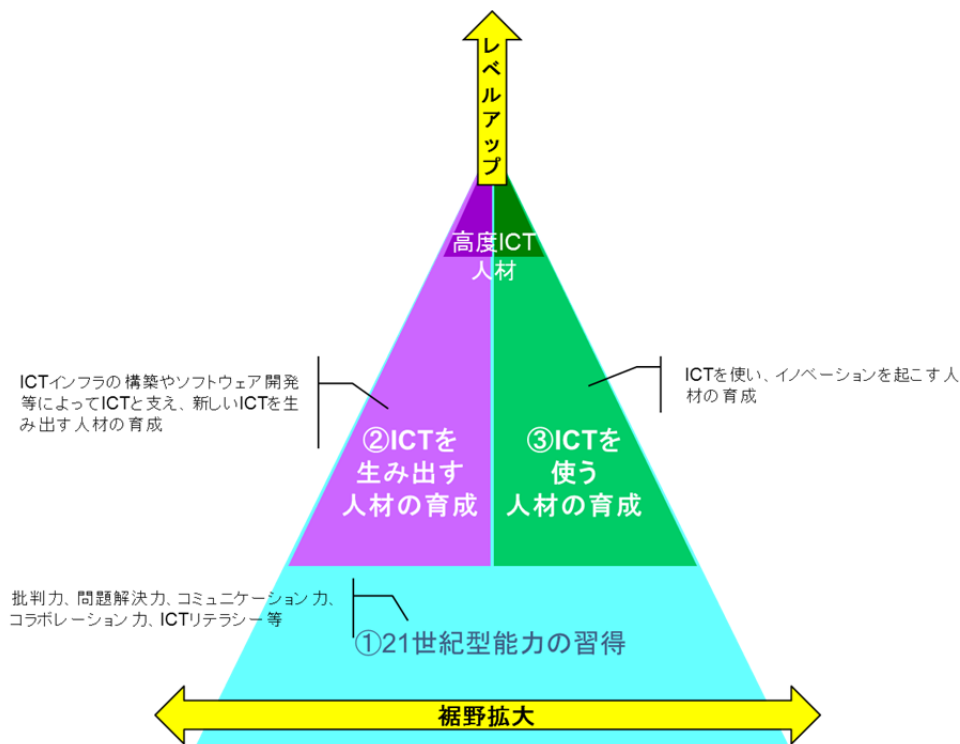


図 9-1 プログラミングに関する教育の方向性

プログラミング人材育成を普及展開することで、21世紀型スキルを獲得する子どもが増加（裾野拡大）し、全体のレベルアップに繋がる。このように、ICTを生み出す人材だけでなく、使い手側のレベルアップや、そもそも基本的な能力の底上げといった効果があるにも関わらず、現在はプログラミングに関する教育の定義や目的の整理が進んでいないため、世間一般には、②ICTを生み出す人材育成にだけ必要とされる教育、といった印象を持たれている。今後は①や③のようにこれからの時代に必要とされる人材の育成といった意義を持つものであることについて、関係者及び世論に対して啓発を図る必要があると言える。普及啓発については「10.1 普及啓発に関する課題とその解決策」にて検討する。

## 9.2. プログラミングに関する教育の場

プログラミングに関する教育の場としては、公教育と公教育以外の場での教育が考えられる。現在、プログラミング人材育成の場として主流になっているのは、民間事業者やボランティア、NPO法人等、学校以外が中心となった小規模な取組みである。

学校授業でプログラミングが用いられている事例もあるが、まだまだ公教育全体で見ると数は多くない。今後、①21世紀型スキルの習得、③ICTを作る人材の育成といった側面が評価されることで、プログラミング人材育成が公教育でも実施される機会が増加すると考えられる。ヒアリングでは民間事業者等からプログラミングを公教育で扱うことへ期待する意見は挙げられた。一方、指導方法によっては効果的な教育とならない可能性があるとする意見も挙げられた。

### 公教育への期待

- 公教育でプログラミングを扱って欲しい（複数事業者）

- 公教育で扱い、プログラミング教育の裾野を広げてほしい。

#### 指導方法に関する意見

- 学校の教育は、先生の技量に応じて内容やレベルにムラが大きい。
- プログラミングを学習する際は生徒の自主性が大事だと考えている。学校教育では進捗管理をするために、勝手に進めることを嫌うため、技術は学べるが、面白みがなくなる可能性がある。
- 市内の小学校のいくつかのクラスでは Scratch を扱っていた。ただし生徒の反応は芳しくなかった。学校の授業では Scratch の面白さを伝えにくいのもかもしれない。
- 成果発表の場が筆記試験になることが問題である。
- プログラミングは義務教育の教科としてではなく、数学や理科などの教科の要所でツールとして使ってほしい。
- 公教育では一つの言語に絞らず、複数の言語を選択肢としてほしい。
- プログラミング教育は時間がかかることが問題であり、適性がわかれやすい教育内容でもあり、公教育でやる場合の課題となるのではないか。
- 公教育では短時間しか取れないため、教師主導の課題（問題）設定や教師の意向を踏んだ課題を取り上げることになり、子供自身が課題を設定する時間を取れないのではないか。

### 9.3. プログラミングに関する教育の導入段階

青少年がプログラミングに出会い、プログラミングを継続するステップとして、以下のとおり整理することができる（図 9-2）。

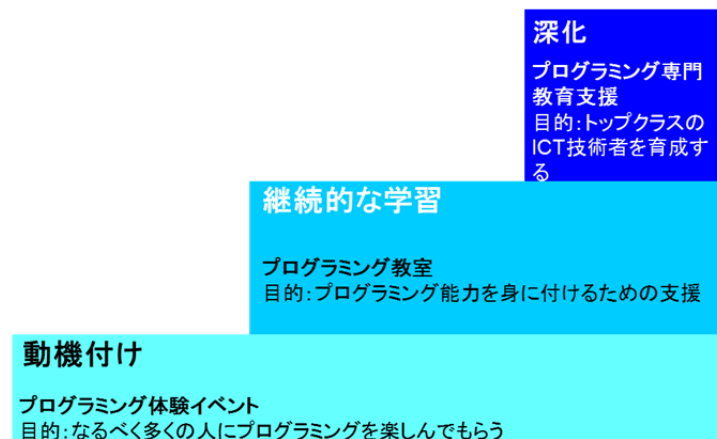


図 9-2 プログラミングに関する教育の導入段階

プログラミングに関する教育自体は比較的新しい教育方法であるため、まずは教育を受ける(子どもへ受けさせる)動機付けとしてのプログラミング体験イベントがある。そこで興味をもった子どもは継続的にプログラミング教室へ通うことになる。さらにその中でも一部の子どもはプログラミングに深くはまり、自主的に高度な内容を習得するため、そのよ

うな深掘りした（深化した）学習に取り組む子どもを支援するプログラミング専門教育支援がある。

### 9.3.1. 動機付け

動機づけのための「プログラミング体験イベント」ではできるだけ広く、多くの子どもに体験・楽しんでもらうことを目的とする。そのために、できるかぎり参加しやすいように大きなイベントのなかの一つのテーマとして開催したり、フリーマーケット等の別のイベントと共催する形で集客するケースがあった。また学校への出張授業等もここへ位置づけることができる。

具体的なイベントのイメージとしては以下のようなものがある。

#### (1) 学校への出張授業

表 9-1 学校への出張授業の概要

項目	概要
指導者	プログラミング教室を展開する事業者や教育スキルをもつ ICT 技術者等と学校の先生が連携するケースが多い。
会場	学校
教材、機器類	教育事業者側の持ち込む機器（タブレット、RaspberryPi 等の小型 PC）や学校にある機器
言語	Viscuit、Scratch 等のビジュアル言語が多い。
指導期間	45 分～90 分程度

プログラミング教室を展開する事業者や教育スキルをもつ講師が、タブレット端末や教材を学校へもちこみ、数時間程度プログラミングを体験する機会を提供する。または学校にある PC ルームやタブレット端末を利用して指導を行う。利用されるプログラミング言語としては、Viscuit や Scratch 等ビジュアル言語が多い。小学校低学年への授業の場合はとくに PC 操作が困難であるため、タブレット端末を用いることが多い。初めてタブレット等に触れる生徒もいるため、30 人～40 人のクラスに対して、指導者とメンター数人体制が生徒の様子を見ながらフォローして進めることが多い。

#### (2) ワークショップ

表 9-2 ワークショップの概要

項目	概要
指導者	子供への教育経験をもった人材と技術面でのサポート役が連携するケースが多い。子どもと上手くコミュニケーションをとれることが重視される。
会場	イベント会場、大学等
教材、機器類	タブレット端末、PC、RaspberryPi+ロボットキット等を組み合わせるケースもある。
言語	Visucuit、Scratch 等のビジュアル言語が多い。



指導期間	数分～半日程度。長くて1日。
------	----------------

子どもへの教育スキルをもつNPO 法人と ICT 技術者が連携し、プログラミングを楽しむイベントを短い場合は数分、長くても半日程度で開催する。

大型イベント（プログラミングに限らない幅広い人が集まる内容のイベント）の中の一つの講座としてプログラミングワークショップが設置されることもある。その場合は数千人以上の子どもや大人にプログラミングを体験してもらったケースやフリーマーケットを併催し、ICTに関心の無い保護者にも足を運んでもらう工夫した事例もあった。

プログラミング単体でのワークショップの場合は、少し時間をかけて、多くて数十人程度へプログラミングを体験してもらう。ロボット等の物理的な工作をとまなう講座や、保護者にも子どもと一緒に体験してもらう事例もあった。開催規模に関わらず、用いる言語は Viscuit や Scratch 等ビジュアル言語が多い。

特徴的な取組みとしては以下のようなものがある。

#### ● 保護者参加型体験会の開催

保護者の抵抗感を弱めるためにも、子どもだけでなく家族で参加するイベントとして開催。県の家族参加イベントの一環として広報することができるため、集客力も高い。保護者、子どもそれぞれに端末を割り当て、体験することで、保護者もプログラミングの楽しさを体感することができるため、好評となっている。またそのイベントから継続して教室へ通う生徒も出てきている。

### 9.3.2. 継続的な学習

動機づけの段階で、プログラミングに関心をもった子どもたちを対象に、継続して学習する機会を提供する「プログラミング教室」では、プログラミングスキルやプログラミングによって①21 世紀型スキルを習得することを目的とする。動機づけのためのイベントよりも高度な内容になるため、ICTの専門的な知識を持つ講師が必要とされるケースが多い。具体的な教室のイメージとしては以下のようなものがある。

#### (1) スクールへの通学

表 9-3 スクールの概要

項目	概要
指導者	ICT 技術者が講師のケースが多い。
会場	企業の会議室、公民館
教材、機器類	タブレット端末、PC、ロボット等
言語	Scratch 等のビジュアルプログラミング言語、HTML5 や JavaScript 等の一般的な言語のケースも多い。
指導期間	1～2 時間×月に数回。

月に数回、教室へ通学して数時間プログラミングを学習する。講師やアシスタント（メンター）1 名に対して生徒数人の割合で指導する。基本的には生徒の自主性を尊重し、生徒の理解度に併せてカリキュラムを進める場合が多い。そのため、講義形式で進める事例は

少ない。プログラミング言語や教材については、人材育成の方向性によって異なる。

②ICT を生み出す人材の育成を重視する事業者の場合は、一般的なプログラミング言語（JavaScript、C 言語、Basic 等）を大人と同様の教材を用いて学習する場合がある。そのため ICT 技術者の経験を持つような、専門的な知識をもった講師が指導するケースがほとんどである。また、①21 世紀型スキルの習得や③ICT を使う人材の育成を目指す場合は、生徒間のコミュニケーションや作品の発表機会を重視したカリキュラムとなっている場合が多い。初めは Scratch 等の教育用言語を用い、その後一般的なプログラミング言語や進むケースや、レゴマインドストーム EV3 等のロボット操作やスイッチ端末の自作等、物理的な動作を伴う教材を利用するケースがある。

特徴的な取組みとしては以下のようなものがある。

#### ● オンライン講座の提供

プログラミングを継続して学習したくとも、通える範囲に教室がないケースがある。オンラインで授業を提供している事業者には、学習希望者からの問い合わせが増えている。ただしオンラインで指導する場合、インフラとして利用する無料 TV 電話サービスに頼ることになる。

