

平成 28 年度

総務省プログラミング教育実証

実施団体別報告書

■ 実施ブロック【沖縄地区】

■ 子供の自発的な気づきと参画を促す

「じんぶなー」育成モデル

平成 29 年 2 月 15 日

公益財団法人 学習ソフトウェア情報研究センター

特非 沖縄県マルチメディア教育研究会

株式会社 エウレカスイッチ・インタラクティブ、株式会社 電腦商会

1.モデルの概要

1.1 モデル名称

子供の自発的な気づきと参画を促す実践的指導案やプログラミング教育の評価指標、評価ツールの開発「じんぶなー」育成モデル

1.2 モデルの全体概要

プログラミング教育については教育効果の評価手法が確立されていないため、実際にプログラミング教育を実施する際に、子供たちに効果があったのか、メンターの指導内容が適切であったか、などが評価されているとはいえない。

そこで本事業では、プログラミング教育を通して子供たちに育みたい能力を体系的に定義したうえで、評価指標を設計し自己評価ツールを作成することで、メンターによるプログラミング教育効果の可視化を目指した。

なお、本モデルの名称である「じんぶなー」は、沖縄の方言（ウチナーグチ）で、「カッコいい人・できる人」を意味する。子どもたちに、「プログラミングができるってカッコいい！」ということを強調したいとの現地メンターの思いから名付けられた。

本事業では、ベースとしてプログラミングに関する全体スキルマップを設計した。その中から「コーディング・プログラミング能力」を中心として、今回のカリキュラムで取り扱うスキル領域を選択した。（図1）

21世紀型能力	プログラム教育が目指すスキル	スキル領域	レベル1	レベル2	レベル3
基礎力	表現力・コミュニケーション力	表現力	基本的な表現手法で自分の作品を作ることができる	既にある作品の表現を真似てみたり、様々な表現手法を使うことができる	様々な表現手法を駆使して自分なりの表現を行う事ができる
	数量的な判断力	数量的な判断力			
	ICT・情報を扱う能力	コーディング・プログラミング能力	具体的、視覚的操作ができる	論理的思考力	抽象的思考力
コンピュータの原理に関する理解力		コンピュータの基本的な動作を理解している	必要な外部機器をつなぐ等	様々な組合せでコンピュータを使いこなすことができる	
思考力	論理的・批判的思考力	合理性、論理的思考力 課題解決力			
	創造性・独創性	創造力			
	向上心・学習意欲	意欲の向上	与えられたテーマに対してプログラミングに取り組むことができる	自らテーマを持って積極的にプログラミングに取り組むことができる	自分だけでなく、まわりの仲間たちも含めてプログラミングに取り組むことができる
	ものづくりに対する好奇心				
実践力	計画性・管理能力				
	忍耐力・根気				
	協調性・連帯感				
	適応力・社会参画意欲				

図1 プログラミング全体スキルマップ

この全体スキルマップを元にして、コーディング・プログラミング能力のスキル領域を中心として、プログラミング講座で育成するスキルをどのような視点から評価するかを定義し、プログラミングスキル評価シートを作成した（図2）。なお、想定される学習者のレベルとして、1 入門レベル（初めてコンピュータに触れる）・2 初級レベル（コンピュータの操作経験はあるが、プログラミングは初めて）・3 応用レベル（プログラミングの経験を有する）を設定している。

スキル領域	学習者のレベル	コンピタンス	評価内容	評価基準
コーディング・プログラミング能力	1	具体的、視覚的操作ができる	プログラミングツールを使う事ができる	ツールの基本的な操作を実行できる
			動き、見た目を変更できる	動き、見た目を思った通りに変更できる
			基礎的な制御を使う事ができる	制御を使って思った通りに動作させる事ができる
	2	論理的思考力	繰り返し処理を行う事ができる	繰り返し処理を使ってプログラムを動作させる事ができる
			条件によって動きを変える事ができる	条件に応じた動作をさせる事ができる
			もし・ならばの構文を使う事ができる	もし・ならばの構造を正しく動作させる事ができる
	3	抽象的思考力	変数を使う事ができる	変数を作って動作させる事ができる
			プログラムの仕組みを作る事ができる	要素に分けて考える事ができる
			分割されたプログラムを関連させて使う事ができる	モジュールを組み合わせて動作させる事ができる

図2 プログラミングスキル評価シート（汎用）

プログラミングスキル評価シートは、特定の言語や教具に限定されないよう汎用にデザインしているが、本事業ではプログラミング講座の教材としてビジュアル言語のScratchを使用したため、評価内容と評価基準をScratchの環境に合わせたものに書き改めた。（図3）

その際に、学情研が開発しWebにて一般公開しているScratchのデジタル教材に収録されているサンプルプログラムが、スキル項目のどれに該当するかを提示することで、メンターが事前に該当部分のプログラムを操作しておくことによりプログラミング講座の指導イメージを得ることも可能なように配慮している。

