全高情研16

Python & DNCLで書かれた典型的なアルゴリズムの解読と実装

愛知県立一宮高等学校

鈴木 淳子

目次

1．学校紹介＆自己紹介

勤 務 校 紹 介

愛知県立一宮高等学校

普通科
８クラス×３学年

ファッション創造科
１クラス×３学年

創立１０５年

SSH

データサイエンス教育

R5年度の重点目標の１つ

自 己 紹 介 ～ オモテVer. ～

鈴木 淳子

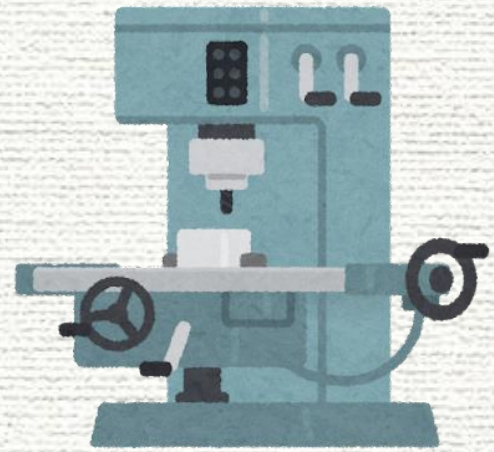
触った言語

大学で物理専攻（宇宙線）

Basic（お遊び）

↓

電機メーカーでソフトウェア開発（CADCAM）



C言語

↓

個人事業主（高速道路掲示板システム

↓ ・出退勤管理システム等）

情報の先生に！

C＃

自 己 紹 介～ ウラVer. ～

情報教員になったきっかけ＝趣味のスマホアプリ作り

android/iOS/windowsPhone(故人)　java・objective-C

体重管理アプリ・時間割アプリ・仏語学習アプリ等

Google本社 @San Francisco

自 己 紹 介～ ウラVer. ～

アプリを作る活動を通して
感じたこと

アウトプットすることの楽しさ

広がる世界

良いアウトプットにするには幅広く正しい知識が必要

情報教育にかかわりたい

（本校における）

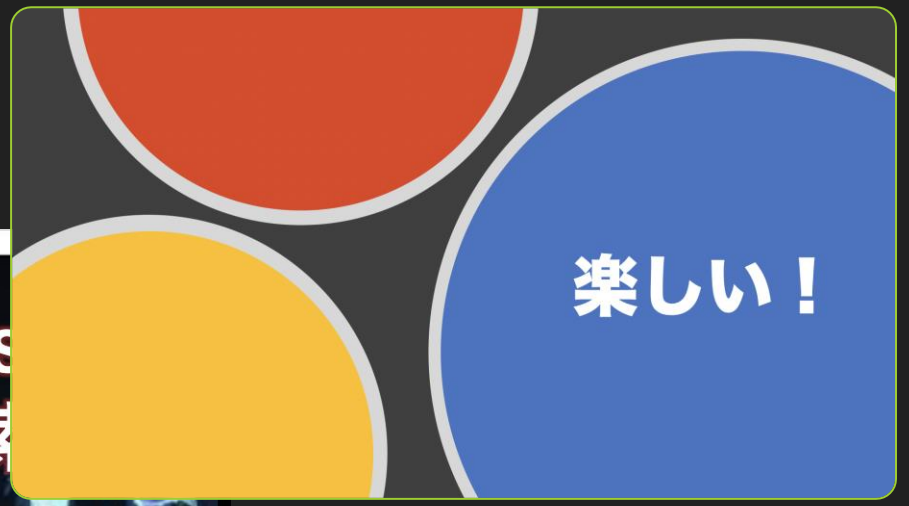
2. プログラミング教育の方向性

p5.js

p5.js web editor

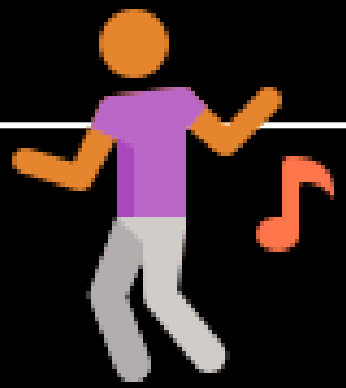
で、クリエイティブコーディングを！

Suzuki Junko



「楽しい＝正義」だったけど・・・？

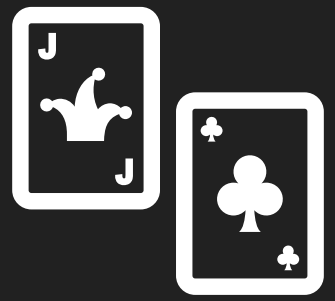
楽しいプログラミングで
自発的な学びを促す？

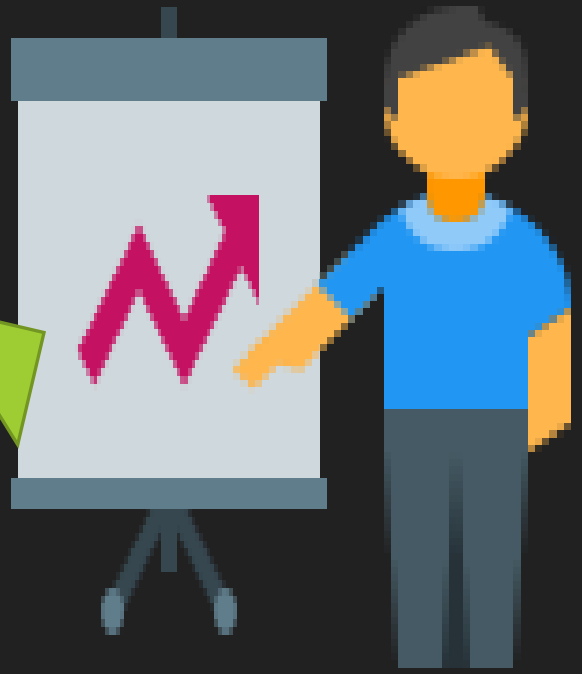


典型的なアルゴリズムを
しっかり伝授する？

https://icons8.com/

Episode：SSH課題研究の発表にて





- 分岐
- 繰返し
- 乱数

この子たちの
「楽しい」
は、こういうことなのか！

基本構造・
配列・関数
など

基本装備

各種武器

シミュレーション
・探索など

コンピュータ上
で実体験する

修行

← ← ←　生徒の状況に応じて　→ → →

みんな初めて