通信教育コースよりも TV 電話をとおしたオンライン講座のほうが、継続率・成長スピードともに良い。

#### ● ロボット教室

レゴマインドストーム EV3 を使った教室が全国に展開している。世界大会があるため、子どもたちが明確な目標を持つことができ、子どもたちが長期にわたって継続して通っている。

### (2) アフタースクール・児童館でのクラブ活動

表 9-4 アフタースクール、児童館でのクラブ活動の概要

項目	概要
指導者	IT スキルをもたない主婦等が指導し ICT 技術者がサポート役として支援しているケースがある。
会場	児童館、学校の PC ルーム等
教材、機器類	タブレットよりも RaspberryPi を含む PC が多い。
言語	Viscuit、Scratch 等ある程度子供だけで進めることができるもの。
指導期間	放課後に週 1 回程度の開催。事例によっては不定期開催。

週に数回～月に数回、学校の授業終了後に学校や児童館で保護者や NPO 法人が子どもを預かる。指導者は積極的に指導するのではなく生徒の自主性を尊重し、子どもの相談役となる場合が多い。生徒の自主性に任せつつも、他の子どもたちへの発表の機会等を提供することがモチベーションを維持するために重要である。ICT の知識を持った講師がいることが望ましいが、知識の有無に関わらず地域の保護者やボランティアが中心となって、子どもとうまくコミュニケーションをとりながら活動を継続するケースが多い。全国的にアフタースクールでプログラミングを取り入れているケースは少ないが、会場や機器類の課題

が解決され、地域貢献にも繋がる取組みとして参考になると考えられる。

特徴的な取組みとしては以下のようなものがある。

- **小学校の PC 教室を使ったアフタースクール教室**

地域のボランティアスタッフが指導者になり、放課後の PC 教室をつかって Viscuit 教室を開催している。指導者はもともとプログラミングスキルをもっていないが、要所で IT スキルを持つ人の支援を受けながら継続させている。

普段は指導者 1 人で数人～数十人の子どもを受け持つため、生徒一人一人へ指導するのではなく子どもが自ら自然と取り組みたくなる環境を作っている。例えば Viscuit 解説書を子どもたちが作り、新しく参加する生徒はその解説書を参考に、生徒同士で教え合いながら進めている。

- **児童館のプログラミングクラブ**

地域の児童館を運営する NPO 法人が新しい取組みとして RaspberryPi と Scratch を使ってプログラミングクラブを開催している例がある。初めはワークショップ形式で開催した後、とくに興味をもった子どもたちが中心となって継続した活動になっている。NPO 法人は子どもや保護者の ICT リテラシーの向上を重要視しているが、児童館の運営メンバーには ICT の専門知識を持つ人材はほとんどいない。そのため、プログラミングクラブ始まって数か月で子供は大人以上のレベルに到達している。大人から技術的な指導はできないが、子どものモチベーションを維持するために子どもたちの作った作品を鑑賞してアドバイスをしたり、児童館の子供たち全体に向けた発表会の機会を提供している。

また、学校での集団授業に対して、児童館での取組みは子供たちの自主性や探究心をもとに進めることを重要視するため、ICT を利用する上での課題や問題点に子どもたち自身が気づくといった効果もあるといった意見もあった。

### 9.3.3. 深化

継続的にプログラミングを学習する中で、自ら楽しみ方を見つけ、教室やカリキュラムのレベルを超えて学びだし、大きく伸びる子どもを支援するための取組みがある。対象となる子どもの絶対数が少ないこともあり、「深化」段階にあたる取組みは少ないが、一部の教育事業者では奨学金制度を設定しているところもある。公的な取組みとしては、情報オリンピック<sup>49</sup>や U-22 プログラミングコンテスト<sup>50</sup>等がある。

---

<sup>49</sup> 情報オリンピック日本委員会 <http://www.ioi-jp.org/>

情報オリンピック日本委員会は日本情報オリンピックを主催し、国際情報オリンピックに日本代表選手を派遣する事業を軸に、数理情報科学教育に貢献するための諸事業を実施する。

<sup>50</sup> U-22 プログラミングコンテスト <http://www.u22procon.com/>

経済産業省、総務省、文部科学省、IPA、情報処理学会などが後援するコンテスト。優れた才能を持ったイノベティブな IT 人材の発掘と育成、単にプログラムのできる人材ではなく、アイデアに富んだソフトウェア開発に取り組む人材の発掘を目的として開催される。

## 10. プログラミングに関する教育の普及展開に有効な取組手法

プログラミングに関する教育は現状としては民間事業者等が中心となって取り組んでいる。とくに継続的な学習の場は今後も民間主導になる可能性が高く、また公教育については文部科学省において検討されているため、本章では公教育以外でのプログラミングに関する教育を普及展開方策について検討した。10.1 では普及啓発に関する課題とその解決策を、10.2 では運営方法に関する課題とその解決策を整理した。10.3 では 10.1、10.2 で整理した結果に基づき、今後取組むことが期待される方向性についてまとめた。

### 10.1. 普及啓発に関する課題とその解決策

プログラミングに関する教育の普及啓発における課題とその解決策を検討するため、教育事業・サービスに関して、「認知」、「選定」、「参加後」のフェーズに分けて分析する（図 10-1）。フェーズごとに課題を分析することで、ボトルネックを特定し、効果的な普及啓発方法を検討する。

最初のステップとして、プログラミング教室・講座の告知方法をアンケートとヒアリング調査をもとにフェーズごとに整理した。また、教室・講座への参加には保護者の同意が必要となるため、広告に接した際に想定される保護者と子どもの行動・思考を記載している。

	認知	選定	参加後
告知方法	<ul style="list-style-type: none"> <li>WEBサイト、SNS</li> <li>知人の紹介</li> <li>学校の配布物</li> <li>市の広報物</li> <li>貼り紙</li> <li>TV報道</li> </ul>	<ul style="list-style-type: none"> <li>WEBサイト</li> <li>クチコミ</li> <li>関連情報の検索結果</li> </ul>	<ul style="list-style-type: none"> <li>教室のチラシ/説明</li> <li>今後のイベントのチラシ/説明</li> <li>イベントの結果報告</li> </ul>
子どもの行動と思考	<ul style="list-style-type: none"> <li>広告を見る</li> </ul>	<ul style="list-style-type: none"> <li>詳細情報を探し、検討する</li> <li>保護者へ相談する</li> </ul>	<ul style="list-style-type: none"> <li>親に感想を伝える</li> <li>継続したいか考える</li> </ul>
保護者の行動と思考	<ul style="list-style-type: none"> <li>広告を見る</li> <li>➢ デザイン</li> <li>➢ 情報の信頼性(広告媒体は？主催、共催は？)</li> <li>※TV報道や市の広報物、行政の後援があるものは信頼されやすい</li> </ul>	<ul style="list-style-type: none"> <li>詳細情報を探し、検討する</li> <li>➢ 参加可能性(いつ、どこで、何を？いくら？)</li> <li>➢ 参加必要性(効果は？子どもに必要なか？子どもは行きがっているか？)</li> </ul>	<ul style="list-style-type: none"> <li>子どもに尋ねる</li> <li>広告を見る</li> <li>➢ 評価(子どもの為になったか？今後継続すべきか？他の人にも勧めるべきか？)</li> </ul>

図 10-1 プログラミング教室・講座に対する想定される保護者の思考・行動

認知フェーズでは、WEB サイトや SNS 等インターネット経由での情報発信、またはチラシの配布、ポスター掲示等が行われている。広告のデザインや情報の信頼性によって、保護者や子供はこれらの情報を目に留め、内容を読むかを決める。認知フェーズでプログラミング教室等が認知された後は、選定フェーズとなる。選定フェーズでは、広告で得られた情報やさらに保護者や子どもが得た詳細情報をもとに、イベントや教室へ参加するかどうか判断される。そのときの判断基準としては、参加可能性（いつ、どこで、いくらで、

何を開催するのか?)や参加必要性(子ども・自分が参加したいか?、子ども・自分にとって必要か?等)があると考えられる。

選定フェーズで参加を決めた場合、保護者や子どもが参加申し込みを行い、イベントや教室に参加する。参加後フェーズでは、子どもと保護者で感想の共有や、イベント運営者からのフィードバック・告知等をもとに、今後継続してプログラミングを学習するかどうかを検討する。

### 10.1.1. 現状と課題

各フェーズにおける課題をアンケートとヒアリングから抽出した。

#### 10.1.1.1. 認知フェーズにおける現状と課題

##### (1) 効果的な告知方法

WEBサイトで告知した場合、既に関心をもった人たちへ広くアプローチすることができるが、SNS告知した場合は人の繋がりが基本となるため、知人ベースの狭い範囲へのアプローチとなる。インターネットを経由した告知ではあっても、広いターゲットへアピールするにはWEBサイト、対象を絞って依頼するにはSNS、と使い分ける必要があると言える。

参加者数が多くなる広告手法としては、広告の信頼性が明確になっているときだと言える。特に公共性が高い、テレビや市の広報物で取り上げられると参加者が多くなるという意見が複数挙げられた。幅広い人々へアプローチできるうえに、媒体としての信頼性も高いため、効果があると考えられる。

また、学校を経由して告知を行うことができると、教室・講座のターゲットの年齢層に直接アプローチすることができる。ただし、学校での告知には、国や教育委員会からの後援名義がなければ難しいが、後援名義の取得方法を知らない事業者もいるといった意見もあった。

アンケートとヒアリングから得られた意見は以下の通り。

#### インターネット経由

- WEBサイトで告知した際、市外(東京)からの参加が多かった。関心を持っている人が探しているようだ。ただし地域のイベントとして企画しているため、今後どう対応するか考えたい。
- SNS経由か知り合いや参加者経由での広報を実施している。今後どうやって広報するかが課題

#### 公共性の高い媒体

- テレビで取り上げられたあとは満員近くに申込みがあった。チラシ配布だけではそこまで大きくない。
- 募集は市の広報誌に掲載すると参加者が多い。各学校へチラシを置くこともあるが、なかなか集まらない。

#### 学校経由

- 学校に広報してもらっただけでも効果が出るのではないかと。総務省後援と書いてあると、学校で配りやすい。受け入れやすい肩書きがほしい。
- 生徒の募集は、学校での張り紙や、学校の先生に電話を掛け、プログラミングに関心の高い先生にチラシを配布してもらっている。
- 教育委員会等の後援の取り方を別の団体へ教えることがある。

## (2) 広告デザイン

プログラミング教室・講座を運営する事業者が広告作成のノウハウを持っていない場合は、効果的なデザインを作成することができない。例えば子どもに対しては子供たちが楽しんでいる写真等による訴求力が高いと言われている。このようなノウハウや、フリー素材等を共有する場があることは効果的な告知に繋がると考えられる。

アンケートとヒアリングから得られた意見は以下の通り。

- チラシのデザインやメッセージについて、別の団体から相談されることがある。フリー素材を提供することもある。
- イベントの告知文章をみていると、何を伝えたいかが上手くない。プログラミングを前面に出さずに、「●●を作ろう」または「アイデアを形にできる!」、等をメッセージしたほうが伝わりやすいだろう。プログラミングを手段としたときの目的を見極めることが大事だろう。
- 子どもは同年代が楽しんでいる姿をみると興味をもつ。

### 10.1.1.2. 選定フェーズにおける課題

#### (1) 参加可能性

関心があっても、金銭的、地理的な理由によって通えない人がいる。一方で、別のイベントと共催し、金銭的・物理的ハードルを下げることで、非常に多くの参加者を集めることができる。

アンケートとヒアリングから得られた意見は以下の通り。

##### 金銭的・地理的な制約

- (関東) 習い事の金額からするとかなり月謝は高い。そこにハードルがある。いいなと思っても受講料が高い。現状は経済的に余裕のある家庭の子供の受講が多数である。
- 子どもは興味があっても保護者が興味を持たなければ有料の講座への参加が難しい。きっかけづくりの初回講座を無料で提供する必要がある。
- (東日本) 地方で開催したイベントに参加して興味を持っても、地方には教室がないため通えない子もいる。
- (西日本) 現在は知名度の高いイベントの一部として募集するため、参加者は多い。し

かしプログラミング講座単体で集客しようと考えたと、プログラミングへの理解や関心が低いので工夫が必要になる。

- (西日本) フリーマーケットとイベントを共催した。IT やプログラミングに関心の有無に関わらず家族が集まるため、ついでにイベントに立ち寄る子どもが多くなった。
- (西日本) 地方は東京よりもお稽古へかける相場の金額が低い。
- 一般的にオンライン教室の問い合わせが多い。今後プログラミング教育の通信教育も増えるのではないかと。Scratch の基本操作を効率よく指導する方法を検討している企業がある。基本をオンラインで身に付けさせた後に対面で発展させることが現実的かもしれない。