レベル	レベル詳細	評価内容	Scratchの評価基準	該当サンプル
1	具体的、視覚的 操作ができる	プログラミングツールを使う事ができる	Scratchの基本的な操作を実行できる	Scratchのメニュー Scratchのプログラミング ブロックをつなげてみよう ハードルを作ろう ネコの鳴き声を作ろう
		動き、見た目を変更できる	動き、見た目のブロックを使う事ができる	ネコを歩かせよう ネコを回転させよう ネコの位置を変えてみよう ネコをマウスで移動させよう ネコにしゃべらせよう ネコの絵を変えてみよう ネコの色を変えてみよう ネコの大きさを変えてみよう
		基礎的な制御を使う事ができる	制御ブロックを使う事ができる	キーボードでネコを動かそう
2	論理的思考力	繰り返し処理を行う事ができる	繰り返し処理を使ったプログラムを作れる事ができる	同じ処理を繰り返そう
		条件によって動きを変える事ができる	制御ブロックを使って条件に応じた動作をさせる事ができる	条件によって動きを変えよう ネコをジャンプさせよう
		もし・ならばの構文を使う事ができる	条件を設定して、もし・ならばのプログラムを動作させる事ができる	ハードルを動かそう ネコとハードルを動かそう ゲームをスタートさせよう スタート背景を作ろう スタート背景を出そう ゲームオーバーの背景を作ろう ゲームオーバーの背景を出そう クリア画面を作成しよう クリア画面を表示しよう 背景を作ろう あたり判定を作る準備 あたり判定を作ろう ぶつかったときに鳴かそう ぶつかったときに色を変えよう 得点に応じてメッセージを表示しよう 得点に応じて音を鳴らそう
3	抽象的思考力	変数を使う事ができる	変数を作って使う事ができる	得点の変数を作ろう 得点を加えていこう 変数を初期化しよう
		プログラムの仕組みを作れる事ができる	プログラムの機能や仕組みを考えて作る事ができる	ゲームオーバーの仕組みを作ろう ハードル越えゲームを作る リセット機能を作ろう ボーナスを追加しよう ボーナスの条件を作ろう ボーナスの動作を作ろう しっばいをゼロにしよう 各変数の値を初期化する ハードルの動きを早くしよう いろいろなハードルの動きを作ろう
		分割されたプログラムを関連させて使うことができる	メッセージを作って動作させる事ができる	メッセージを作ろう メッセージを受け取ったときの処理 ボーナスのメッセージを作ろう ボーナスのメッセージを送ろう

図3 プログラミングスキル評価シート (Scratch 版)

プログラミングスキル評価シートを元に、実際の沖縄のプログラミング講座で教材として用いることを念頭に置き、スキルの自己チェックシートを設計した。(図4) その際に、メンターの意見を入れて、プログラミング教育における言語活動の実践を意識した評価項目を追加した。また、該当するスキル項目について、子どもたちが知識を得た段階なのか、実行する段階までなのかが自己評価できるように、評価基準欄を分割した。

レベル	評価内容	知識	スキル
1	Scratchを使う	基本的な操作を知っている	基本的な操作ができる
	動きブロックを使う	動きブロックのはたらきを知っている	動きブロックを使うことができる
	見た目ブロックを使う	見た目ブロックのはたらきを知っている	見た目ブロックを使うことができる
	音ブロックを使う	音ブロックのはたらきを知っている	音ブロックを使うことができる
	イベントブロックを使う	イベントブロックのはたらきを知っている	イベントブロックを使うことができる
	制御ブロックを使う	制御ブロックのはたらきを知っている	制御ブロックを使うことができる
	調べるブロックを使う	調べるブロックのはたらきを知っている	調べるブロックを使うことができる
	演算ブロックを使う	演算ブロックのはたらきを知っている	演算ブロックを使うことができる
	スプライトを使う	スプライトのはたらきを知っている	スプライトを変更できる
	ステージを使う	ステージのはたらきを知っている	背景を変更できる
2	順次実行を使う	命令を順次実行するプログラムの作り方を知っている	命令を順次実行するプログラムを作ることができる
	繰り返しを使う	命令を繰り返し実行するプログラムの作り方を知っている	命令を繰り返し実行するプログラムを作ることができる
	条件分岐を使う	順次実行から繰り返しを切り出すやり方を知っている	順次実行から繰り返しを切り出すことができる
		条件によって実行する命令が変わるプログラムの作り方を知っている	条件によって実行する命令が変わるプログラムを作ることができる
	不等号を使う	もし・ならばの条件によって実行する命令が変わるプログラムの作り方を知っている	もし・ならばの条件によって実行する命令が変わるプログラムを作ることができる
		不等号を使った条件によって実行する命令が変わるプログラムの作り方を知っている	不等号を使った条件によって実行する命令が変わるプログラムを作ることができる
	論理演算を使う	論理演算を使った条件によって実行する命令が変わるプログラムの作り方を知っている	論理演算を使った条件によって実行する命令が変わるプログラムを作ることができる
	入出力の使い方	指定した入力によりプログラムの動作を変えられる事を知っている	指定した入力により、動作が変わるプログラムを作ることができる
		プログラムによって指定したとおりに出力できる事を知っている	プログラムによって指定したとおりに出力する事ができる
	変数を使う	変数の意味が分かっている	変数を使うことができる
		変数の作り方を知っている	変数を作ることができる
		変数を使ったプログラムの作り方を知っている	変数を使ったプログラムを作ることができる
メッセージを使う	メッセージの意味が分かっている	メッセージを使うことができる	
	新しいメッセージの作り方を知っている	新しいメッセージを作ることができる	
	メッセージを使ったプログラムの作り方を知っている	メッセージを使ったプログラムを作ることができる	
3	プログラムによる表現	やりたいことを図や絵で表現するやり方を知っている	やりたいことを図や絵で表現することができる
		やりたいことを他人に発表するやり方を知っている	やりたいことを他人に発表することができる
		他人の発表を聞いて、やりたい事を理解する方法を知っている	他人の発表を聞いて、やりたい事が理解できる
	プログラムの読解力	プログラムの読み方を知っている	プログラムの読むことができる
		プログラムを読んで、やりたい事を読み取るやり方を知っている	プログラムを読んでやりたい事を読み取ることができる
	プログラムの編集力	他人の作ったプログラムを変更するやり方を知っている	他人の作ったプログラムを変更することができる
		他人のプログラムを、自分のプログラムに反映させるやり方を知っている	他人のプログラムを、自分のプログラムに反映させることができる
	プログラムの仕組みを作る	やりたいことを要素に分解するやり方を知っている	やりたいことを要素に分解する事ができる
		分解したやりたいことの要素を命令の組み合わせで表現するやり方を知っている	分解したやりたいこと要素を命令の組み合わせで表現できる
	プログラムの評価	使用できる命令に合わせて、要素を修正することを知っている	使用できる命令に合わせて、要素を修正することができる
プログラムがやりたいことを実現できているか評価するやり方を知っている		プログラムがやりたいことを実現できているか評価することができる	
		プログラムがやりたいことを実現できていないとき、どうすれば良いか知っている	プログラムがやりたいことを実現できていないとき、どうすれば良いか実行できる

