



教えないアルゴリズム プログラミング教育の実践

情報I が迫りくる

プログラミングやらなっちゃ



普通の高校生に、どうやって
配列やソートを教えるの???

パワポ100%+ Amazon Poly 音声合成



関東第一高等学校
千葉県立生浜高等学校
情報科非常勤講師 太田 剛



説明内容

- 1 はじめに:目的
- 2 プログラミング教育の難しさ
- 3 問題点への対応
- 4 2020年度用に改良した教材
- 5 想定授業方法
- 6 今後の課題



1

はじめに: 目的





1

はじめに: 目的

プログラミングが得意でない先生でも授業ができる。

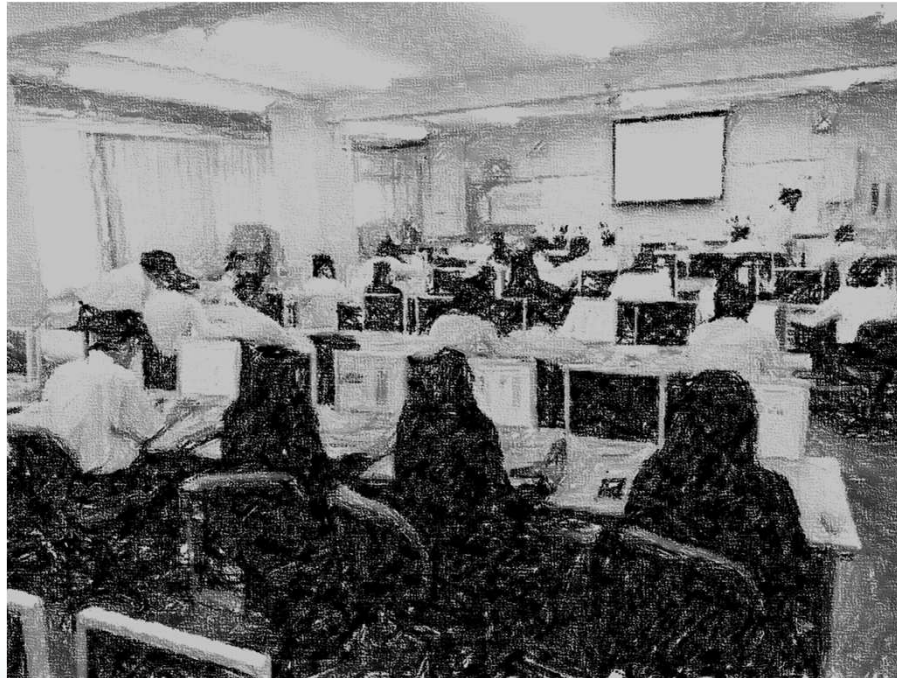
普通の学校の生徒で、プログラミング経験が無くても検索ぐらいのアルゴリズムのプログラムを作成できる。