#### 無料体験会の開催

- 無料体験会に参加してから入塾してもらっている。そのため長続きしている。
- 無料体験会を開いて、実際にみてもらって、やりたいという人だけに入ってもらおうようにした。

## (2) ターゲット

アンケート結果からも受講生は男子児童・男子生徒が 9 割近い。女子児童・女子生徒やその保護者からはプログラミングは男子生徒向けというイメージによって、参加しづらくなっている可能性がある。潜在ニーズがあると考えられるため、女子児童・女子生徒が参加しやすいイベントの提供によって、参加者が増加することも考えられる。

アンケートとヒアリングから得られた意見は以下の通り。

- 間口を広めることも重要と考えている。女子向けイベントを実施したところ、募集 40 人枠のところから 120 人の応募が殺到した。

## (3) 効果に関する理解・必要性

IT リテラシーが高い保護者や教育熱心な保護者はプログラミングに関する教育への関心が高い。ただしこれは関東での傾向である。地方における ICT リテラシーは首都圏に比較した場合高くないため、「プログラミング」という単語すら認識されない地域がある。

また、ICT に対してネガティブなイメージをもっている場合や、IT 企業への偏見が強いため子どもを ICT に近づけたくないと考える保護者もいる。プログラミングに関する教育の効果が見えていないことが大きな要因といえる。

アンケートとヒアリングから得られた意見は以下の通り。

#### 保護者の ICT リテラシー、地域性

- (関東) 現在イベントやスクールへ参加する子供の親は、子供に様々な機会を与えたいと思っているか、IT リテラシーの高い親である。IT リテラシーが余り高くなくとも、子供のために広くアンテナを貼って、イベントを見つけて子供に勧める親もいる。

- (西日本) 来るきっかけは2パターンある。校門配布やチラシを直接子供に渡して子供が興味を持つパターンと親が調べてくるパターン。純粋に IT、プログラミングがこれから大事になると考えている敏感な保護者は1、2割程度。
- (西日本) IT 系の企業がない。プログラミングへのイメージの前に、そもそもプログラミングをまったく知らないため、プログラマーへのイメージがまったくないのが現状。
- 東京が極端に盛り上がっているのは事実である。IT に対する考え方も地域で違う。
- (西日本) 子供や親への訴求力が地方では低い。地域特性によると思うが、東京のようにプログラミング講座がすぐに埋まる状況ではない。
- (東日本) 地方では IT の浸透が遅れている。東北など、親はインターネットはあまり使っていない人も多い。
- 関東のイベントはキャンセル待ちがでるほどだが、地方のイベントへの参加人数はそれほど多くない。

#### プログラミングに対するネガティブイメージ

- 社会的にプログラミング教育を受け入れられる態勢が整っていないのではないかと。民間の塾等が増え、一般的なお稽古として認識される必要がある。
- (西日本) 市のメディア教育方針としてはコンピュータを触らせない方針になっている。小学生へのプログラミングに関する教育をやりやすい環境ではない。
- 保護者世代の偏見が強い。IT へのマイナスイメージと低賃金・長時間労働の ICT 技術者になってほしくないという懸念の2つがある。
- 保護者の方から「子どもをエンジニアにしたいわけではない。将来そんな激務な職にしたいわけではない。」とよく言われる。プログラミングにそのようなイメージを持つ人が多い。
- 誤ったイメージを持たれる可能性があるため、言語スキル習得は前面に出していない。
- 現在募集するときもあまりプログラミングであることを全面には出していない。

#### プログラミングに関する教育の効果

- 子どもにとって何がよいか、という統一的なメッセージが無い。ただし保護者の中での何となく今の教育のままではよくない、という危機感があり、それによって盛り上がっているのではないかと。
- プログラミングに関する教育の効果について、多くの人にとっての共通理解となるようなメッセージを作り上げていく必要がある。

### 10.1.1.3. 参加後フェーズにおける現状と課題

#### (1) 継続者

プログラミングに関するイベントを開催すると、そのなかでも約1割の子供がプログラミングに対して強い関心を持ち、継続して受講することを希望しているという意見があ



った。

保護者に理解してもらうには、体験会を通して子どもの姿を見てもらうことが一番効果的である。また、体験会を通して通塾を検討することができるため、体験会の存在は参加のハードルを下げる効果もある。

アンケートとヒアリングから得られた意見は以下の通り。

#### 継続率

- 講座等のイベントへ参加した子どものうち、1割ぐらいはタブレットに食いついて離れなくなる。
- 20～30人くらい教えると、約2割は子供だけで取り組むようなプログラミング好きが混じっている。一方で1割程度の子供は飽きて辞めてしまう。
- 体験会参加者の約10%が教室へ通う。想定より低い割合ではある。
- 受講者のリピート率は5割ぐらい。今後は8～9割を目指している。

#### 継続できない要因

- 継続して通える教室がないため、単発のイベントに連続して何度も通う子どもがいる。
- 体験会やイベントに出てもそこで終わってしまうことが多く、そこで終わらないように、定期的に学べるようにしたい。

#### 継続者確保のための取組み

- 機器の設置だけでは子供は継続して遊ばなかった。環境提供だけでなく、一緒に遊ぶ人がいることが重要だとわかった。
- 言葉で説明しても伝わらない。子どもが楽しくプログラミングで遊ぶ姿をみると、すぐに保護者にとっても有用性を理解してもらえる。
- 継続して学んでいる子供たちを核として取組みを告知することで、参加者が増やせそうである。
- メインは考える力の育成だが、そこを保護者にわかってもらうには時間がかかる。見えにくい力である。わかっただけのように、保護者に資料をインターネット上で見れるようにしている。
- 継続的に通ってもらうには、保護者の理解が不可欠であるため、体験教室では保護者にも体験してもらう。

## (2) 子どものモチベーション維持

モチベーションを維持するためにはわかりやすい目標の設定が必要である。目標の代表例としてはコンテストが挙げられるが、コンテストの開催数が限られている。プログラミングに関する教育を行う事業者は小規模なところが大部分であり、コンテストを事業者単体では開催が難しい。小学校低学年から高校生まで、広い年代が目標にすることができるコンテストがあると、プログラミング教室・講座を継続するモチベーションの維持に繋がると考えられる。

アンケートとヒアリングから得られた意見は以下の通り。

- プログラミングの成果を発表する場がないと継続しない。
- 目標となるプログラミングの全国大会がない。

### 10.1.2. 普及啓発における課題のまとめ

10.1.1 でプログラミングに関する教育の普及における課題を整理した。その結果、各フェーズのなかでも認知フェーズ・選定フェーズに対する課題が多く挙げられていた。選定フェーズについては、特にプログラミング人材育成の目的や効果、対象について偏ったイメージを持つ人が多いために、参加希望者が限られていると考えられる。各フェーズでの離脱（プログラミング教室・講座への参加、継続に繋がらなかった割合）のイメージは図 10-2 のようになる。ピンク色が次フェーズへ進んだ保護者・子ども、青色が離脱した保護者・子どもである。各フェーズでポジティブな行動に繋がった理由として挙げられたものを赤色の吹き出しに、ネガティブな行動となった理由として挙げられたものを青色の吹き出しに記載している。

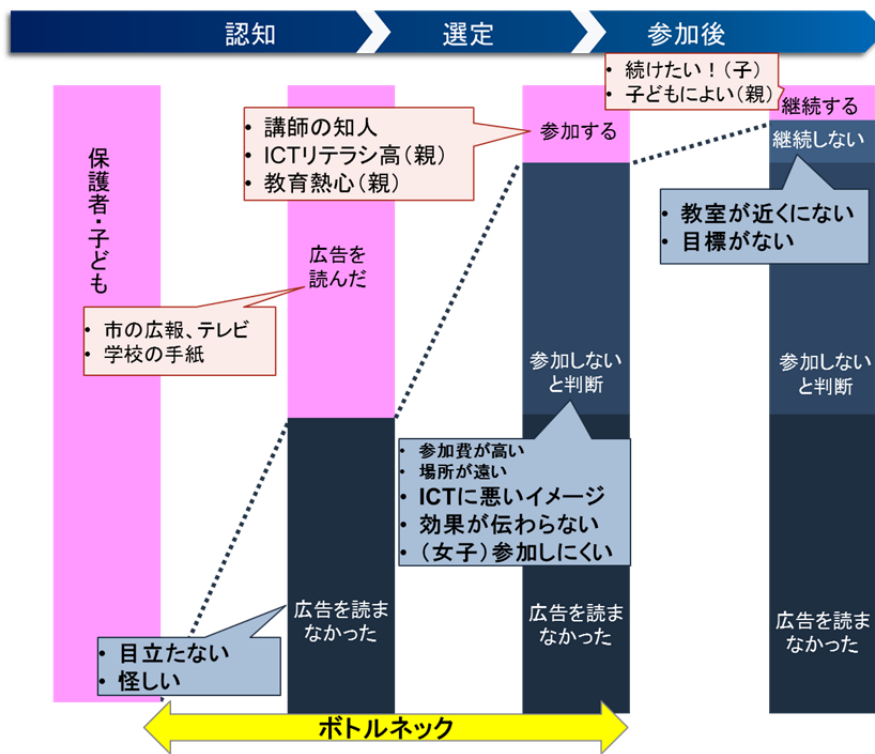


図 10-2 各フェーズにおける離脱率<sup>51</sup>のイメージ

全フェーズにおいて離脱率を減らすことが重要ではあるが、効率的な普及展開のためにはボトルネックになっていると考えられる認知フェーズ、選定フェーズについて重点的に課題を解決する必要があると考えられる。それらを検討するため各フェーズの主要な課題と、事業者が実際に行っている課題解決のために参考となる取組みを整理した。

<sup>51</sup> 各フェーズで離脱した割合

表 10-1 各フェーズの課題と課題解決に参考となる取組み

カテゴリ	課題	参考となる取組み
認知 (10.1.1.1)	<ul style="list-style-type: none"> <li>・ 広告の信頼性が不足している(1)</li> <li>・ デザインに問題がある(2)</li> </ul>	<ul style="list-style-type: none"> <li>・ 公共性の高い媒体や学校経由での告知(1)</li> </ul>
選定 (10.1.1.2)	<ul style="list-style-type: none"> <li>・ 物理的（開催地が遠い）、金銭的（受講料が高い）な制約がある(1)</li> <li>・ 地方では盛り上がっていない(3)</li> <li>・ ICT リテラシーが高くない保護者へプログラミング人材育成の効果、メリットが伝わりにくい(3)</li> <li>・ IT企業やプログラミングに対する偏見がある(3)</li> <li>・ 女子が参加しにくいイメージを持たれている(2)</li> </ul>	<ul style="list-style-type: none"> <li>・ 集客力のあるイベントと共催(1)</li> <li>・ 女子の参加しやすいイベントの開催(2)</li> <li>・ 無料体験会の開催(1)</li> </ul>
参加後 (10.1.1.3)	<ul style="list-style-type: none"> <li>・ 明確な目標、成果発表の場がない(2)</li> </ul>	<ul style="list-style-type: none"> <li>・ 保護者向け説明資料の提供(1)</li> <li>・ 子どもが楽しんでいる姿を保護者へ見せる(1)</li> </ul>

(本報告書における該当箇所)

## 10.2. 運営方法に関する課題とその解決策

ヒアリングとアンケート調査に基づき、プログラミング人材育成を行う事業者にとっての運営上の課題とその解決策についてまとめる。

### 10.2.1. 現状と課題

プログラミング教室・講座運営上の課題としては講師の確保、学習環境の確保、機器類の準備、教材の作成に関するものが主な意見として挙げられた。

#### 10.2.1.1. 講師の確保

一斉講義形式の教室・講座は少なく、受講者数人に対して1名の講師を割り当てたスタイルが多いため講師の人数確保が必要となる。現状では講師が教えられる人数の受講者を指導しているといった状態の事業者もいた。更には今後、プログラミングに関する教育の展開が進み、受講者数の増加に比例して講師数が不足することが考えられる。

現在、プログラミング講師には、ICT技術者や大学、高校等で情報系の専門コースに在席する学生が務めているケースが多い。また、ICT技術者や大学・高校の学生でプログラミングに関する教育の講師役に関心があったとしても、本職や学業との両立の難しさから諦めている場合もある。そのため講師には幾ばくかの謝金を支払い、少しでも講師になるハードルを下げたいと考える事業者もいる。また、そのような人材との接点がないことが課題となっている場合もあり、講師候補と講師募集事業者のマッチングがうまくできていないと考えられる。