図4 プログラミングスキル自己チェックシート (Scratch用)

この自己チェックシートを、プログラミング講座の受講者配布用にデザインしたものが下記である（図 5、図 6）。
自己チェックシートの目的は、以下のとおりである。

- ① 子どもたちが、自分がどのレベルまでプログラミング能力を身につけているかを自分自身で可視化すること
- ② 子どもたちが、本講座の目指すゴールとしてのプログラミング能力の全体像を把握できること
- ③ 自己の成長を「シールを貼り付ける」という身体行動で確認できること
- ④ 自分が作ってみたいプログラムやその機能・ルールを言語化することで、目的のあるプログラミングを実現すること



（図 5）プログラミングスキル自己チェックシート・表

Lv	ポイント	分かった！	シール	やれた！	シール	
1	Scratchを使う	基本的な操作が分かる		基本的な操作ができた		
	動きブロックを使う	動きブロックのはたらきが分かる		動きブロックが使えた		
	見た目ブロックを使う	見た目ブロックのはたらきが分かる		見た目ブロックが使えた		
	音ブロックを使う	音ブロックのはたらきが分かる		音ブロックが使えた		
	イベントブロックを使う	イベントブロックのはたらきが分かる		イベントブロックが使えた		
	制御ブロックを使う	制御ブロックのはたらきが分かる		制御ブロックが使えた		
	調べるブロックを使う	調べるブロックのはたらきが分かる		調べるブロックが使えた		
	演算ブロックを使う	演算ブロックのはたらきが分かる		演算ブロックが使えた		
	スプライトを使う	スプライトのはたらきが分かる		スプライトを変更できる		
	ステージを使う	ステージのはたらきが分かる		背景を変更できる		
2	順次実行を使う	命令を順番に実行するプログラムのはたらき が分かる		命令を順番に実行するプログラムが作れ た		
	繰り返しを使う	命令を繰り返し実行するプログラムのしくみ が分かる		命令を繰り返し実行するプログラムが作れ た		
	条件分岐を使う	繰り返し部分を切り出すやり方が分かる 条件により実行内容が変わるプログラムのしく みが分かる		繰り返し部分が作れた 条件により実行内容が変わるプログラム が作れた		
	不等号を使う	もし、ならばの条件で実行内容が変わるプロ グラムのしくみが分かる		もし、ならばの条件で実行内容が変わる プログラムが作れた		
	論理演算を使う	論理演算の条件で実行内容が変わるプログラ ムのしくみが分かる		不等号を使った条件で実行内容が変わる プログラムが作れた		
	入出力の使い方	指定した入力により動作が変わるプログラ ムのしくみが分かる		指定した入力での動作が変わるプログラ ムが作れた		
	変数を使う	プログラムで変数を使うやり方が分かる メッセージの意味が分かる		プログラムで指定したとりに出力でき た		
	メッセージを使う	変数のはたらきが分かる 変数の作り方が分かる		変数を使ったプログラムが作れた		
	3	プログラムの表現	新しいメッセージの作り方が分かる プログラムでメッセージの使い方が分かる		メッセージを使ったプログラムが作れた	
		プログラムの読解力	やりたいことを図や絵で表現するやり方が分 かる やりたいことを他人に発表するやり方が分か る 人の発表を聞いて、やりたい事を理解する方 法が分かる		やりたいことを図や絵で表現できた やりたいことを発表することができた 人の発表を聞いてやりたい事が理解でき た	
プログラムの編集力		プログラムの読み方が分かる プログラムの読んで、やりたい事を読み取る 方法が分かる		人のプログラムを読んだ 人のプログラムを自分でやりたい事が分か った		
プログラムの評価		人のプログラムを変更するやり方が分かる 人のプログラムを自分のプログラムに反映す る方法が分かる		人のプログラムを変更できた 人のプログラムを自分のプログラムに反 映できた		
プログラムの仕組みを作る		やりたいことを要素に分ける方法が分かる やりたい要素を命令の組み合わせにする方法 が分かる 使用できる命令に合わせて要素を変更するや り方が分かる		やりたいことを要素に分けた やりたいことを命令で表現できた 命令に合わせて要素を変更できた		
プログラムの評価		プログラムが思いどおり動くか確かめる方法 が分かる プログラムが思いどおり動かないとき、どう 直すか知っている		プログラムが思った通りに動くように直せ た		