生徒が主体的、対話的に楽しんでプログラミングを学習する。



2

プログラミング教育の難しさ



現状の高校でのプログラミング教育の実践



Python/JavaScript
等で分岐や繰り返し
を学習

アンプラグドで
並び替えなどの
アルゴリズムを学習

データサイエンスや
Webアプリの
実用プログラム開発

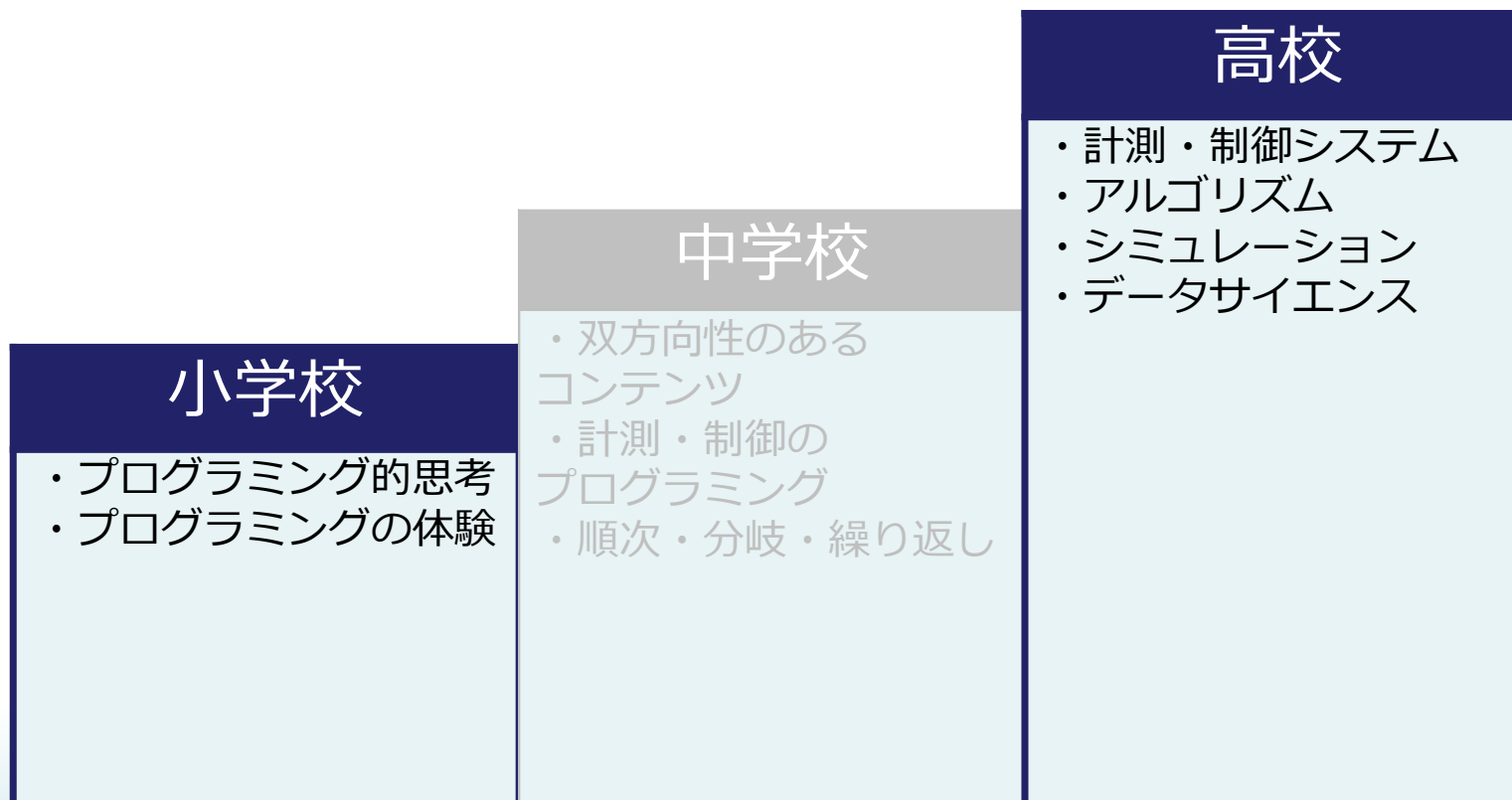
ロボットや
フィジカルな装置を
制御・計測

根本的に、検索や並び替え等のアルゴリズムをプログラミング
しているという実践はあまりないみたい





高校プログラミングの問題1: 小中高の連携の欠如



高校プログラミングの問題2: 論理的な難しさ

情報関連の学部・学科においてアルゴリズムのプログラミング教育は長年行われてきました。

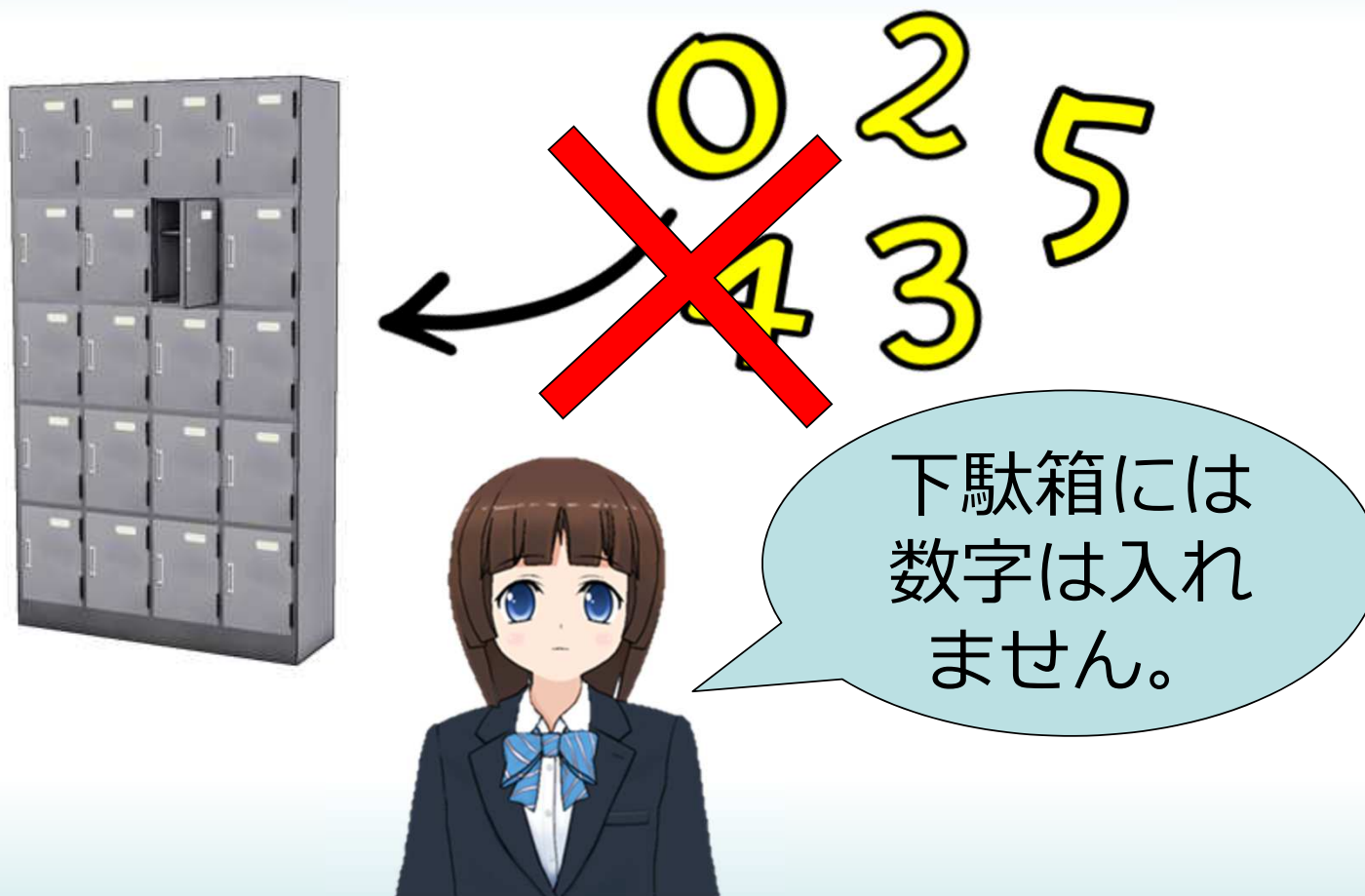
その経験から配列・添え字による配列の操作・二重ループ等が学生にとって難しいものと認識されています。

```
sort(arr):  
    change = False  
    while change:  
        change = False  
        for i in range(len(arr) - 1):  
            if arr[i] > arr[i + 1]:  
                arr[i], arr[i + 1] = arr[i + 1], arr[i]  
                change = True
```