講師の確保に関する意見は以下のとおりである。

### 講師の確保

- プールやそろばんは生徒数が増えても大丈夫。一方、プログラミング教室は講師数が必要。
- 生徒を受入れるためには、講師数を増やす必要がある。
- 平日午後開催のため、頻度を高めたくとも講師がいないため増やせない。
- 講師数の問題で、受講者を拡大しにくい。現在は教室の参加募集を行っていない。
- テキストベースの言語を使用する場合、アシスタントが多数必要となる。
- タイピングの壁はあるが、教材と人員でカバーしている。
- 受講者のレベルが上がると、ICTの専門知識をもつ講師が必要となる。

### 講師となる人材

- 講師は理系の大学学部生・院生が中心でほとんどがプログラミング経験者である。プログラミング能力とコミュニケーション能力の高い学生であり、市場価値は高い。
- プログラミング教育に関心を持つ高校専門学校や大学の学生をバイトとして雇いたい。
- スキルの高い指導者の継続的な活用も課題。現在は、不定期に大学の先生などと呼んでいる。
- 学生でも、指導できて、プログラミングもできるような子は優秀で、お金がかかる。
- ボランティア講師だけでは限界があるため、将来的には講師へ謝金をお渡ししたいと考えている。
- ICT技術者は会社の研修などで後輩にプログラミングの指導経験がある。指導することが好きな人も多い。
- 地元で講師向け講習会を開催したところ、地域のIT企業からの参加者が多かった。
- IT環境は常に変化し続けるため、講師は常に勉強し続けなければならない。

### 講師確保のための仕組み

- 講師は民間のエンジニアや大学と連携するネットワークを持っているため間に合っている。
- 副業として講師役ができるような仕組みが欲しい。講師希望者がいても牽引できる人事制度が少ない。会社を務めながら指導したいと思う人は多いだろう。CSRで担保するか。
- 保護者をボランティアとしてリクルートするための仕組みが欲しい。
- 地域のプログラマーを発掘できるようなものがあればよい。やろうとなったときに地域に見当たらないためできない。
- 教えたいと考える人は地域関係なく存在する。むしろ地方程取り組みたいと考えていることがある。しかし保護者や子どもへリーチするかが難しい。
- 最近はプログラミングワークショップに関する問い合わせが多く、様々な地域から要請をうけて講師向けの講習会を開催している。

## 10.2.1.2. 学習環境の確保

学習環境の確保に関する課題としては、会場確保のコストやネットワーク環境の整備の

難しさが挙げられた。とくにネットワーク環境については、プログラミング開発環境として、子どもたちが自分の手で調べながら学ぶために必須の環境だとする意見が複数事業者から挙げられた。

学習環境に関する意見は以下のとおりである。

#### 会場

- 東京都は都の施設を借りにくい。営利目的では公立施設を借りられないため、どうしても会場代が高くなってしまふ。
- 常設教室にイベント的に開催する場合、会議室代が高くついている。
- 会場は近隣大学の PC ルームを借りている。
- 中学校等の PC ルームをプログラミング教室のために開放できないか。
- 会場の提供を学校にもアプローチした場合、学校現場では好感触でも教育委員会の許可が下りないケースが多い。

#### ネットワーク環境

- 教室では WiMAX を使っているが高いビルになると使えない。ネットワーク環境を改善して欲しい。
- ネットワークへ接続できない環境でプログラミングを学習するのは不自然ではないが、公教育で教える場合も、インターネット接続環境で教えるべきだろう。
- 大勢が集中するために、集中的にインフラ整備を行うことが大変。
- ワークショップでは、端末全てをインターネットへ繋いでいる。会場には公衆 LAN 環境がないため、イベントのためにインターネット回線をひいて実施している。
- インターネット環境の整備について疑問に答える機会も提供していただけるとよい。

### 10.2.1.3. 機器類の準備

NPO 等非営利で活動する場合には端末の調達・設定が大きなインシヤルコストとなっていた。とくに RaspberryPi のような安価な PC 本体が入っても、モニターやキーボードなどの周辺機器類の確保が大きなコストとなっている。また、ICT 技術者がいない事業者では機器のセッティングが課題となっている。

機器類の準備に関する意見は以下のとおりである。

#### 機器の購入

- 事業の規模は、会場確保、生徒の需要、講師数の 3 要素によって決まる。
- Raspberry Pi の無償提供を受けたが、モニターやキーボードの確保が難しかった。寄付や運営費からねん出し、揃えた。
- 端末の調達はハードルになる。イベントではスポンサー企業から提供していただくが、それでも難しい場合がある。
- Raspberry Pi を 30 個をいただいた。ただ、ディスプレイやキーボードなどの方が高く、それらについては学校から 30 セット教育の廃棄用を譲渡してもらった。
- 市のイベントとして開催したときは、機材を教育委員会から提供してもらい、会場を市役所に用意してもらった。

#### 機器の設定

- 機材を設定し、使える状況を整備することは、IT に詳しい人がいない施設では難しいだろう。
- 大学生が技術的なサポートをしてくれるため、機器の整備にはあまり苦労は無い。

#### 10.2.1.4. 教材の作成

NPO 等非営利の活動を中心とする教育事業者にとっては教材の提供を望む声も上がっていたが、営利活動を行っている事業者からは反対の意見も上がっていた。

教材の作成に関する意見は以下のとおりである。

- 教材作成等を政府が実施することには違和感がある。教材は競争力と考える。
- Scratch の次に取り組むべきものがわかればありがたい。教材などが提供されていれば、活用できるかもしれない。
- 教材のクオリティを上げることは課題である。
- 小学校低学年はスマホやタブレットを持っていない。アプリやゲームだとつくったものを、実感できないという課題がある。
- 継続的なカリキュラムについては整理できていない。まずは入門編として、きっかけづくりがメインとなるだろう。次に踏み出す場合のカリキュラムを作ることが難しい財団が主体となって実施するよりも、大学や企業側の広報の一環として提供できないか。
- カリキュラムは個別の子どもに併せる必要があり、カスタマイズが必要となる。教材事例を共有できるような集まりの場を支援するのがよいのではないか。

#### 10.2.2. 運営方法における課題のまとめ

運営方法における課題としては、講師や会場、機器類の確保と教材作成が挙げられた。講師の不在によって教室を拡大できないとする事業者もいた。また、会場や機器類、教材にコストがかかることも、プログラミングに関する教育が普及しない一因であると考えられる。これらの解決策を検討するため、運営における課題とその解決のために参考となる取組みを以下の通り整理した。

表 10-2 各カテゴリにおける課題及び参考となる取組み

カテゴリ	課題	参考となる取組み
講師 (10.2.1.1)	<ul style="list-style-type: none"><li>・ 講師候補（地域の ICT 技術者や理系大学生）と講師募集事業者のマッチング</li><li>・ 将来的に講師数が不足する</li></ul>	<ul style="list-style-type: none"><li>・ 地域の大学生、専門学校生がアシスタント</li></ul>
学習環境 (10.2.1.2)	<ul style="list-style-type: none"><li>・ 会場費が高い</li><li>・ ネットワーク環境が必要</li></ul>	<ul style="list-style-type: none"><li>・ 地域の学校の PC ルームを利用</li></ul>
機器類	<ul style="list-style-type: none"><li>・ 機器類が高い</li></ul>	<ul style="list-style-type: none"><li>・ 学校等の廃棄ディスプレイ・キー</li></ul>

(10.2.1.3)	・ (ICT 技術者のいない事業者) 機器類の設定が難しい	ボードの譲り受け
教材 (10.2.1.4)	・ 継続的なカリキュラムが見えていない	

(本報告書における該当箇所)

### 10.3. 普及展開のための取組み案

10.1 および 10.2 にまとめた課題と解決策に基づき、今後期待される取組みの方向性を整理すると表 10-3 の通りである。

表 10-3 普及啓発及び教室運営における課題とその解決策

課題	課題分類	課題、取組み	解決策案
認知度・インセンティブに関する課題	普及啓発	デザインに問題がある、広告の信頼性が低い	<ul style="list-style-type: none"> <li>・ コンテストの開催</li> <li>・ 評価手法の策定</li> <li>・ 普及啓発パンフレットの作成</li> </ul> (施策案①)
		保護者の ICT リテラシーが高くないと学習効果、メリットが伝わりにくい	
		IT 企業やプログラミングに対する偏見 明確な目標、成果発表の場がない	
地域に関する課題	普及啓発 運営	地方で盛り上がっていない	<ul style="list-style-type: none"> <li>・ 地域コミュニティとの連携</li> <li>➢ 施設活用</li> <li>➢ 講師マッチング</li> </ul> (施策案②)
		講師候補（地域の ICT 技術者や理系大学生）と講師募集事業者のマッチングや（将来的に）講師数の不足	
		会場費が高い ネットワーク環境が必要	
プラットフォームに関する課題	運営	機器類が高い 機器類の設定が難しい 継続的なカリキュラムが見えていない	<ul style="list-style-type: none"> <li>・ 共通して利用できるプラットフォームの活用</li> <li>➢ ポータルサイト構築</li> </ul> (施策案③)

課題は大きく分けて、認知度・インセンティブに関する課題、地域に関する課題、プラットフォームに関する課題に分けることができ、それぞれ

- ① コンテスト等の開催、評価手法の策定等
- ② 地域コミュニティとの連携による施設活用・講師マッチング
- ③ 共通して利用できるプラットフォームの活用

の 3 施策が有効だと考えられる。以下で、それぞれの施策について説明する。

#### 10.3.1. モチベーションの向上および効果に関する認知度の向上

普及展開においては、参加者のモチベーションの向上と効果に関する認知度の向上を図ることが効果的である。

コンテスト等のイベント開催は、青少年のモチベーションの向上に効果的であるが、個々の組織によるコンテストの開催には限界があり、現状では実施できる場所は限られている。そのような中で、全国規模で開催する公的なコンテストを支援することで、参加者にとって目指すべき目標となり継続参加のモチベーションを高めることができると考えられる。また、保護者に対しては公的なイベントを大々的に告知することによって、プログラミングに対するイメージの向上や認知度の向上が期待できる。

例えば、LEGO マインドストーム EV3 の教室は全国に数十か所以上広がり、生徒の継続率も高いが、その一因として世界大会「World Robot Olympiad<sup>52</sup>」の存在がある。予選大会で勝ち抜いたチームによる日本大会、そしてその後世界大会へと繋がるロボット競技会があるため、子どもや保護者からの支持を集めている。国内で、プログラミングに関するコンテストとしては、U-22 プログラミングコンテストや中高生国際 Ruby プログラミングコンテスト<sup>53</sup>等があるが、対象年齢が比較的高い（中学生以上が中心）ことや、教育用プログラミング言語を使用したコンテストではないため、応募する数居が高いといえる。ヒアリング等でも、小学生から参加できる間口の広いコンテストを望む意見は複数挙がっており、そのようなコンテストを国が開催することで、取り組むためのモチベーション向上に繋がると考えられる。

また、普及啓発のためには、プログラミングに関する教育の目指す方向性や効果を伝えることが重要である。現状では、効果を客観的に評価する方法が整っていないが、他の教科と同様に、獲得した能力やスキルを客観的に評価するテストなどを開発することで、効果を視覚的に把握することが期待できる。21 世紀型能力は従来のペーパーテストでの評価が難しいため、ICT 機器類の操作ログを活用し、思考プロセスを評価することが想定される。

併せて、教育事業者がイベントや教室の広告と共に配布できるような、教育の目指す方向性や効果等についてわかりやすく整理したパンフレットの作成は有効である。総務省が作成したパンフレットによって事業者の広告や取組みに対する信頼性が上がり、プログラミング教室へ関心をもつ保護者も増えると考えられる。

### 10.3.2. 地域コミュニティとの連携による地域格差の解消

調査結果から、プログラミング人材育成の地域格差は大きく、東京など一部地域の盛り上がりに対して、地方では認知度が低く生徒数が少ない傾向がある。プログラミング人材育成は、国の地域創生事業で求められている地域人材育成や地域産業の活性化へと繋がる効果が期待できるため、特に首都圏以外の地域での取組みを支援することは有効であると考えられる。