（図 6）プログラミングスキル自己チェックシート・裏

このプログラミングスキルの評価の仕組みは、メンターの意見を入れて改良・改善を加えている。

2.モデルの内容

2.1 メンターの募集・研修について

2.1.1 メンター募集期間

2016年7月から9月

2.1.2 メンター募集方法

連携主体である（特非）沖縄県マルチメディア教育研究会を通して、同会の会員を中心に募集を行った。

沖縄県マルチメディア教育研究会は、ICT を活用した教育を推進する沖縄県の小中高の教員を中心に構成された団体である。

2.1.3 メンター募集対象（メンター種別）

募集の際には資格を設けていないが、那覇市を中心とした小中学校の教員を中心に応募が合った。

2.1.4 メンター種別の選択理由

募集前は、メンターに応募される方の属性として(1)大学生や地域の ICT 指導者のように、プログラミングに詳しいが教育経験に乏しい者 (2) 教員など、子どもたちの教育・指導経験は豊富だがプログラミング経験の無い者 の2種類を想定して教材の準備を進めていた。ところが、実際の応募者のほとんどが(2)に該当する方であったため、コーディングスキルを重視するカリキュラムに変更を加えて実施した。

2.1.5 メンター募集に関する工夫

メンター募集に際しては、沖縄県マルチメディア教育研究会の委員の意見を取り入れて、プログラミング教育の実践に役立つ内容となるようにアップデートを行った。沖縄県マルチメディア教育研究会のホームページやメールニュースへの記事掲載により、プログラミング教育に対する問題意識の高い参加者を確保することができた。

2.1.6 メンター研修期間

2016年9月24日、10月19日、12月10日の3回が開催された。

第1回の9月24日の受講者が12月10日ではメンター指導者となるという構成により、効果的にプログラミング教育の実践経験を積むことができるよう配慮した。

2.1.7 メンター研修方法

基本的なカリキュラムとしては、(A)プログラミング教育の特徴の理解とプログラミング講座の運営についての講習（座学）、(B)ワークショップ形式でのプログラミング体験で構成されており、全3回実施された。研修形式としては、指導者2名が担当し、(A)パート1名、(B)パートを2名が指導を行っている。実際には、メンター指導者が担当した回（2回）と、メンター講座を受けた方が指導者を担当した回（1回）がある。

2.1.8 メンター研修に関する工夫

沖縄の場合は、主なメンターが小中学校の教員であるため、どのように子どもたちに指導するかといった教育方法に悩むことはなかったが、そもそもプログラミングをどう扱えばよいかというきっかけをメンターの方に掴んでいただくまでにディスカッション

ョンを行ってカリキュラムに反映させた。エンジニアであるメンター指導者は、プログラムやツールといった知識を理解させようとする押し付け型の教材を作りがちだが、それをメンター研修を体験することによって、子どもたち自身が達成感を得ながら自らプログラミング能力を身に付けていくための教材に軌道修正することができたと考える。

2.1.9 他地域にも再現可能なノウハウ

前述の「プログラミングスキル評価シート」の仕組みは、使用する言語や教材に合わせて修正することによって多様に展開することができる。しかも、低年齢の学習者でも「わかった」「できた」ら 100 均のシールを貼るという行動が、低コストでプログラミングを学ぶ楽しさを増してくれるであろう。