配列操作
二重ループ

?????

根本的な問題: プログラミングの認知的ギャップ



3

問題点の対応



1 2018年度の実践

- 変数の可視化やトレースなど考慮
- 個人ペースの協働学習
- アルゴリズムを中心に
- Scratchを使用

2 2019年度の実践

- プログラミングの認知過程に注目
- タンジブルな変数の利用
- 適切なステップ
- フローチャートとプログラムの対応

3 2020年度の実践準備

- 簡易的なタンジブル変数
- 初歩内容の充実
- より細かな指示の追加
- 動画解説の追加



対応1: チェックシートにを使って個人ペースの学習



	No	内容	スライド No	チェック			
				[動画]	[理解]	[打ち込み]	[開発]
ここまで、やってみよう。	1	学習の進め方	2		<input type="checkbox"/>		
	2	準備運動1: Scratch の四則演算	4		<input type="checkbox"/>	<input type="checkbox"/>	
	3	準備運動2: ネゴに自分の名前を言わせる	5		<input type="checkbox"/>	<input type="checkbox"/>	
	4	プログラムと変数	6	<input type="checkbox"/>			
	5	課題1: 入力した2個の数で四則演算	7				<input type="checkbox"/>
	6	打ち込み1: 合格判断	8		<input type="checkbox"/>	<input type="checkbox"/>	
	7	課題2: 合格不合格判断	9				<input type="checkbox"/>
	8	課題3: 3つの数の合計	10				<input type="checkbox"/>
	9	課題4: 正三角形の判断	11				<input type="checkbox"/>
	10	打ち込み2: 1から10までの数を言う	12		<input type="checkbox"/>	<input type="checkbox"/>	
	11	プログラムに名前をつけて保存	13		<input type="checkbox"/>		
	12	打ち込み3: 1から10までの合計を言う	14		<input type="checkbox"/>	<input type="checkbox"/>	
	13	プログラムとフローチャート	15	<input type="checkbox"/>			
	14	変数/リストシートを使って考えよう	16		<input type="checkbox"/>		
	15	課題5: 2からXまでの偶数の合計	17				<input type="checkbox"/>
	16	たくさんの数の合計の説明	19		<input type="checkbox"/>		
	17	打ち込み4: リストを使った3つの数の合計	20		<input type="checkbox"/>	<input type="checkbox"/>	
	18	打ち込み5: リストを使った5つの数の合計	22		<input type="checkbox"/>	<input type="checkbox"/>	
	19	リストと繰り返し	24	<input type="checkbox"/>			
	20	最後目標の課題の説明: 数の並び替え	25		<input type="checkbox"/>		
	21	(休憩: おみくじ)	26		<input type="checkbox"/>	<input type="checkbox"/>	
	22	リストに数をセットする方法	28		<input type="checkbox"/>		
	23	課題6: リストの中から数を探す	30				<input type="checkbox"/>
ここまで、できたらすごい。	24	課題7: リストの中の一番小さい数を見つける	33				<input type="checkbox"/>
	25	課題8: 一番小さい数を配列の先頭に入れ替える	35				<input type="checkbox"/>
	26	打ち込み6: 二重繰り返しに挑戦	39		<input type="checkbox"/>	<input type="checkbox"/>	
もっと、やろう	27	課題9: 数の並び替え	43				<input type="checkbox"/>
	28	発展課題1: 他の並び替えの方法	46				<input type="checkbox"/>
	29	発展課題2: 並び替えの方法の処理速度の違い	47				<input type="checkbox"/>



アルゴリズムとプログラム実習シート

○変数

A B X Y

I J T S

○リスト(配列)

D[]

D[1] D[2] D[3] D[4] D[5]

- 教材
- Web上の指示書
 - 課題を示したチェックリスト
 - 変数リストシート



対応2: ビジュアル言語のScratchの利用

- ・ ビジュアル言語なので使い易い
- ・ 変数の内容を見る(可視化)ことができる
(大学等の教授法として可視化は有効)

おすすめ品!