そのためには、地方でのプログラミング人材育成について 10.3.1 のような取組みをとおして普及啓発しつつ、事業者を支援することが望まれる。特に地方の活動拠点に関する取組みとしては、小中学校の施設提供や学童クラブとの連携が考えられる。ICT 技術者がいない事業者では機器の確保や設定が課題となっており、学童クラブ等の地域の団体によるプログラミング教室を支援するためには、プログラミング体験用機器の無料貸し出し、不

<sup>52</sup> LEGO Education 世界につながるロボット大会 (2014 年 3 月アクセス)

<https://education.lego.com/ja-jp/preschool-and-school/secondary/mindstorms-education-ev3/competitions>

<sup>53</sup> 中高生国際 Ruby プログラミングコンテスト 2014 in Mitaka

<http://www.mitaka.ne.jp/ruby/>



要な周辺機器類の提供等によって、イベント・教室の開催ハードルを下げる必要がある。

他にはプログラミング人材育成に関する取り組みの支援を希望する組織・人材と教育事業者をマッチングさせるような仕組みの提供も有効だと考えられる。特に、地域におけるプログラミング人材育成の認知度が向上し受講者数が増えれば、講師数の確保が課題となる恐れがある。特に地方では IT 企業が少なく、ICT スキルを持った人材が少ない傾向にある。地方での普及展開のためには、ICT スキルを持っていない人材でもプログラミング教室を開催できるように、講師人材育成も重要なテーマになると考えられる。

すでに事業者側が主体となり、そのような情報交換コミュニティを立ち上げている事例がある。例えば宮城県石巻市の「イトナブ石巻<sup>54</sup>」の取り組みは、「震災 10 年後の 2021 年までに石巻から 1000 人の ICT 技術者を育成する」ことを目標に掲げ、地域の若者を対象にプログラミングやウェブデザイン等を学ぶ機会を提供している。多数の ICT を学ぶワークショップが開催され、盛り上がりを見せている。このようにすでに地域にコミュニティがある場合は、その拠点の支援、そのようなコミュニティがない地域に関してはコミュニティの立ち上げ支援を行うことが有効だと考えられる。

### 10.3.3. 共通して利用できるプラットフォームの活用

教育ツールや教材など教育環境を個々の教育事業者が整備するためには大きな負担が生じる。インターネットを介して全国から共通して利用できる教育プラットフォームを整備・活用することで、全国レベルで教育事業を行う際の障害が低くなり、少ない負担で、効果的な教育を実現することができるようになる。

普及展開のための取り組みに関する情報や地域のコミュニティの支援を行う上で、プログラミング人材育成に関する情報をまとめたプラットフォームの提供は有効である。プラットフォーム上に地域ごとの教室、イベント情報の掲載によって、プログラミングに関心をもった子ども、保護者がまとまった情報を得ることができる。また、事業者にとっても他の事業者と情報を交換できる場ができることで、教材や指導方法に関するアイデアの共有、相談等も可能となる。

海外では「Code.org<sup>55</sup>」の活動が有名である。学習者への情報提供、講師側には情報共有の場を提供している。また、Scratch が評価されている理由として、Scratch コミュニティの存在が挙げられていた。これらの取り組みのように、情報や作品を共有するプラットフォームによって、活動に相乗効果が生まれると考えられる。

### 10.3.4. その他の有効な取り組み案

前述の大きな施策案以外に、プログラミングに関する教育の普及展開において有効だと考えられる取り組み案をまとめた。

- 国際交流イベントを通じたモチベーションの向上

各国の同世代の子供たち（小中高校）によるプログラミング成果発表会を開催し、

---

<sup>54</sup> イトナブ <http://itnav.jp/>

<sup>55</sup> Code.org <http://code.org/>

各国の様々なアイデアに基づく成果を、プログラミングという世界共通の言語を用いた交流により促進することで、参加者の好奇心とモチベーションを高めるとともに、グローバル人材の育成につなげる。また、メディアの注目を集めることで、報道による認知度の向上にもつなげる。

- **シニア ICT 人材によるプログラミング教室**

ICT 技術者として活躍してきた人材が定年を迎え始めている。技術力も活力もあるシニア ICT 人材が地域の子どもたちへプログラミング教室を開催する。シニアと子どもがプログラミングを通じて世代間交流することでお互いに刺激を得ると考えられる。また、定年まで職場での人間関係を中心としていたシニアが地域内で人間関係を築く機会にもなると考えられる。それ以外にも、医療費削減、子育て支援、地域人材育成等の効果が期待できると考えられる。

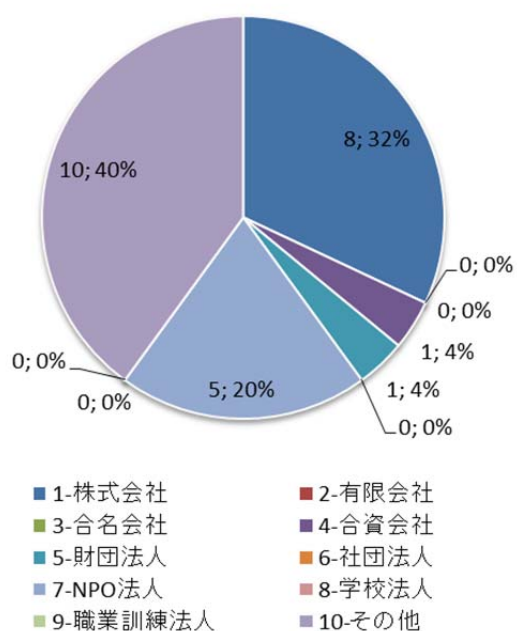
## A. 付録

### 1. プログラミングに関する教育の現状・実態アンケート

有効回答数は25件であった。

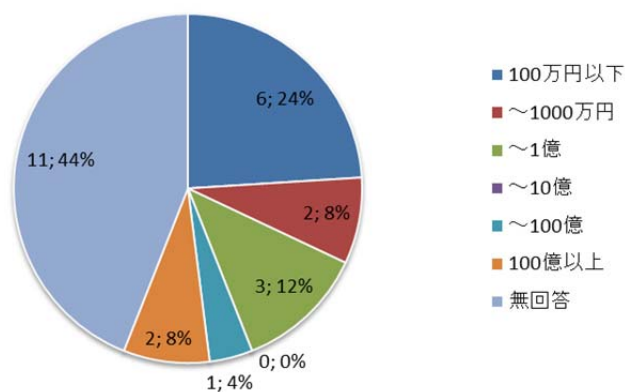
問1 経営形態として最も近いものを選択してください。

選択肢	回答数	(%)
1-株式会社	8	32%
2-有限会社	0	0%
3-合名会社	0	0%
4-合資会社	1	4%
5-財団法人	1	4%
6-社団法人	0	0%
7-NPO法人	5	20%
8-学校法人	0	0%
9-職業訓練法人	0	0%
10-その他	10	40%



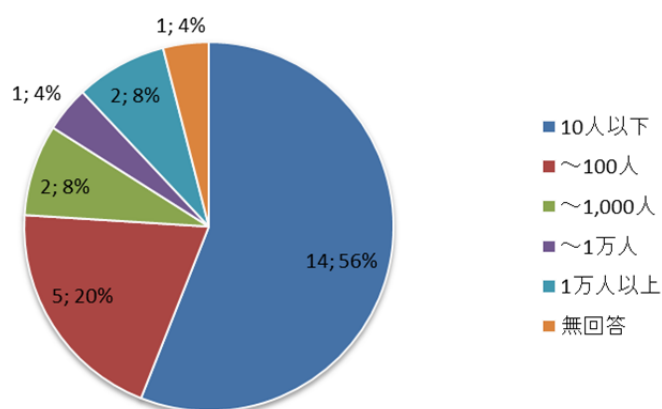
問2 組織全体の直近の1年間の売上高を分かる範囲で記入してください。

選択肢	100万円以下	～1,000万円	～1億円	～10億円	～100億円	100億円以上	無回答
回答数	6	2	3	0	1	2	11
(%)	24%	8%	12%	0%	4%	8%	44%



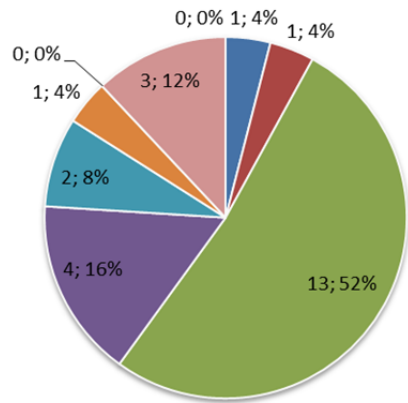
問3 組織全体の従業員数を記入してください。

選択肢	10人以下	～100人	～1,000人	～1万人	1万人以上	無回答
回答数	14	5	2	1	2	1
(%)	56%	20%	8%	4%	8%	4%



問4 本社所在地エリアを選択してください。

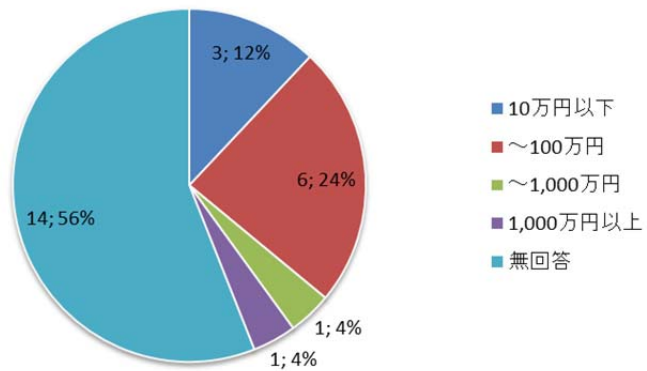
選択肢	1-北海道	2-東北	3-関東	4-中部	5-近畿	6-中国	7-四国	8-九州	9-海外
回答数	1	1	13	4	2	1	0	3	0
(%)	4%	4%	52%	16%	8%	4%	0%	12%	0%



- 1-北海道
- 2-東北
- 3-関東
- 4-中部
- 5-近畿
- 6-中国
- 7-四国
- 8-九州
- 9-海外

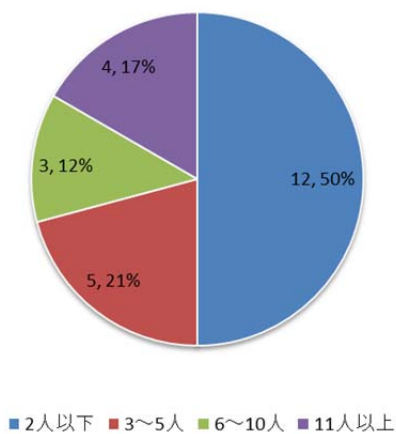
問5 プログラミング教室・講座に関する事業の直近1年間の売上高を記入してください。

選択肢	10万円以下	～100万円	～1,000万円	1,000万円以上	無回答
回答数	3	6	1	1	14
(%)	12%	24%	4%	4%	56%

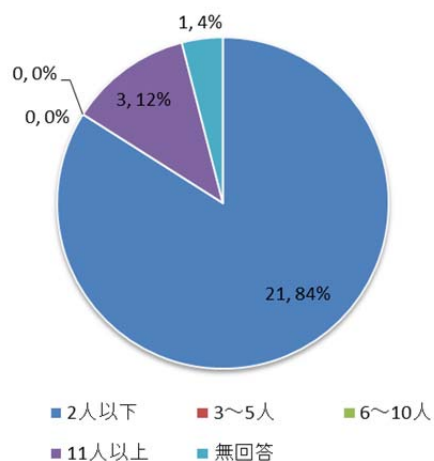


問6 プログラミング教室・講座に関する従業員（事務系と講師）のおおよその人数を記入してください。

講師数  
(常勤・非常勤・学生アルバイト含む)

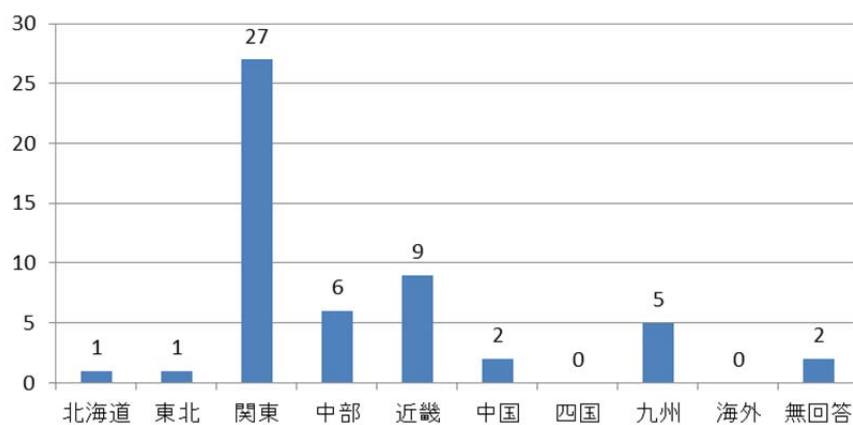


事務系職員数  
(常勤・非常勤・学生アルバイト含む)