2.2 児童生徒の募集・学習について

2.2.1 児童生徒の募集期間

2016 年 7 月

2.2.2 児童生徒の募集方法

実証校である琉球大学附属小学校の生徒を対象に、校内掲示と教員による案内で募集を行った。

なお、基本は全 4 回とし、連続して参加することが望ましいとして募集した。

2.2.3 児童生徒の対象学年

小学校 4 年生から 6 年生

2.2.4 対象学年の選択理由

プログラミング能力は必ずしも学齢に制限されないが、教材で使用する漢字の表記など生徒の読解力を鑑みて、小学校 4 年から 6 年までとした。

2.2.5 児童生徒募集に関する工夫

教材に使用する「Scratch」という名称を前面に出すのではなく、「ゲームを作ってみよう！」というゴールを示すことで、子どもたちに興味関心を持ってもらうよう配慮した。

2.2.6 児童生徒の学習期間

第 1 回 2016 年 11 月 26 日 初級編

第 2 回 2016 年 12 月 10 日 中級編

第 3 回 2016 年 12 月 23 日 (午前) 上級編

第 4 回 2016 年 12 月 23 日 (午後) 上級編 成果発表

2.2.7 児童生徒の学習内容

プログラミング講座では、内容を 3 レベルに分けている。

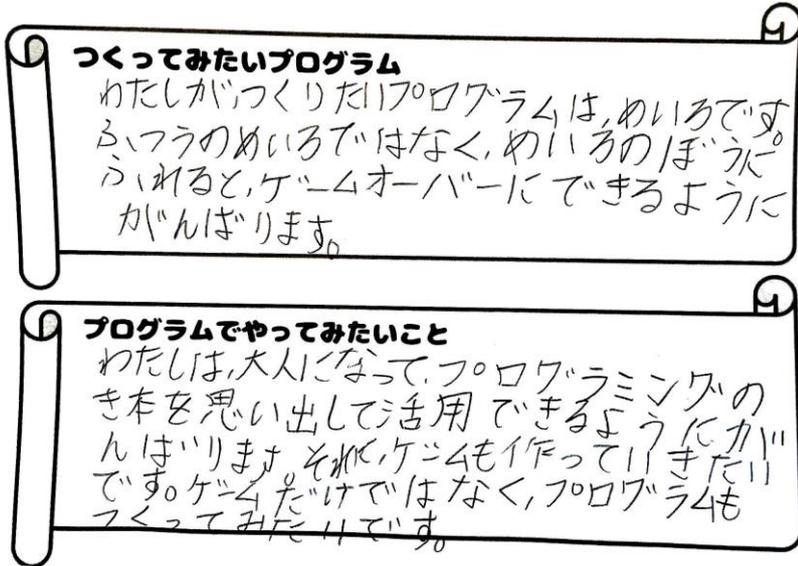
レベル 1 は、初めてコンピュータに接する生徒が、PC や Scratch の基本的操作に慣れるまでをゴールにしている。

レベル 2 は、Scratch の基本的な操作を理解している生徒が、より高度で応用的な表現や論理的な制御ができるよ

うになることをゴールにしている。

レベル3では、Scratch の応用的操作ができる生徒が、プログラムの開発として発展的に行動できることをゴールにしている。実際の講座は、レベルごとに開催するのではなく、1回の講座に各レベルの内容をミックスして実践している。

また、それぞれのレベルでは、自分のプログラムで実現したいことを文章にして、他人に発表し、プログラムを作る途中で他の生徒の話聞いて自分のプログラムの構想に反映させ、完成したプログラムを「工夫したところ」「見てもらいたいところ」を言語化して実演するという学習内容の言語活動を実践させている。(図7)



(図7)

6年女子による自分で作りたいプログラムの記述例

「つくってみたいプログラム」欄は、全般的には「○○のようなゲーム」とう記載が多い中で、彼女は機能的な面を記述している。

専門職としてのプログラマーという存在に興味を持ったのは彼女だけであった。

プログラミング講座の第3回、第4回では、自己チェックシートによって生徒は自己評価を行うが、メンターが指導した学習内容が知識としてうまく伝わったか、それを行動に結び付けられたかが可視化できる。(図8)

事前には「シールを並べたくて、分からない項目やできない項目にまでシールを貼ってしまうのではないか？」という懸念があったが、実際には子どもたちの自己評価はかなりシビアであった。

ポイント	分かった!	シール	やった!	シール
Scratchを使う	基本的な操作が分かる		基本的な操作ができた	
動きブロックを使う	動きブロックのはたらきが分かる		動きブロックが使えた	
見た目ブロックを使う	見た目ブロックのはたらきが分かる		見た目ブロックが使えた	
音ブロックを使う	音ブロックのはたらきが分かる		音ブロックが使えた	
イベントブロックを使う	イベントブロックのはたらきが分かる		イベントブロックが使えた	
制御ブロックを使う	制御ブロックのはたらきが分かる		制御ブロックが使えた	
調べるブロックを使う	調べるブロックのはたらきが分かる		調べるブロックが使えた	
演算ブロックを使う	演算ブロックのはたらきが分かる		演算ブロックが使えた	
スプライトを使う	スプライトのはたらきが分かる		スプライトを変更できる	
ステージを使う	ステージのはたらきが分かる		背景を変更できる	
順次実行を使う	命令を順番に実行するプログラムのはたらきが分かる		命令を順番に実行するプログラムが作れた	
繰り返しを使う	命令を繰り返し実行するプログラムのしくみが分かる		命令を繰り返し実行するプログラムが作れた	
条件分岐を使う	条件により実行内容が変わるプログラムのしくみが分かる		条件により実行内容が変わるプログラムが作れた	
不等号を使う	もし・ならばの条件で実行内容が変わるプログラムのしくみが分かる		もし・ならばの条件で実行内容が変わるプログラムが作れた	
論理演算を使う	もし・ならばの条件で実行内容が変わるプログラムのしくみが分かる		もし・ならばの条件で実行内容が変わるプログラムが作れた	
入出力の使い方	指定した入力により動作が変わるプログラムのしくみが分かる		指定した入力により動作が変わるプログラムが作れた	
変数を使う	プログラムで指定したとおりに出力するしくみが分かる		プログラムで指定したとおりに出力できた	
	変数のはたらきが分かる		変数を使った	
	変数の作り方が分かる		変数を作れた	