でも、Scratchなんか言語じゃないという人も

学習指導要領解説での言語要件

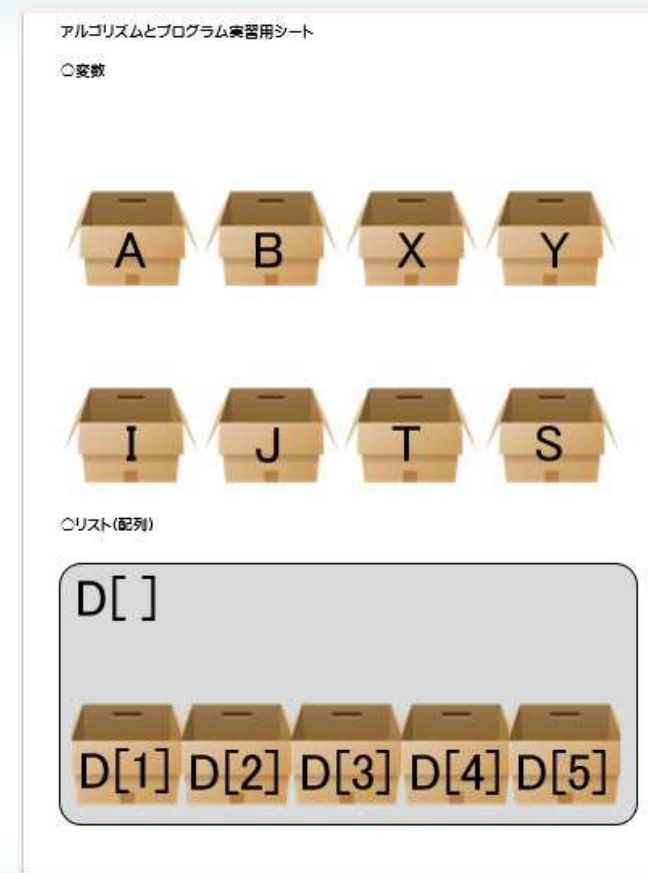
アルゴリズムをプログラムとして表現	○
ライブラリや APIを利用	○
プログラムを分割して構造化/関数作成	△
ツールやアプリケーションを開発	△
カメラやセンサ及びアクチュエータを利用	○
画像認識や音声認識及び人工知能等利用	○