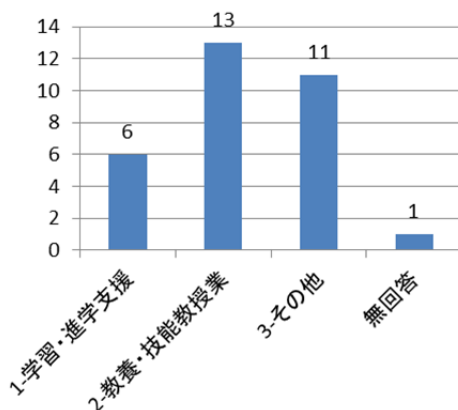
問7 プログラミング教室・講座を実施する教室の数を地域ごとに記入してください。

選択肢	1-北海道	2-東北	3-関東	4-中部	5-近畿	6-中国	7-四国	8-九州	9-海外	無回答
回答数	1	1	27	6	9	2	0	5	0	2



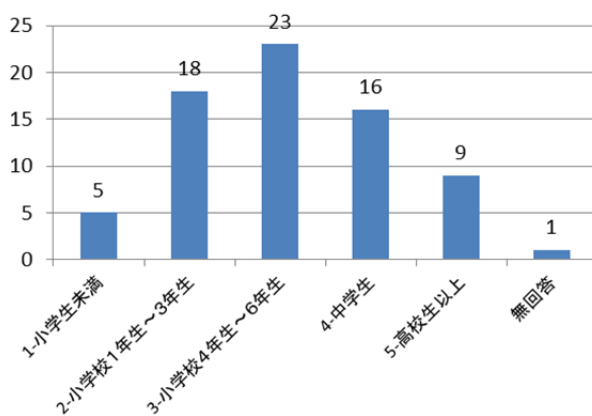
問8 プログラミング教室・講座に関する事業の分野について、あてはまるものを全て☑してください。

選択肢	回答数
1-学習・進学支援	6
2-教養・技能教授業	13
3-その他	11
無回答	1



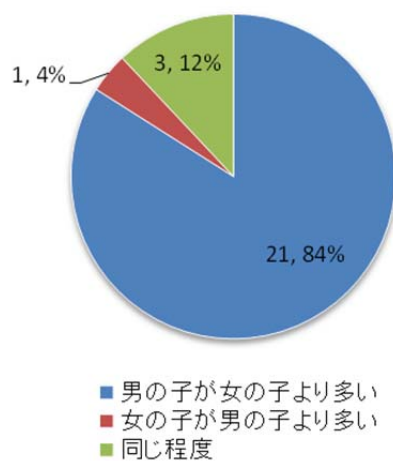
問9 プログラミング教室・講座に関する教室・講座の対象年齢についてあてはまるものに全てチェック☑してください。

選択肢	回答数
1-小学生未満	5
2-小学校1年生～3年生	18
3-小学校4年生～6年生	23
4-中学生	16
5-高校生以上	9
無回答	1



問 10 プログラミング教室・講座に関する教室・講座に参加する男女のおおよその比率をご記入ください。

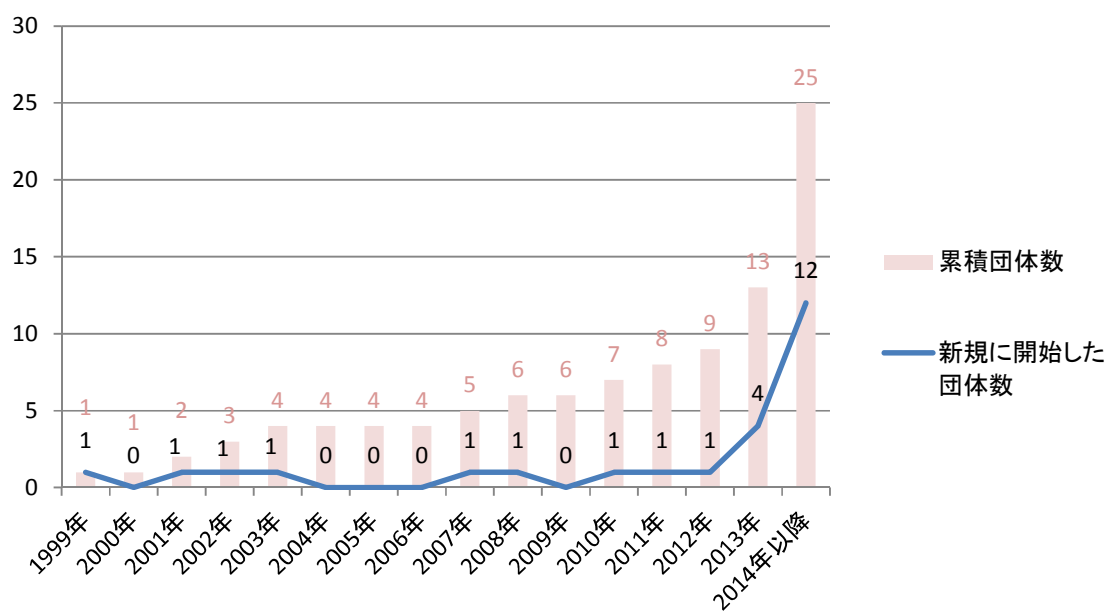
選択肢	回答数
男の子が女の子より多い	21
女の子が男の子より多い	1
同じ程度	3





問 11 プログラミング教室・講座を始めた時期をご記入ください。

開始年	回答数	累積数
1999年	1	1
2000年	0	1
2001年	1	2
2002年	1	3
2003年	1	4
2004年	0	4
2005年	0	4
2006年	0	4
2007年	1	5
2008年	1	6
2009年	0	6
2010年	1	7
2011年	1	8
2012年	1	9
2013年	4	13
2014年以降	12	25



問 12 プログラミング教室・講座を始めたきっかけは何ですか。きっかけとなったことを具体的にご記入ください。

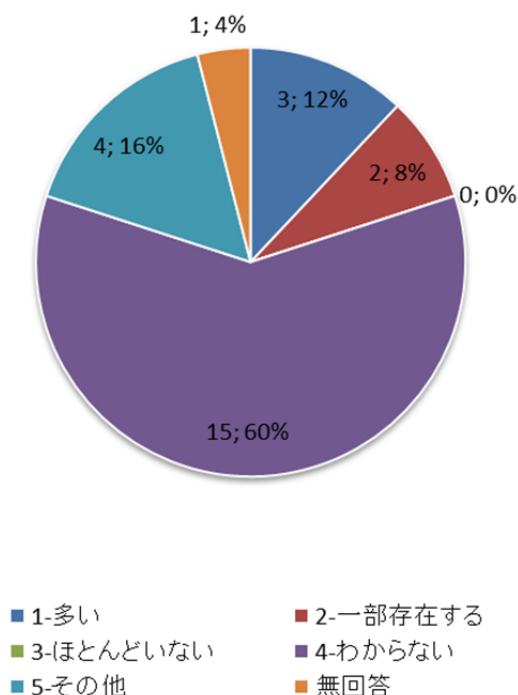
(自由記述のため割愛)

問 13 プログラミング教室・講座でどのような人材を育てることを目指していますか。具体的な能力や人材像について、ご記入ください。

(自由記述のため割愛)

問 14 プログラミング教室・講座の修了生で、高等専門学校や大学の情報系学科・学部等へ進む人はいますか。

選択肢	回答数
1-多い	3
2-一部存在する	2
3-ほとんどいない	0
4-わからない	15
5-その他	4
無回答	1

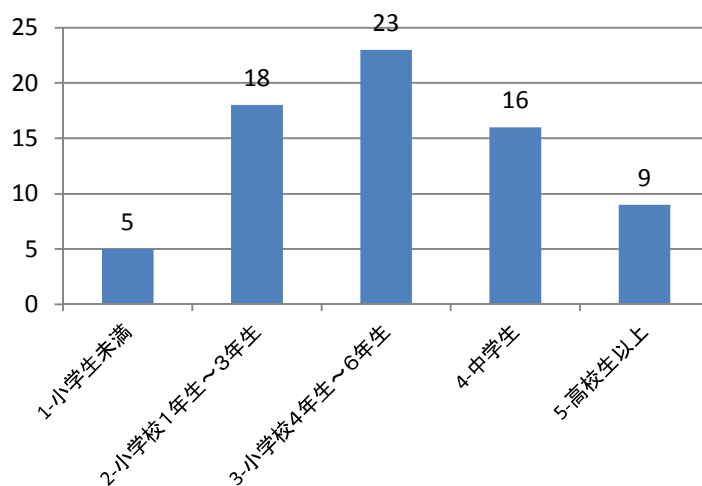


問 15 プログラミング教室・講座の種類（コース）とそれぞれの教材にはどのようなものがありますか。

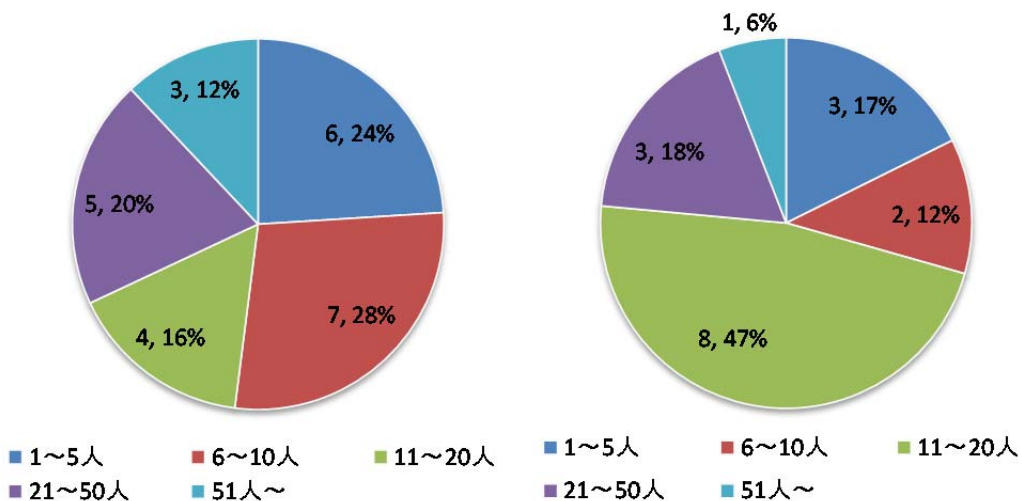
1) コース・イベント名

(自由記述のため割愛)

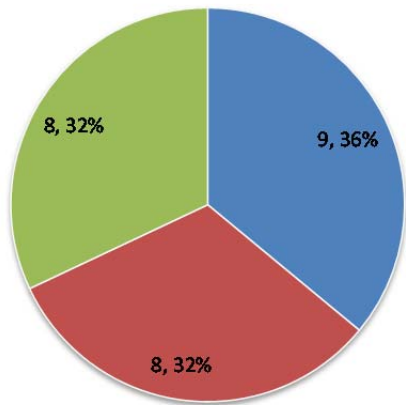
2) 対象年齢



3) 受講者数

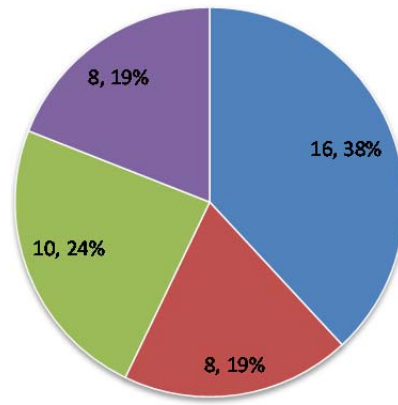


4) 頻度・所要時間、学習形態、受講形式



- 定期的な教室・講座/イベントを実施
- 定期的な教室・講座のみ実施
- イベントのみ実施

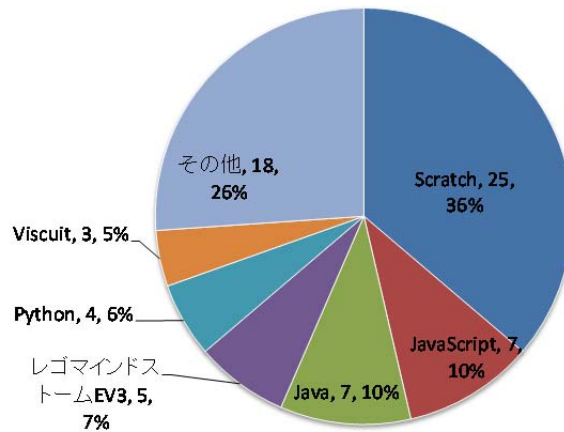
教室・講座の実施形態



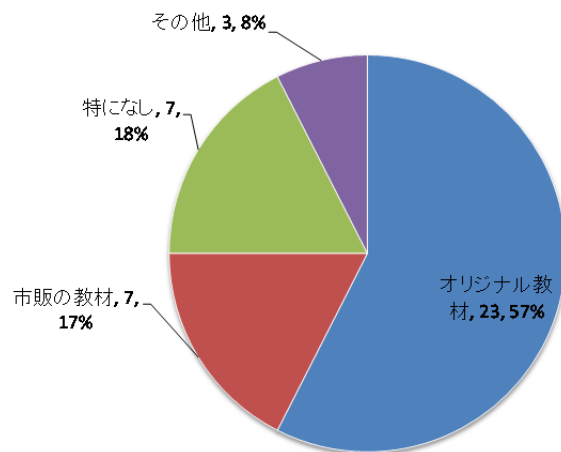
- 1-個別学習: 講師と受講生が1対少数
- 2-集団学習型: 講師による講義形式
- 3-演習型: 講師形式でグループ作業
- 4-その他