(図8)

同6年女子による自己チェックシート使用例

「分かった」欄にシールが無いのに「できた」となっている項目があるが、これはサンプルプログラムを編集していたら実現できたものと思われ、自身だけでは再現できないと想定される。

将来的には、こうしたの自己評価をメンターにフィードバックすることにより、個人の理解度や弱点に合わせた指導計画を立てることも可能であろう。

同時に、プログラミング講座の第3回第4回では、「沖縄の良いところが分かるゲームを作ろう！」をテーマにすることで、学校教育におけるしらべ学習を応用したり、検索エンジンから素材を集め、既存の素材の著作権について体験的に学んだりといった活動も行っている。

2.2.8 児童生徒への講座に関する工夫

メンターがプログラミング講座を実践するに際しては、メンター指導者が制作した教材や指導案を元に内容を自分でやりやすいように修正していただいた。実際のプログラミング講座では、実証校の生徒を異なる学校の教諭が（入れ替わりで）指導することが行われたが、生徒はとくに違和感なく学習することができた。（図9）

プログラミング講座 中級編 きほんのプログラミングを体験してみよう		
■本講座のねらい ・プログラムの「繰り返し」「きっかけ(トリガー)」「条件分岐」を順番に理解し、実際にプログラムをおこなってみることで、基本のプログラミングができるようになる。 プログラミングツール(スクラッチ)の「ずっと」「××がクリックされたら」「もし××なら」を使うことができる。 ■理解度の確認と評価テストについて ・ワークショップ実施前と実施後にワークシートに記載させることで判定を行う 事前と事後で行うことで態度変容などの判定を行う ■本時の学習		
学習内容	指導上の留意点	
1)はじめに(5分) ・プログラムに関するアンケートを実施	中級編の内容について、どのような理解レベルかをアンケートで確認する。	ワークシートへのアンケート記載で事前と事後の態度変容などを判定する。
2)スクラッチによるプログラミング ・「ずっと」を使ってみる(10分)	実際に、資料のとおりスクラッチを使わせて基本の動作を理解させる。 「ずっと」ブロックが使える	最後に「ワークシート2」に記載してもらおうが、画面を見て動作して入ればよい。
・「みどりのはたがクリックされたら」を使ってみる(10分)	資料に沿って、スクラッチを使わせて「みどりのはた」ブロックと「もし××だったら」ブロックを使わせてみる「みどりのはたがクリックされたら」	
ワークシート1(20分) ・「もし××だったら」を使ってみる(15分)	「もし××だったら」のブロックが使える	
・「へんすう」を使ってみる ・「メッセージ」を使ってみる(20分)	基本的な使い方を覚えて、自分でいろいろ試せるようになる	
3. さいごに(15分) ・アンケート2に、学習した内容を記載してもらう。		

(図9) 作成した指導案

実際のプログラミング講座ではこのまま指導を行うのではなく、担当するメンターがこれを参考にして、ニュースの話題やロボットなどの教材を組み合わせるように改変して使用している

2.2.9 他地域にも再現可能なノウハウ

自己チェックシートなど各種のツールは、場所を選ばず低コストで再現できると同時に、地域特性を活かして改変し、生徒の成長に合わせて改良することが容易になるようデザインしてある。とくに自己チェックシートを地域の特徴を活かしたデザインにすることで、生徒のプログラミング講座に対する参加意識を高めることにもつながるであろう。

また、これらのツールが整備されたことにより、児童生徒への教育体験のないメンターであっても、案を参考にして（あるいはそのまま利用して）自分の指導案を作成することができるため、プログラミングは分かるが教え方が分からないといったエンジニアがメンターとして活躍することが容易になることが期待できる。

3. モデルの訴求ポイント

3.1 モデルのねらい・意義

沖縄「じんぶなー」モデルは、特別な教材・教具を使用すること無く、地域のメンターが自立自走して、今後も自力でプログラミング教育を継続して実践できる環境を構築することをゴールとしている。

沖縄のメンターは、現職の小中学校の教諭がメインであるが、この講座を受講するまではプログラミングを行ったことがないという方がほとんどであった。そこで本事業のメンター講座をデザインするにあたり、まずはメンターの先生方に「プログラミングを好きになってもらうこと」「プログラミングの楽しさを体験してもらうこと」を目標に置いた。