対応3: タンジブル(手で操作できる)な変数環境の利用



2019年度

アンプラクドと
プログラミングの連携

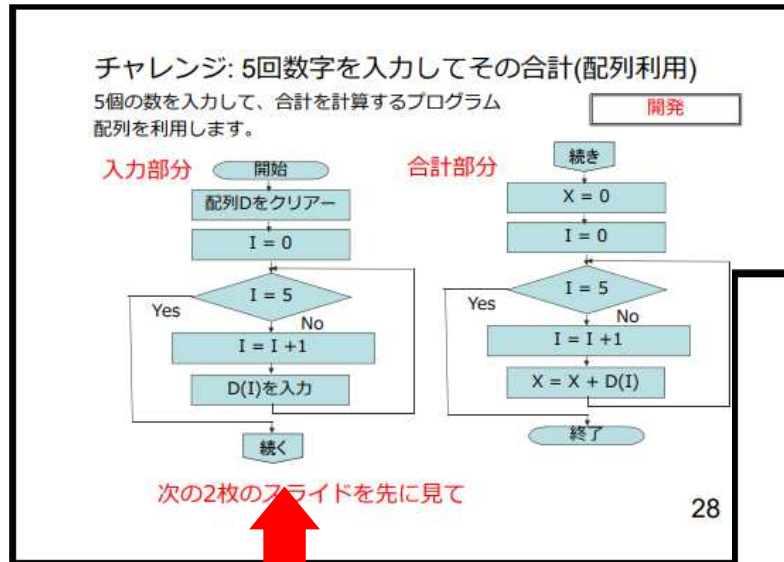


2020年度



対応4: フローチャートをヒントとして活用

タンジブルな変数、フローチャート, プログラムの 対応の明確化



プログラムと タンジブルな変数の対応

チャレンジ: 5回数字を入力してその合計(配列利用)
入力部分(スライド28の左側)のプログラムで

開発

カチカチやりながら、
D[I] を5回入力してい
るイメージです。

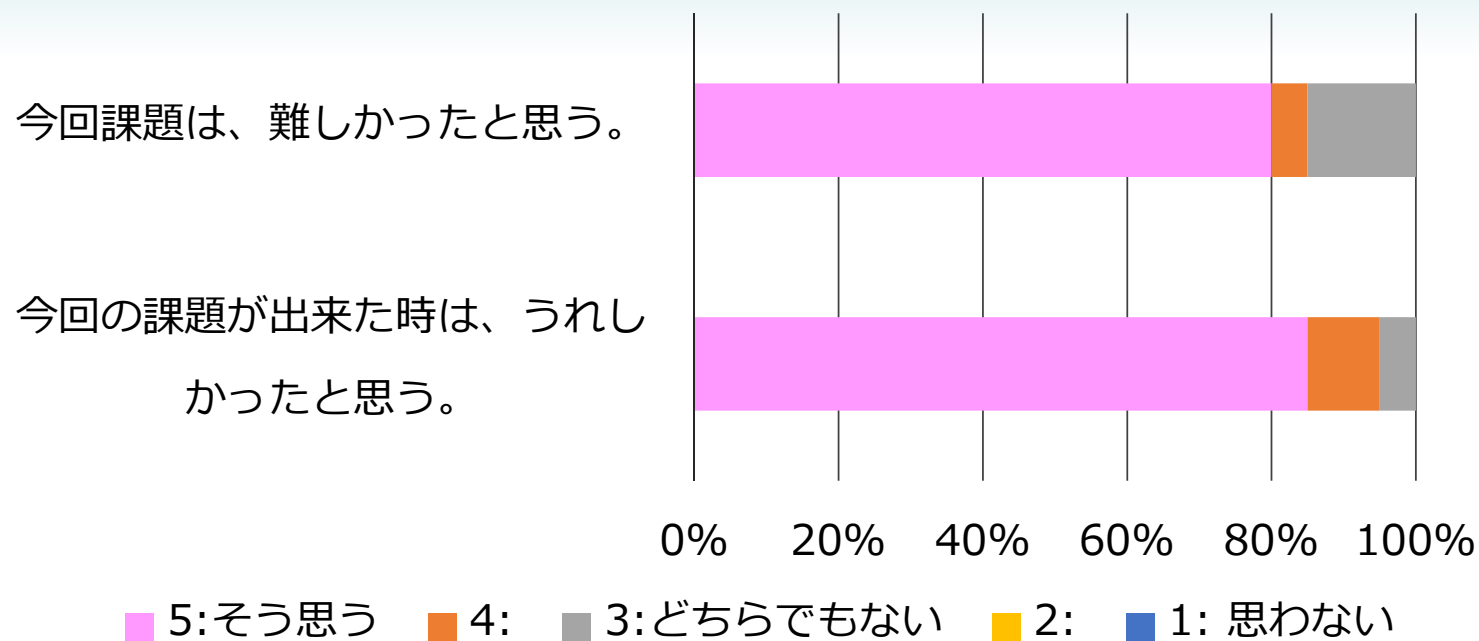
合計のイメージの
次のスライドも見て

この下に、スライド28の右側
のプログラムを追加して

29

フローチャートと プログラムの対応

対応5: 考えるスモールステップ

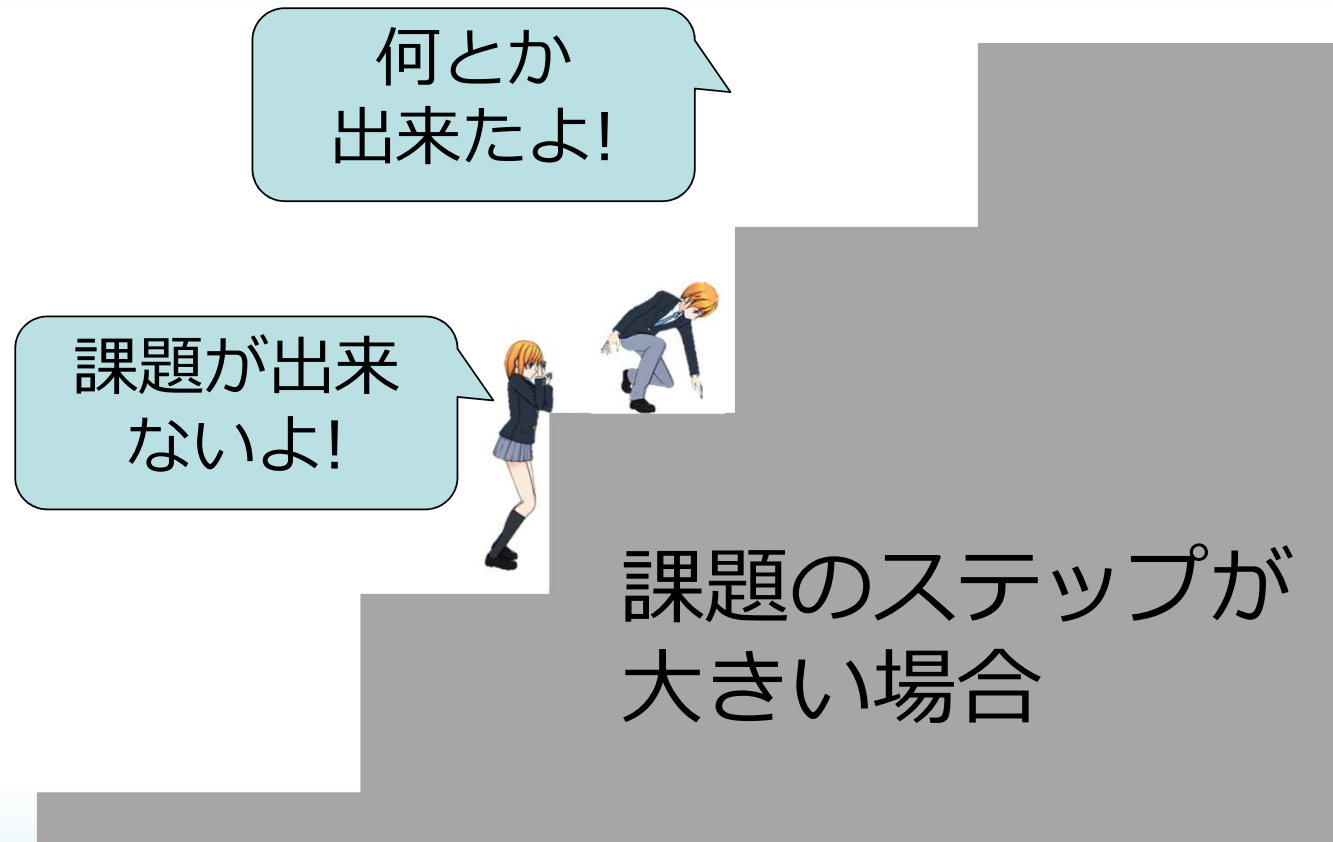


できると、うれしい

わかると、楽しい



考えるスモールステップ



考えるスモールステップ

指示の通りやったら出来たけど、なんで出来たかわかりません



課題のステップが小さい場合



考えるスモールステップ

難しかった
けど出来た!