基本装備

各種武器

修行

活用

入試

← ← ←　生徒の状況に応じて　→ → →

3．具体的な方法の提案

3-1 概要

```
public class BubbleSort {
    public static void main(String args[]) {
        int a[] = {56,11,10,73,55,50,72,57,3,11,21,87};

        for (int i = 0; i < a.length; i++) {
            System.out.print(a[i] + " ");
        }
        System.out.println();

        boolean flag = true;
        while (flag) {
            flag = false;
            for (int i = 1; i < a.length; i++) {
                if (a[i-1] > a[i]) {
                    int w = a[i];
                    a[i] = a[i-1];
                    a[i-1] = w;
                    flag = true;
                }
            }
            for (int i = 0; i < a.length; i++) {
                System.out.print(a[i] + " ");
            }
            System.out.println();
        }
    }
}
```

バブルソートのプログラム例（言語：java）

- プログラミング言語には、細かな記述ルールがある
- 近年、タイピングの遅い生徒が増加
- 多発する実行時エラーへの教員の対応が追いつくか
- アルゴリズムの理解まで深められるか

～　提　案　～

コードを読み解き、

発展させるプログラミング活動

具体的な方法は・・・

The BEST answer!

解読 ⇨ 一部修正

実質的な理解に重きを置ける

c.f. H30愛知県センター研究発表会

「コードを読み解き，発展させるプログラミング活動」

具体的な方法は・・・

The BEST answer!