従来のプログラム言語の教育では、とすれば見本のプログラムを再入力し、文法エラーを起こすと減点といった評価がなされることもあったが、沖縄「じんぶなー」モデルでは、「プログラミングができるってカッコイイよね！」というイメージを重視して、プログラミング「できるようになったことを評価する」仕組みを構築しようと考えた。そこでデザインされたものが、プログラミング能力を要素に分解したスキルマップと、自己チェックシートなどのツールである。

このようにプログラミング能力が客観的に可視化されたことによって、評価を共有したり個別の指導内容のデザインが容易になるなど、メンター活動を支援するツールを得ることができたと考える。

3.2 モデル実施により得られた成果

3.2.1 受講した児童生徒の変化

最大の変化は、ユーザーとして既存のゲームソフトをプレイしていただけの生徒が、「もし自分だったらこうしたい」と、自分なりの機能やルールを考え始めたことである。

プログラムにもそれを作っている人がいる、という気付きを得たことから、社会におけるプログラムの役割や、プログラマーという専門職へのキャリアデザインに言及する生徒が出るなど、単にプログラミング言語のコーディングスキル育成にとどまらない効果を得られた。

3.2.2 担当したメンターの変化

メンター講座開講前は、一部の受講者にプログラミング言語の習得を目的とする内容との誤解があるようだったが、「プログラミングができるようになる楽しさ・実感」を体験してもらうことがねらいであり、その達成感を生徒にも伝えたい、というゴールを理解していただいてからは、メンターからも積極的にツールの改良意見を得られるようになった。

実際のプログラミング講座でも、用意した指導案にメンターが独自の拡張を加えて子どもたちの興味関心を引きつけることができたなど、メンターが現職の教員であるメリットを十分に発揮することができた。

3.2.3 保護者の反応

プログラミング講座を受講した生徒が、自宅でも継続してプログラミングをしたいとの要望があったため、家庭のPCやタブレット端末でも実行する方法を指導している。また、保護者自身がプログラミング講座を受講したいとの声も上がっているため、メンターが自主的に対応することを検討している。

3.2.4 教員、教育委員会の反応

公開で実施したメンター講座を受講した他県の高校教員から、本事業の内容を高校生向けにアレンジできないかとの要望があった。内部的に中学生向け教材案は作成しているが、高校生向けとなると表現などに変更が必須となるため、今後の課題としている。

3.2.5 協力大学、団体等の反応

学習ソフトウェア情報研究センターは、本事業で開発したスキルマップ、カリキュラムと教材を元に、中学生向けのScratchプログラミング体験講座を開発し、埼玉県の中学生を対象として3回開催している。

また、ロボットを使用して小学生にプログラミング教育を実践している団体にスキルマップとチェックシートを提供し試用・評価していただいたが、非常に好評であり自団体の活動に使用したいとの要望を受けている。

4.モデルの改善点

4.1 実施にあたって直面した困難

プログラミング教育は、高価なハードウェアを教材に使用することが多いが、そうした教材に依存したプログラミング教育のカリキュラムは、バージョンアップなど環境の変化によって陳腐化してしまうリスクを負いかねない。

沖縄モデルでは、なるべくハードやソフトに依存しないカリキュラムをデザインすることを目指しているが、それでも使用するソフト（Scratch）にメジャーアップデートが行われてしまうと、教材とのミスマッチが生じる可能性がある。とくに、プログラミングに不慣れなメンターに配慮して、教材コンテンツにツールの詳しい操作や説明を加えると陳腐化のリスクが高くなる。

カリキュラムや教材の寿命を考えて教材コンテンツの表現を抽象的なレベルに留めるか、メンターのスキルを考慮して具体的な表現にするか、カリキュラムデザインをするにあたっては、このバランスをどう見切るかが難しいところである。

4.2 実施を通して把握した反省点

メンター指導者が現役のソフトウェア事業者であるため、当初想定していたメンターのスキルと実際とのギャップが大きかった。メンター指導者にとっては「当たり前」の知識が、メンターにとっては「専門的」と評価されることもあった。そこで、まずはメンター指導者とメンターの間で、育成すべきプログラミング能力についての合意形成を持ち、それに合わせてカリキュラムを修正する必要があった。

そのため、沖縄モデルを他地域に展開する場合でも、事前に地元メンターとのプログラミングスキルを合意形成し、ゴールとする地域ならではの「じんぶな」像の設計をしておくことが重要であると考えた。

4.3 モデル普及に向けた改善案

本年度は成果として、メンター講座教材、プログラミング講座教材、プログラミングスキルマップ、児童生徒による自己チェックシートを開発した。このうち、自己チェックシートによりプログラミング講座を受講した児童生徒のスキルを可視化することが可能になったが、これを指導するメンターのスキルは可視化されていない。また、自己チェックシートの記載内容を活用する仕組みまではデザインしていない。