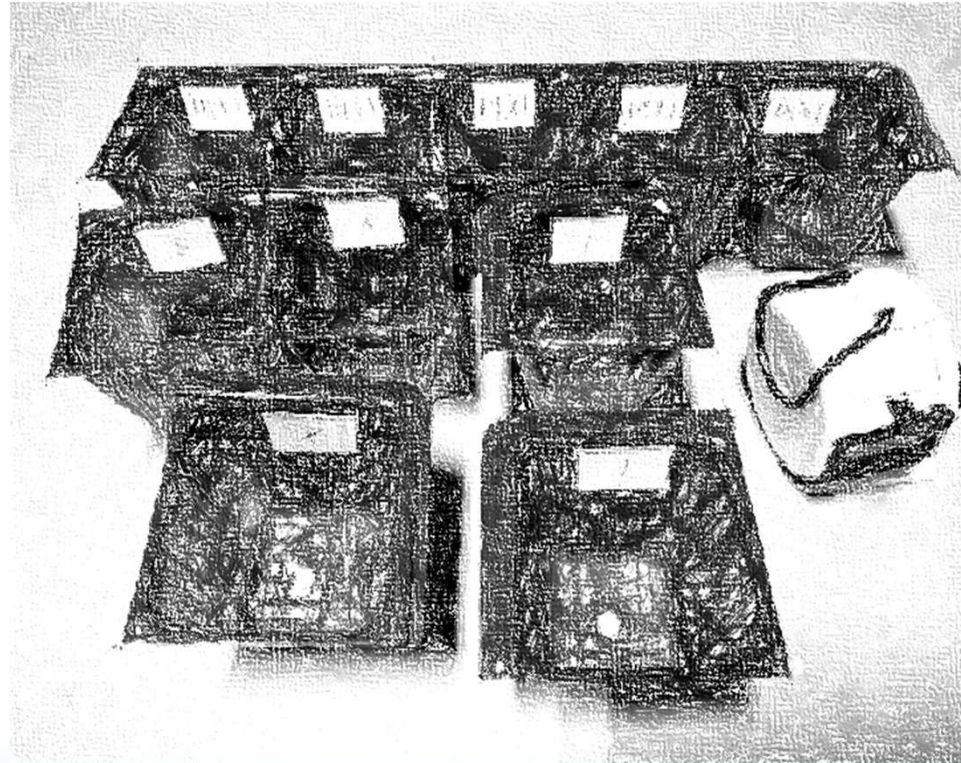


課題のステップが
うまく設定された
場合



4

2020年度用に改良した教材



基礎的なプログラミング方法の充実

準備運動2: ネコに自分の名前を言わせる 理解・打込み



自分の名前をキーボードから入力して、ネコに言わせてみます。

ヒント: Aという名前の変数を作って使います。



こんな簡単なプログラムは打ち込むのは面倒という人は飛ばして進んでいいです。

5

四則演算やデータ入力に関する説明や課題を充実させる



基礎からアルゴリズムの比較までの幅広い内容



	No	内容
こまでは、 やってみ よう。	1	アルゴリズムとプログラム:学習の進め方
	2	準備運動1: Scratchの四則演算
	3	準備運動2: ネコに自分の名前を言わせる
	5	課題1: 入力した2個の数で四則演算
	7	課題2: 合格不合格判断
	8	課題3: 3つの数の合計
	9	課題4: 正三角形の判断
	15	課題5: 2からXまでの偶数の合計
	23	課題6: リストの中から数を探す
	ここまで、 できたら すごい。	24
25		課題8: 一番小さい数を配列の先頭に入れ替える
27		課題9: 数の並び替え
もっと、 やろう	28	発展課題1: 他の並び替えの方法
	29	発展課題2: 並び替えの方法の処理速度の違い

◀ 想定6時間の授業
の目標として検索
プログラムの作成

できる生徒は、異
なるアルゴリズム
での処理時間の比
較まで

ヒントが少なく考える課題

課題4: 正三角形の判断

開発

A, S, Xの3個の三辺の値を入力して、すべての値が同じ場合に、「正三角形」、そうでない場合は「正三角形じゃない」と表示するプログラムを作ってみよう。



ヒント: プログラムを動かしたあと、次のような数を入れて正しく動作するか確認してみよう。

数の組み合わせ	A	S	X	ネコ
すべてが同じ	10	10	10	正三角形
全部違う	10	20	30	正三角形じゃない
AとSだけ同じ	10	10	20	正三角形じゃない
SとXだけ同じ	20	10	10	正三角形じゃない