実際に動かす

- 実体験・経験として落とし込む
- 問題解決に活用する
- 試行錯誤の経験（デバッグ）

具体的な方法は・・・

The BEST answer!

様々なアルゴリズムに触れる

- スパイラル方式で繰り返す
- さまざまな組み立て方をたくさん与える

3．具体的な方法の提案

3-2 授業の流れ

１回の実践の流れ（６５分）

説明スライド

今回のコードの解読に必要な文法事項の確認

ペアワーク

提示コードを読み解く

全体

指名された何人かが説明する

個人作業

改良する

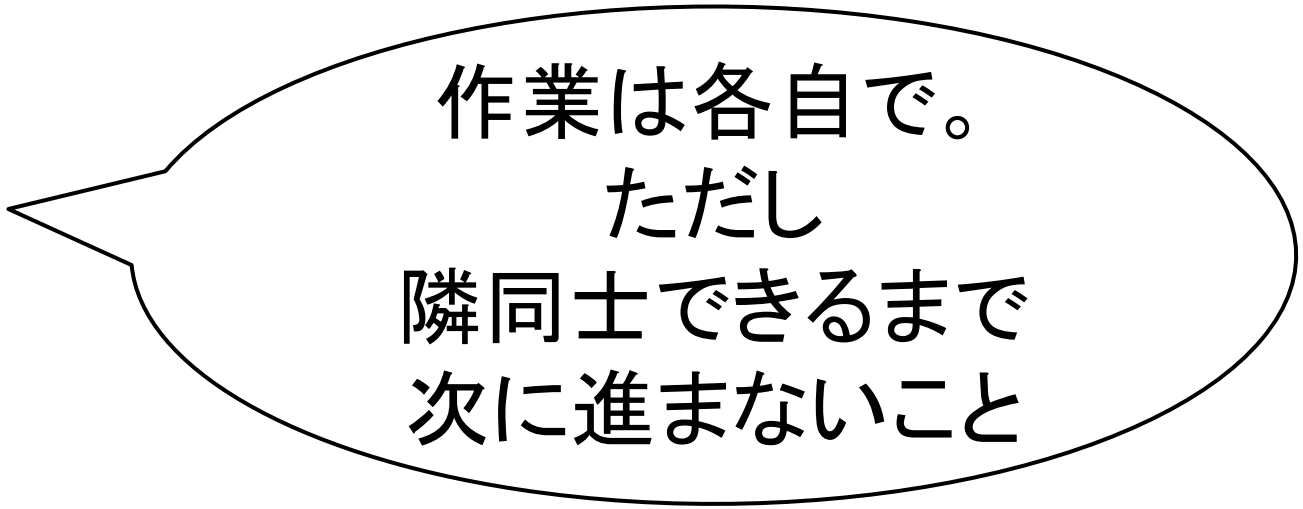
（ペアで相談）

まとめ

指名された何人かが説明する

ペアワーク：読み解き・動作確認

ペアプログラミング
2人で1つのプログラムを作成すること



デバッグも二人でやれば効率的！

コード解読：DNCLを参考にしながら

ペアワークで
相談しながら

Python　　フローチャートは右上参照

```
1  year1 = int(input('西暦年を入力'))
2
3  if year1 > 2018:
4      year2 = year1 - 2018
5      print('R', year2)
6  else:
7      year2 = year1 - 1988
8      print('H', year2)
```

DNCL

```
(01) year1 = 【外
(02)
(03) もし year1 > 2018 ならば：
(04)     year2 = year1 - 2018
(05)     表示する("R", year2)
(06) そうでなければ：
(07)     year2 = year1 - 1988
(08)     表示する("H", year2)
```

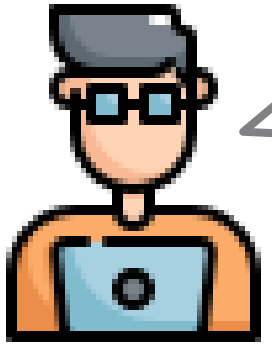
x = input(), year1 = input(x)の 2 行をまとめたもの