そこで、沖縄モデル普及に向けては、以下の改善による深化を検討している。（図10）

(1) メンターの自己チェックシートの開発

プログラミングスキルマップを元にして、メンターもセルフチェックが行える（教えることができたか）ツールを開発する。

メンターと児童生徒を同じ項目で評価できることにより、指導上の問題点が可視化できる。

(2) プログラミング学習カルテの開発

児童生徒の自己チェックシートに記載された内容を元にして、個人ごとの得意・不得意や作りたいプログラムなどの情報をまとめたプログラミング学習カルテを開発する。これによって、メンターが変わっても指導内容を継続したり、得意分野を伸ばすような指導が容易になる

(3) 教材、指導案の見直しと改訂

実際のプログラミング講座の実施結果と受講者の自己チェックシートの評価を反映させて、講座で扱っているものの理解度が低い項目や、実行しきれていない項目についての教材コンテンツの内容の見直しを行う。

同時に、メンターの要望でスキルマップが変更になった分、指導案など周辺ツールが対応しきれていない部分があるので、これを見直して改訂する

(4) 定量的評価の試行

自己チェックシートにより、定性的にプログラミングスキルを評価する素地が構築されたので、これを元に定量的な評価について検討する。

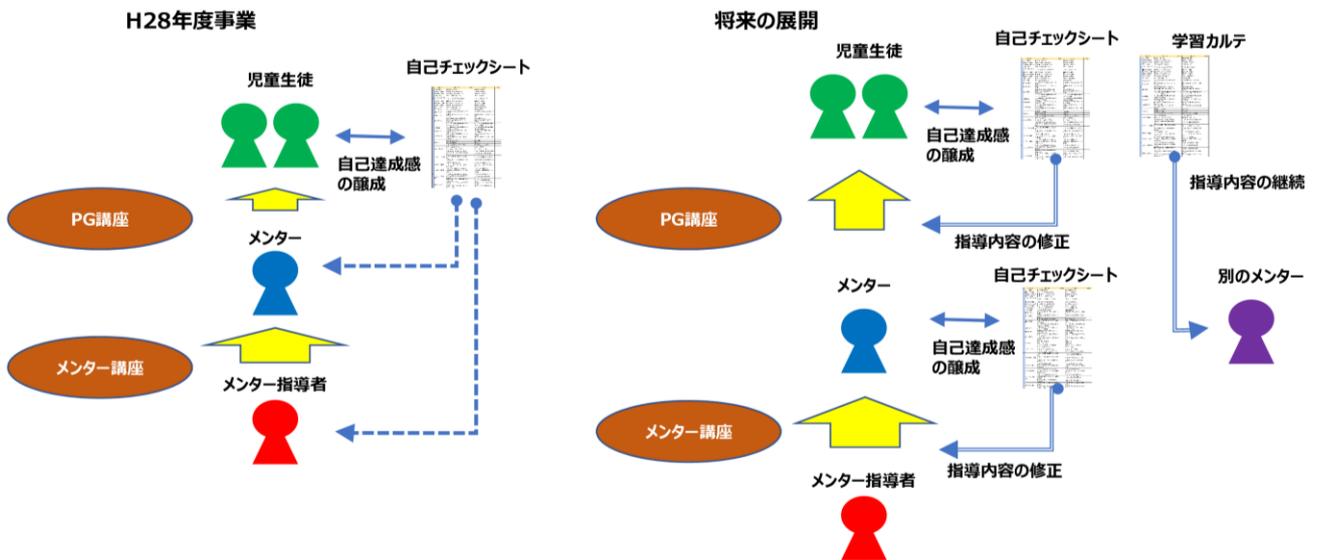


図 1 0 沖縄モデルの改善案

5.モデルの将来計画

沖縄ブロックでは、地域の自立自走によるプログラミング教育の継続実施に向けて、本年度の成果を活かした活動を続けていく計画である。実際には、H29年度の異動によりメンターの学校が変わることが見込まれるため、新任メンターの育成に向けた活動を核に活動を拡大するとともに、教育の質の向上を図る垂直展開を行う。

また水平展開として、他地域との展開で2つの方向性を検討している。

(1) 沖縄の離島でのプログラミング教育の実施

沖縄ブロックでの課題は、離島におけるプログラミング教育の実践である。

評価ツール類の展開は容易であるものの、ネットワーク環境が弱い、機材が不十分などの課題があるため、それを解消するための工夫が必要とされる

(2) 他地域での実践と評価

沖縄モデルを移植して他地域でプログラミング講座を実践する。

これによって、地域特性に依存しないプログラミング教育のモデルを構築する。

なお沖縄モデルでは、プログラミング全体スキルマップで整理したスキル領域の中から、とくに「コーディング・プログラミング能力」「表現力」「意欲の向上」に力点を置いてカリキュラムを設計しているが、現実のプログラミングスキルとして求められるスキル領域は、より幅広いものである。

そこで他地域の展開として、沖縄モデルのスキルマップとスキル自己チェックシートなど仕組みを共通に利用しながらも、異なるスキル領域を取り上げたプログラミング教育のモデルが登場することが望ましいと考える。