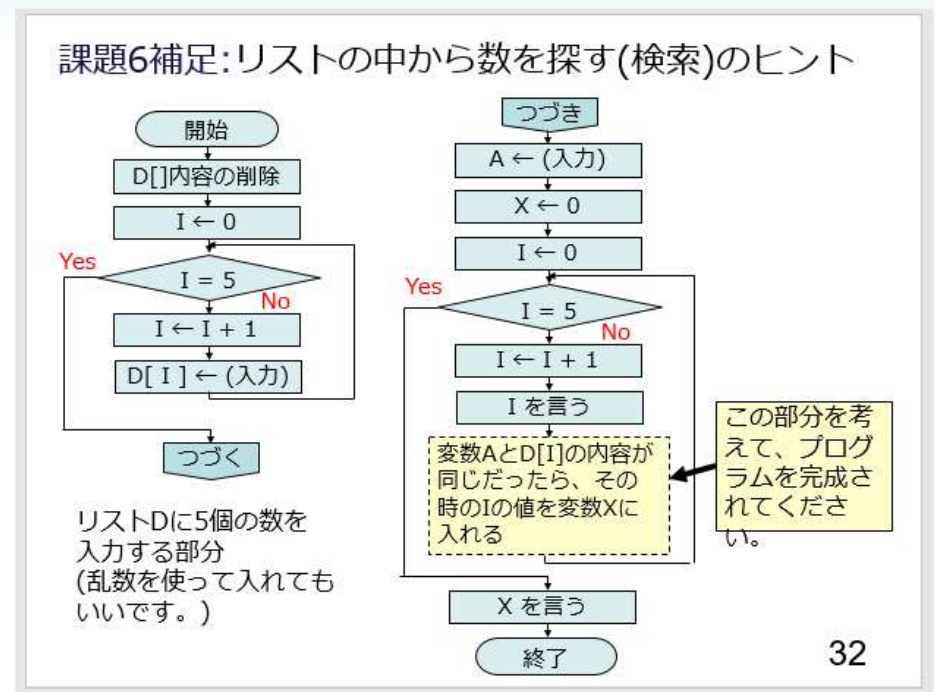
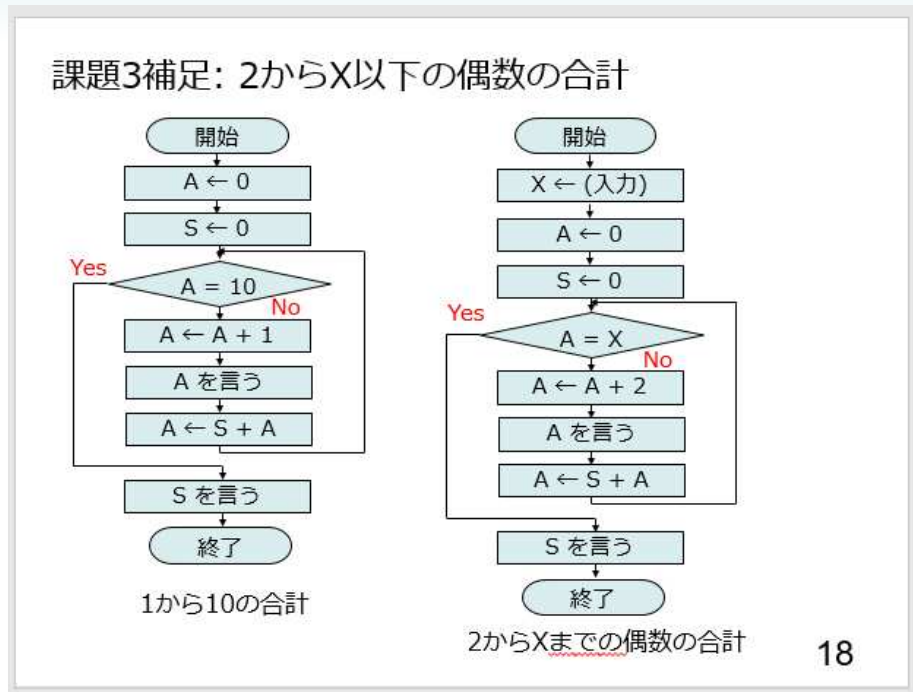
11

プログラムを試行錯誤しながらつくる場面

友達に聞いたり、相談する場面



フローチャートとプログラムの対応



初めの方の課題は、全部のフローチャートをヒントとして提示

後の方の課題では、重要な部分は空白のフローチャートをヒントとして提示

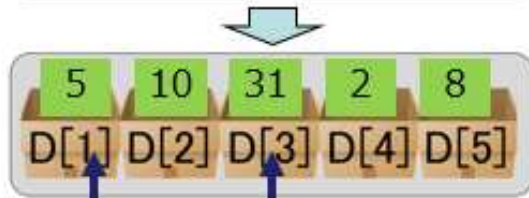
タンジブルな変数での説明

課題8補足: 一番小さい数を配列の先頭に入れ替える

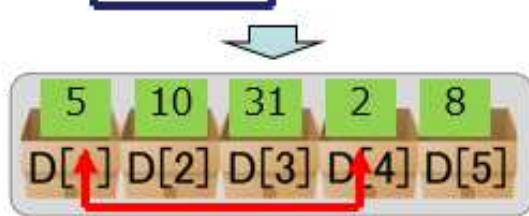
ヒント: プログラムの考え方(実際に紙に数値を書いてやってみよう)



1番目と2番目を比較して5の方が小さいので入れ替え



1番目と3番目を比較して5の方が小さいので入れ替えない



1番目と4番目を比較して2の方が小さいので入れ替え



1番目と5番目を比較して2の方が小さいので入れ替えない

36

アルゴリズムとプログラム実習用シート

○変数



○リスト(配列)



変数シートに対応した説明

リストとその操作の充実した内容



No	内容
16	たくさんの数の合計の説明
17	打ち込み4: リストを使った3つの数の合計
18	打ち込み5: リストを使った5つの数の合計
19	リストと繰り返し
20	最後目標の課題の説明: 数の並び替え
21	(休憩: おみくじ)
22	リストに数を設定する方法
23	課題6: リストの中から数を探す