教室・講座の受講形式(複数回答可)

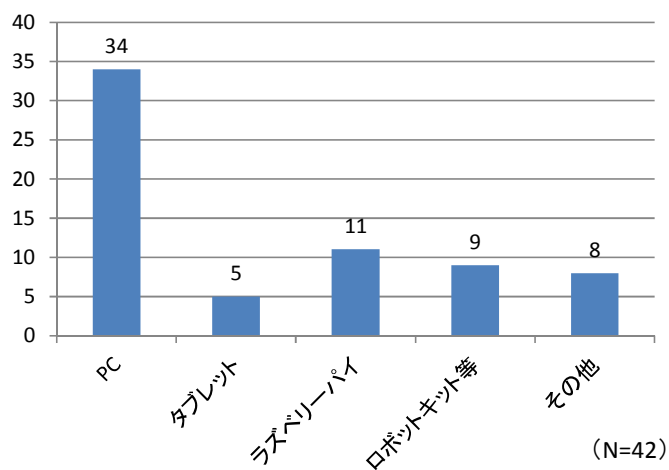
5) プログラミング言語



6) 中心となるテキスト教材



7) 使用機器

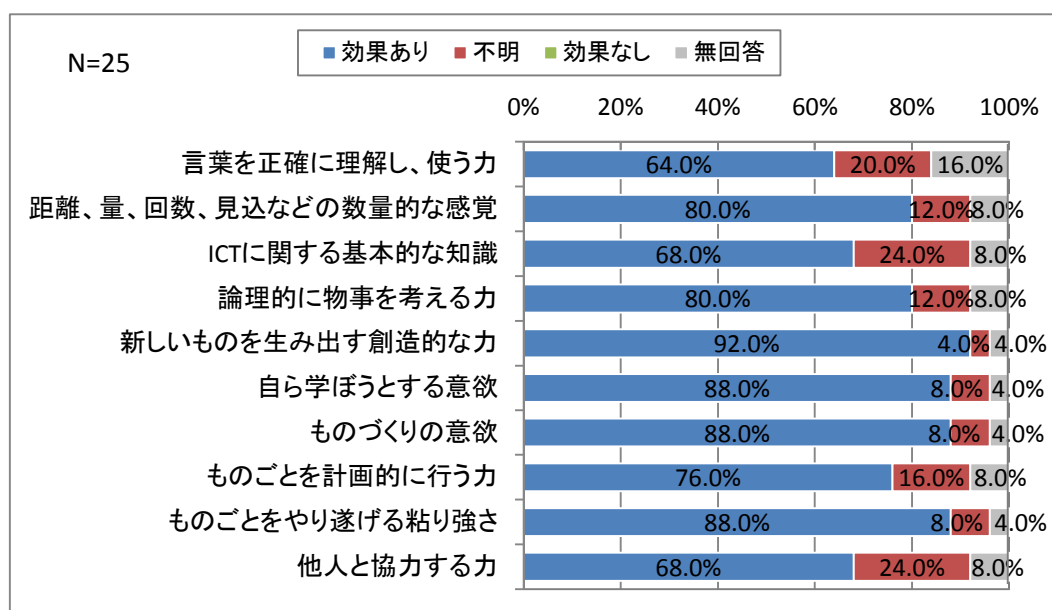


問 16 プログラミング教室・講座の教育全体の方針や指導方法の特徴および講座ごと指導方法のアピールポイントなどあれば、具体的にご記入ください。

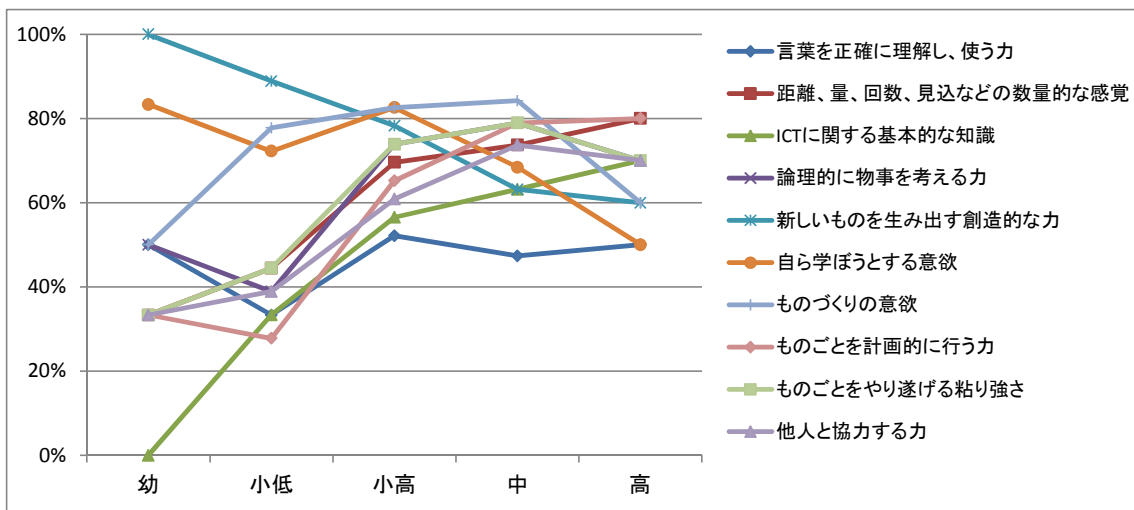
(自由記述のため割愛)

問 17 プログラミング教室・講座によって得られる・伸びると思われる能力や意欲について、各能力等についてどの程度効果があると思いますか。また、各能力等を伸ばすための学習段階として効果的だと感じられる年代についてチェック☑してください。

能力	効果がある	効果がない	わからない	無回答
	回答数	回答数	回答数	回答数
言葉を正確に理解し、使う力	16	0	5	4
距離、量、回数、見込などの数量的な感覚	20	0	3	2
ICTに関する基本的な知識	17	0	6	2
論理的に物事を考える力	20	0	3	2
新しいものを生み出す創造的な力	23	0	1	1
自ら学ぼうとする意欲	22	0	2	1
ものづくりの意欲	22	0	2	1
ものごとを計画的に行う力	19	0	4	2
ものごとをやり遂げる粘り強さ	22	0	2	1
他人と協力する力	17	0	6	2



小学生未満		小学校 1～3 年生		小学校 4～6 年生		中学生		高校生	
回答数	割合	回答数	割合	回答数	割合	回答数	割合	回答数	割合
3	50.0%	6	33.3%	12	52.2%	9	47.4%	5	50.0%
2	33.3%	8	44.4%	16	69.6%	14	73.7%	8	80.0%
0	0.0%	6	33.3%	13	56.5%	12	63.2%	7	70.0%
3	50.0%	7	38.9%	17	73.9%	15	78.9%	7	70.0%
6	100.0%	16	88.9%	18	78.3%	12	63.2%	6	60.0%
5	83.3%	13	72.2%	19	82.6%	13	68.4%	5	50.0%
3	50.0%	14	77.8%	19	82.6%	16	84.2%	6	60.0%
2	33.3%	5	27.8%	15	65.2%	15	78.9%	8	80.0%
2	33.3%	8	44.4%	17	73.9%	15	78.9%	7	70.0%
2	33.3%	7	38.9%	14	60.9%	14	73.7%	7	70.0%

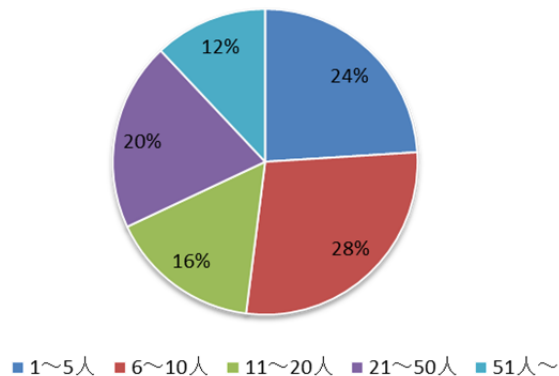


問 18 上記の項目以外の効果や、とくに効果が大いとおもう点について自由にご記入ください。

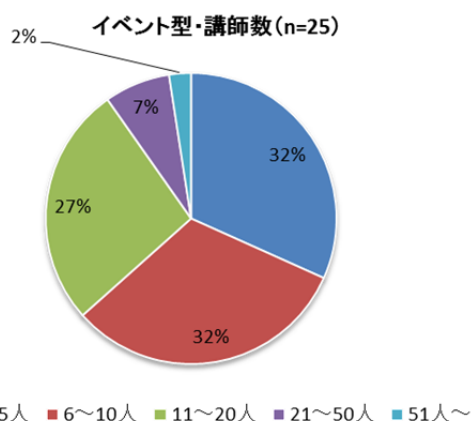
(自由記述のため割愛)

問 19 プログラミング教室・講座の講師について、おおよその人数をご記入ください。いない属性については「0」人としてください。

定期的・講師数(n=17)



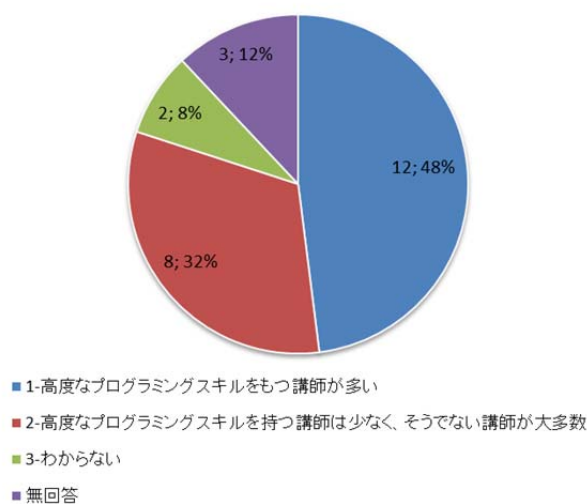
※定期的：月に数回以上の受講日を設定するプログラミング教室における講師数



※イベント型：1日～数日の期間中に集中して受講するプログラミング講座における講師数

問 20 高度なプログラミングスキルをもった講師（例：情報系学部出身者、IT系企業・IT部門出身者、個人でアプリを開発する等）とそうでない講師との割合について、最も近いものを選択してください。

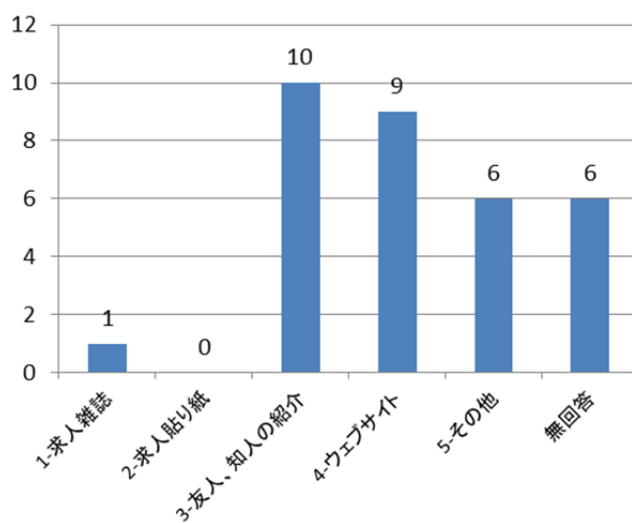
選択肢	回答数
1-高度なプログラミングスキルをもつ講師が多い	12
2-高度なプログラミングスキルを持つ講師は少なく、そうでない講師が大多数	8
3-わからない	2
無回答	3