西暦→和暦
変換プログラム

ワークシート

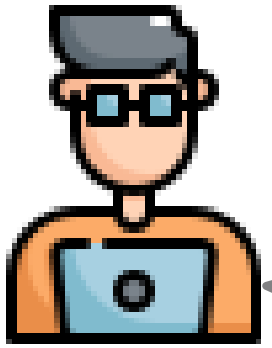
余談

ＤＮＣＬに批判めいた声が・・・？



高校生は
ＤＮＣＬとかいう謎のプログラミング言語
を学ばされるらしい。

プログラミングを
仕事にしている人
・・・のうちの
（多分）ごく一部代表



共通テストがそれ採用だから嫌でもやるしかないらしい。

実務で使えない言語を学ばせて、無駄っ！

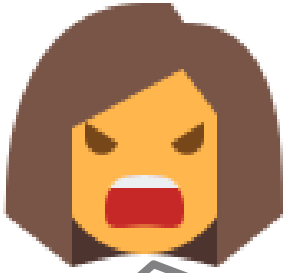
余談

ＤＮＣＬの素晴らしさについて語りたい

ＤＮＣＬはプログラミング言語ではない！

違うっ！

フローチャートに代わる**視覚化ツール**だ！



そもそも我らは**プログラマー育成事業**をやってるわけじゃない！

すべての高校生に**プログラミング的思考力**を身に付けさせる使命があるんだ！

DNCLにはフローチャートと同等の視覚効果が

【強み1】 **日本語**なのでぱっと見で頭に入る

【強み2】 **インデント**で処理のまとまりが一目でわかる

【強み3】 **縦棒**や**L字線**で制御構造が把握しやすい

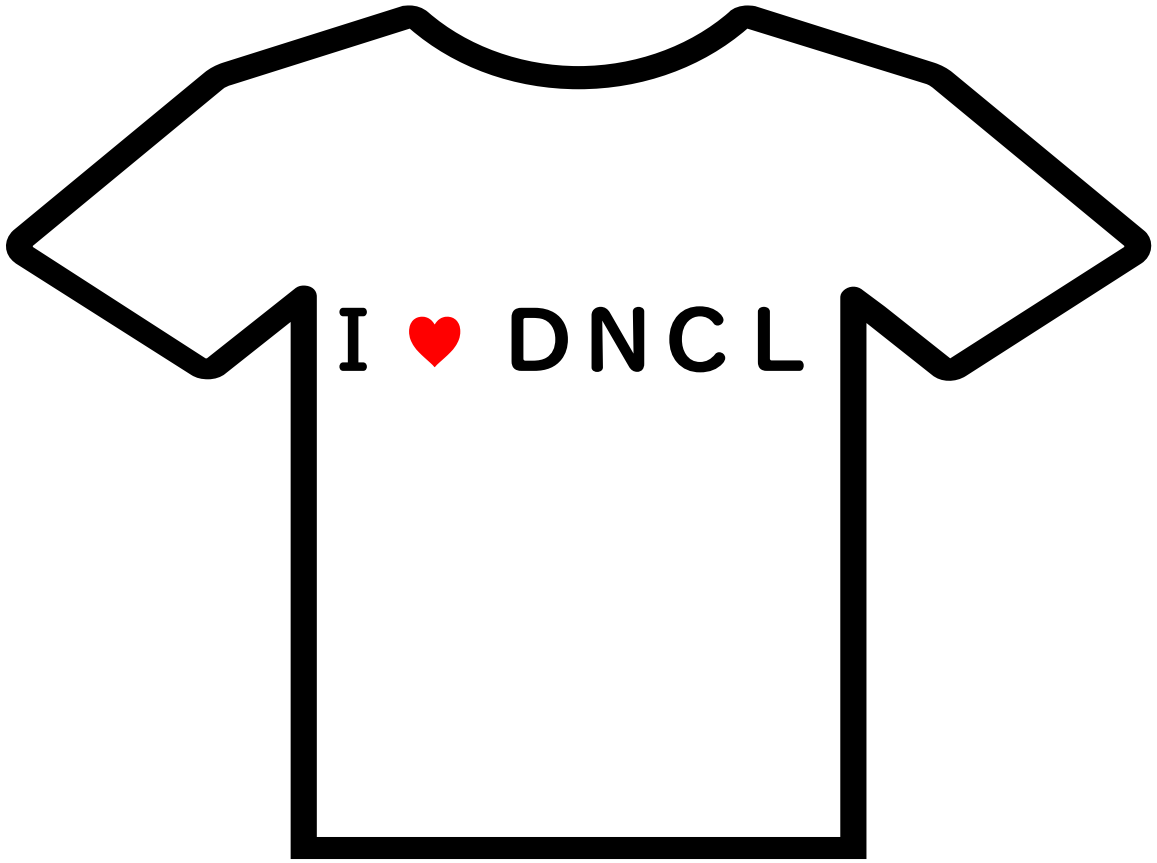
【強み4】 **Pythonコードと**並べたときに行が**1対1対応**する

フローチャートだと思って眺めてください

制御構造が
一目瞭然

DNCL

```
(01)  year1 = 【外部からの入力】
(02)
(03)  もし year1 > 2018 ならば：
(04)      
(05)      
(06)  そうでなければ：
(07)      
(08)      
```



とはいえ、やはり実機で動かして問題解決に活用させたいのでPythonを習得させたい。

DNCLはアルゴリズムの理解を助ける強力なツール、という位置づけ。

DNCLを参考にしながらコードを解読

Pythonコードを読み解く助けに

Python　　フローチャートは右上参照

```
1  year1 = int(input('西暦年を入力'))
2
3  if year1 > 2018:
4      year2 = year1 - 2018
5      print('R', year2)
6  else:
7      year2 = year1 - 1988
8      print('H', year2)
```

DNCL

```
(01) year1 = 【外
(02)
(03) もし year1 > 2018 ならば：
(04)     year2 = year1 - 2018
(05)     表示する("R", year2)
(06) そうでなければ：
(07)     year2 = year1 - 1988
(08)     表示する("H", year2)
```

x = input(), year1 = input(x)の２行をまとめたもの

西暦→和暦
変換プログラム

ワークシート

読み解けたら、実機上で動作 ⇒ 改良

IDLE

sample1.py - C:/Users/a163114500/Downloads/sample1.py (3.9.6)

File Edit Format Run Options Window Help

```
year1 = int(input('西暦年を入力'))

if year1 > 2018:
    year2 = year1 - 2018
    print('R', year2)
else:
    year2 = year1 - 1988
    print('H', year2)
```