リストと繰り返し

動画

リストや繰り返しを使えばプログラムの動きも分りやすくなります。ここでは動画でその動きを説明しています。

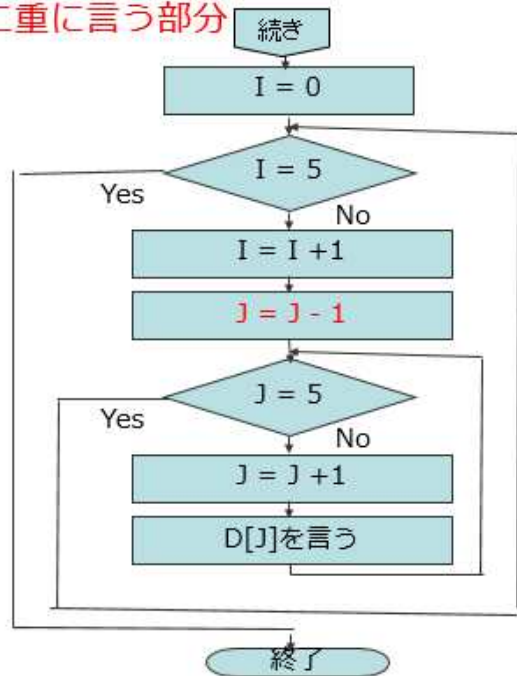
24

リストと、その操作について動画で説明

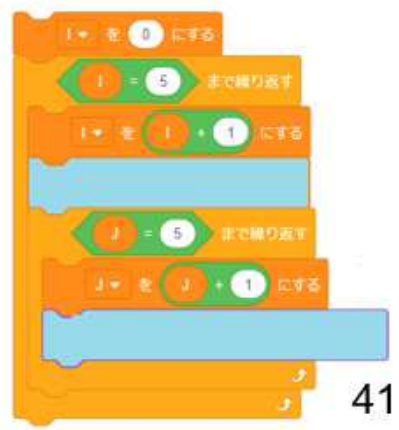
段階的な並び替えのプログラムの開発



二重に言う部分



フローチャートはごちゃごちゃしていますが、プログラムで見ると、Iを使った大きな繰り返しの中にJを使った繰り返しが入っている形になります。



No	内容
23	課題6: リストの中から数を探す
24	課題7: リストの中の一番小さい数を見つける
25	課題8: 一番小さい数を配列の先頭に入れ替える
26	打ち込み6: 二重繰り返しに挑戦
27	課題9: 数の並び替え

並び替えを構成する
プログラム要素を順
番に学習

5

想定授業方法



学習時間と他のScratchを使用した単元との関係



キャラクター(ゆるキャラ)作り

オリジナルでキャラクターと背景を作成する。
人気投票 (2回目の授業の最後)

2

ゆるキャラ作り
(2時限)
・ IDパスワード
・ デジタルデータ



Scratchで面白プログラミング (1)

- ・ Scratchの実習時間の進め方
- ・ タイプ別開発方法
(・パスワードについて)

補足:

1

面白プログラミング
(2~3時限)
・ 情報デザイン
・ プログラミング入門



情報の授業

人が喜ぶスマホアプリを開発しよう

- ・ 開発の進め方
- ・ アイディアの出し方

2

スマホアプリ開発
(6~8時限)
・ 情報デザイン
・ ペーパー
プロトタイピング



おしえないアルゴリズムプログラミング教育(6時限)

生徒は主体的で対話的な学習



二つの協働学習



例：陸上部
基本、個人競技
授業はこっちでいきます。



例：サッカー部
チーム競技

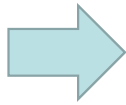
- ・授業中歩き回っていいです。
- ・分からないところがあったら、出来ている友達見つけて聞いてみてください。
- ・困っている友達がいたら助けてあげてください

・好きな席に座っていいです。
-> ペアプログラミング的に二人で協働して課題を解くことが自然発生。

ファシリテータとしての教師の役割



従来の
一斉教授型



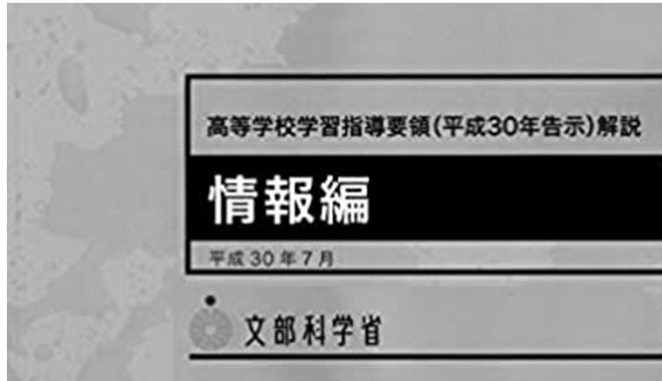
- ・ 支援の必要な生徒に対応する
- ・ 教えすぎないぐらいが丁度いいかも。

支援で一番難しいのは、論理的に十分把握できない生徒を見つけ出し、タンジブルな変数などを使って教えたり、考えることを勧める。

プログラミングの経験が少ない先生は、1年目は生徒のいっしょに勉強するぐらいの気持ちがいいかもしれません。

6

今後の課題



- 2020年度教材の授業での使用
 - 生徒の到達度と難易度の確認
 - フローチャートの有効性の確認
 - タンジブルな変数の有効性の確認
- 2020年度実践結果による、さらなる教材の改良。

補足情報



2019年度に使用した教材等は下記のサイトからダウンロード
できます。

なお、教材は暫定版公開中で、動画の部分は2020年8月中に
作成予定です。

高校「情報科」の教材・指導案作ってみました。

<http://beyondbb.jp/>

高校 情報科 教材	検索
-----------	----

Unit05 アルゴリズムとプログラム