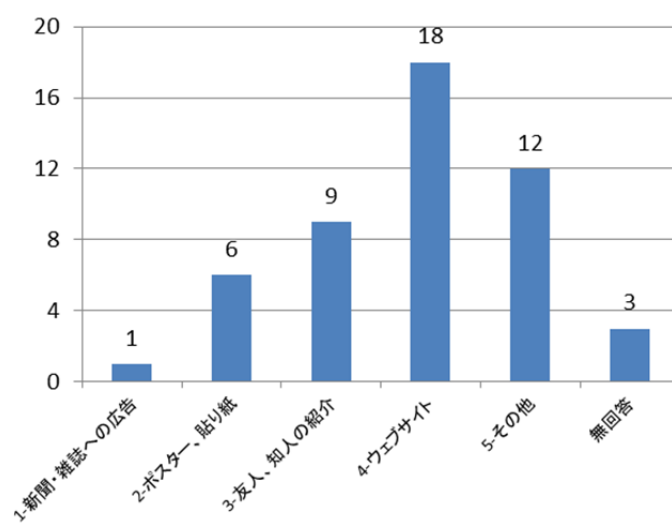
問 22 プログラミング教室・講座の講師はどのように募集していますか。あてはまるものにチェック☑してください。

選択肢	回答数
1-求人雑誌	1
2-求人貼り紙	0
3-友人、知人の紹介	10
4-ウェブサイト	9
5-その他	6
無回答	3



問 25 プログラミング教室・講座の受講生はどのように募集していますか。あてはまるものにチェック☑してください。

選択肢	回答数
1-新聞・雑誌への広告	1
2-ポスター、貼り紙	6
3-友人、知人の紹介	9
4-ウェブサイト	18
5-その他	12
無回答	3

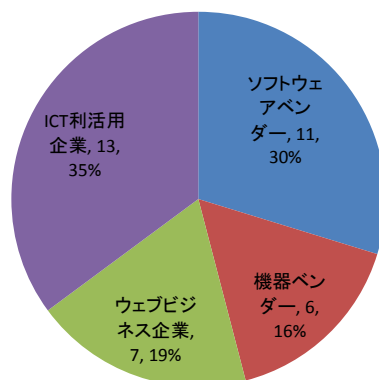


## 2. 企業ニーズアンケート

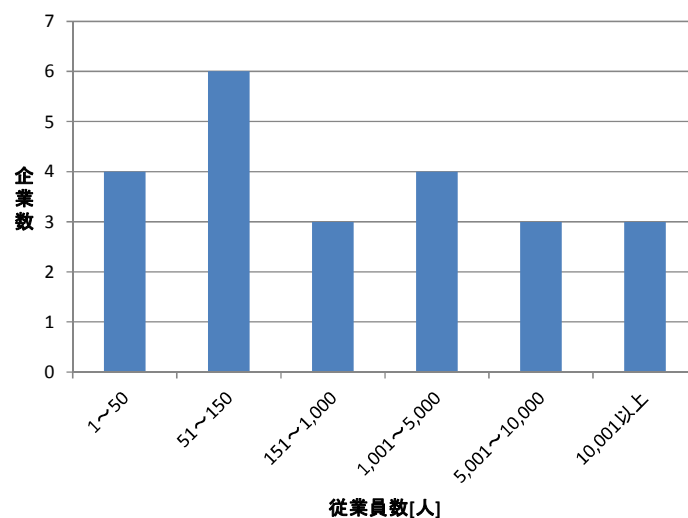
企業が求めるプログラミング人材像について、ベンダおよび ICT 利活用企業の幅広い分野の企業を対象にアンケート調査を実施し、得られた回答をもとにまとめる。アンケートは 116 社に依頼し、有効回答数は 23 社（有効回答率 20%）であった。

### (1) 組織の属性

選択肢	ソフトウェアベンダー	機器ベンダー	ウェブビジネス企業	ICT 利活用企業
回答数	11	6	7	13
(%)	48%	26%	30%	57%



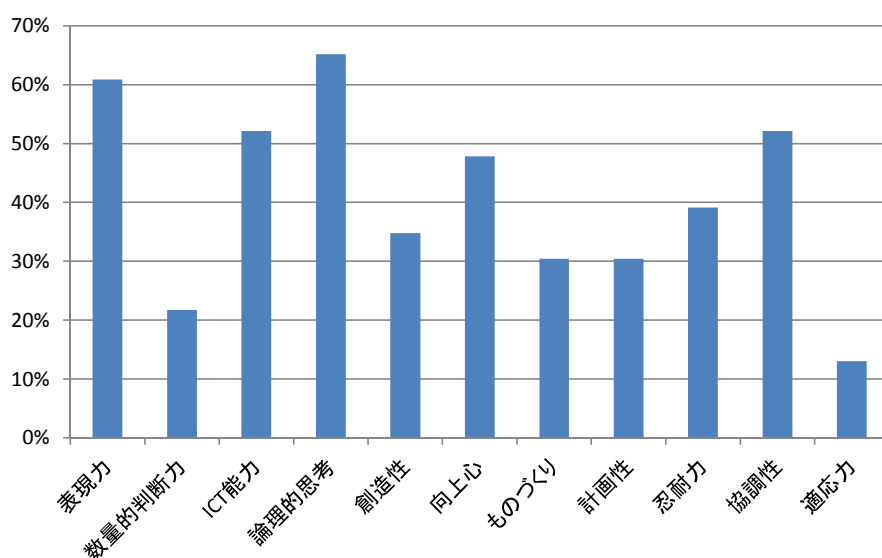
回答企業の従業員数は以下の通り。



## (2) 期待する能力

プログラミングに関する教育を通じて青少年期に身に付けることが期待される能力や資質のうち、以下の項目から貴組織にとって特に重要と考えられるものを選択してください。

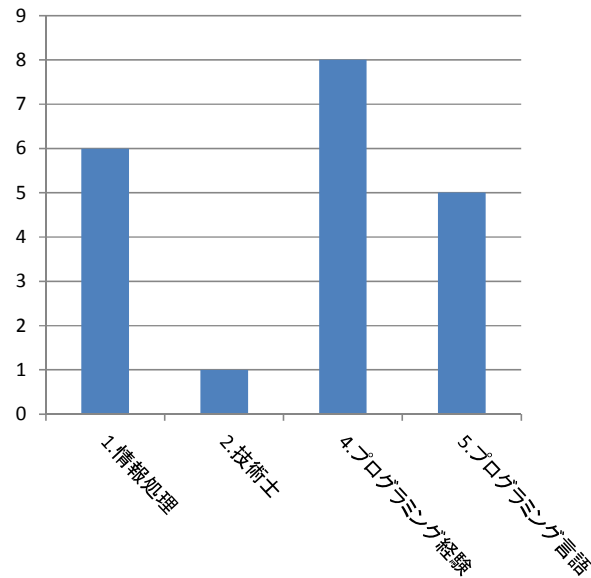
	基礎力			思考力				実践力			
選択肢	表現力	数量的判断力	ICT能力	論理的思考	創造性	向上心	ものづくり	計画性	忍耐力	協調性	適応力
回答数	14	5	12	15	8	11	7	7	9	12	3
%	61%	22%	52%	65%	35%	48%	30%	30%	39%	52%	13%



## (3) 採用時の ICT に係わる評価項目

以下の項目のうち、採用時に評価する項目があれば、該当する番号を選択してください。

選択肢	1.情報処理	2.技術士	4.プログラミング経験	5.プログラミング言語
回答数	6	1	8	5



#### (4) 求める人材の不足数

不足がある人材のおおよその人数をご記入ください。

(不足があると回答した企業の比率)

人材※		人材不足とする企業の比率	環境の変化時の採用を検討する企業の比率
主に技術系	プロジェクトマネージャ	56%	57%
	システム・コンサルタント	63%	50%
	設計・アーキテクト	71%	60%
	プログラミング技術者	75%	67%
主に利活用系	システム管理・運用者	20%	0%
	企画・要求分析者	43%	33%
その他		50%	33%

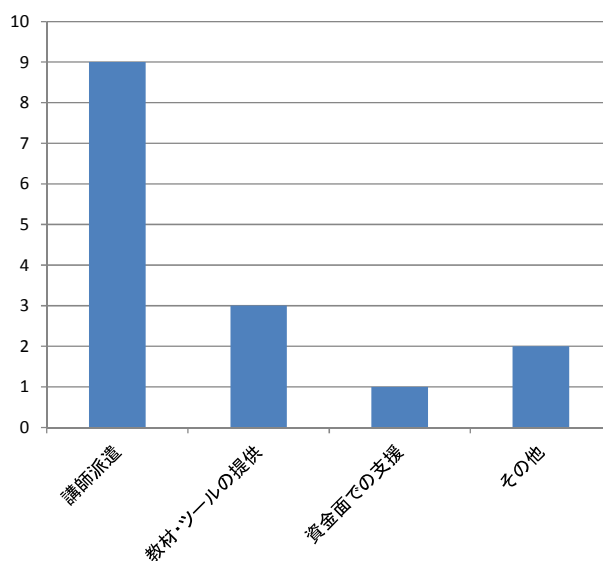
(人材の不足数に関する集計)

人材※		在籍者数 (総数)	不足数 (对在籍者数比率)	環境の変化時の 採用数 (对在籍者数比率)
主に技術系	プロジェクトマネージャ	1236人	21人(1.7%)	8人(0.6%)
	システム・コンサルタント	405人	30人(7.4%)	6人(1.5%)
	設計・アーキテクト	5866人	17人(0.3%)	10人(0.2%)
	プログラミング技術者	1060人	53人(5.0%)	42人(4.0%)
主に利活用系	システム管理・運用者	495人	5人(1.0%)	0人(0.0%)
	企画・要求分析者	106人	8人(7.5%)	2人(1.9%)
その他		150人	11人(7.3%)	1人(0.7%)

(5) 青少年へのプログラミング人材育成の支援

教育イベントやコースにおいて貴組織が支援したい／支援してもよいと考える項目を選択してください。

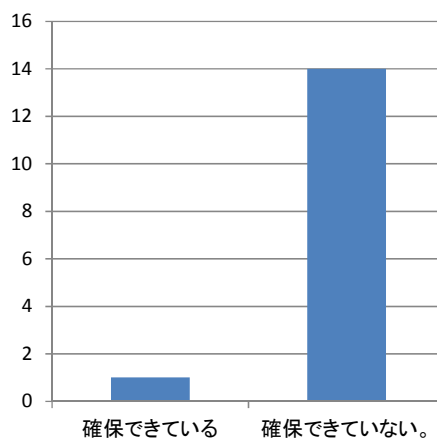
選択肢	講師派遣	教材・ツールの提供	資金面での支援	その他
回答数	9	3	1	2



(6) プログラミングに関する教育を通じて育成される人材の理想像

理想的な人材は、現状で十分確保できていますか。

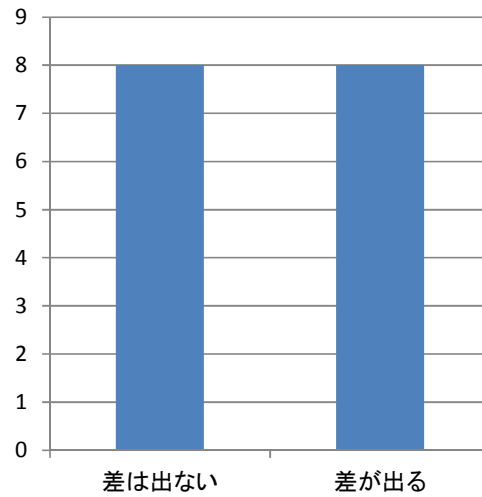
選択肢	確保できている	確保できていない
回答数	1	14



### (7) キャリアパス

青少年期にプログラミングに関する教育を受けた人材と受けていない人材で、キャリアパスに差が出ると思いますか。

選択肢	差は出ない	差が出る
回答数	8	8



以上