IDLE Shell 3.9.6

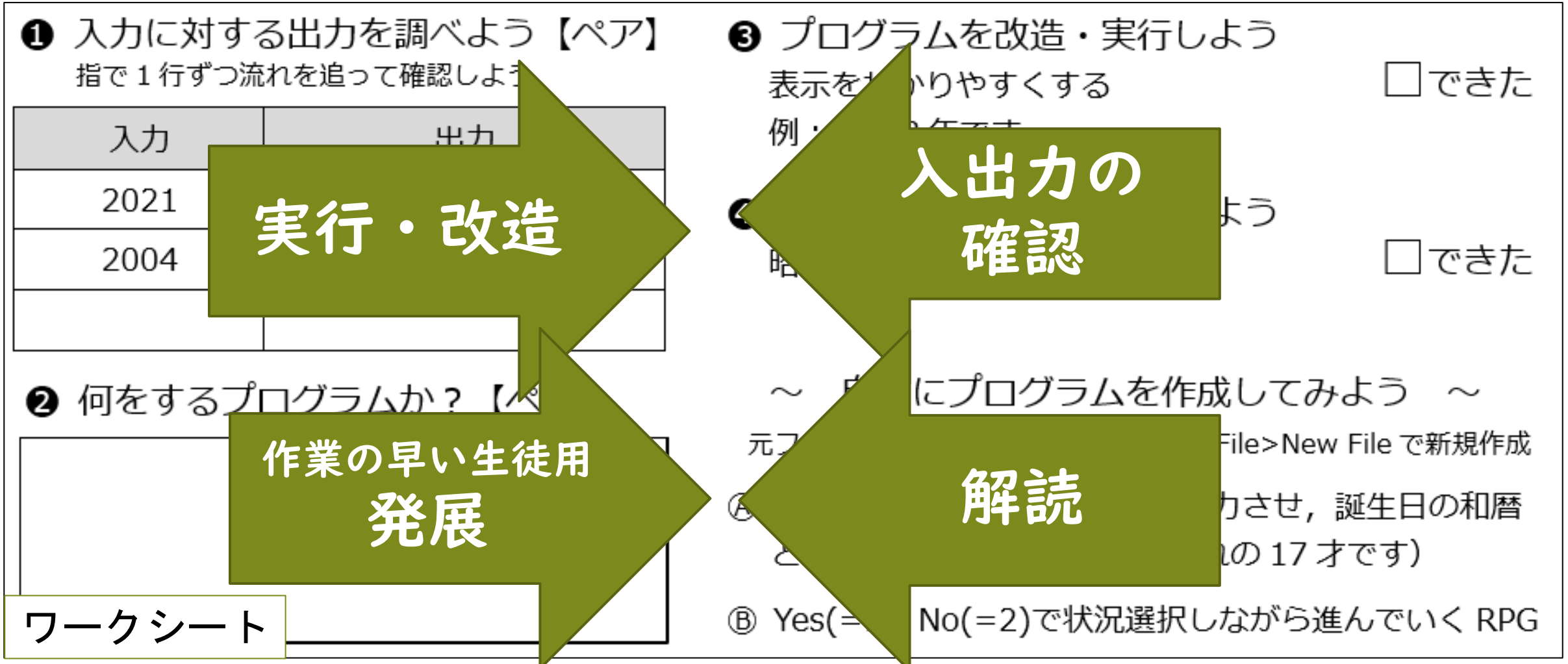
File Edit Shell Debug Options Window Help

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15
D64)] on win32
Type "help", "copyright", "credits" or "license()"
>>>
========== RESTART: C:/Users/a163114500/Down
西暦年を入力2023
R 5
>>>
```

プログラムは
あらかじめ
教員が
準備しておく

改良は生徒

❶入出力の確認 ⇒ ❷解読 ⇒ ❸実行・改造



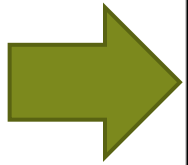
実行・改造

入出力の確認

作業の早い生徒用
発展

解読

```
1  year1 = int(input('西暦年を入力'))
2
3  if year1 > 2018:
4      year2 = year1 - 2018
5      print('R', year2)
6  else:
7      year2 = year1 - 1988
8      print('H', year2)
```



```
1  year1 = int(input('西暦年を入力'))
2
3  if year1 > 2018:
4      year2 = year1 - 2018
5      print('R', year2)
6  elif year1 >1988:
7      year2 = year1 - 1988
8      print('H', year2)
9  else:
       year2 = year1 - 1925
       print('S', year2)
```

改良は
数行でできる

西暦→和暦
変換プログラム

3．具体的な方法の提案

3-3 題材紹介

Python　　フローチャートは裏面参照

```
1  x = input('年齢を入力')
2  nenrei = int(x)
3
4  if nenrei >= 60:
5      daikin = 1200
6  elif nenrei > 18:
7      daikin = 1800
8  else:
9      daikin = 1000
10
11 print(daikin)
```

DNCL*　　*共通テストで使われる日本語ベースの言語

```
(01) x = 【外部からの入力】
(02) nenrei = 整数(x)
(03)
(04) もし nenrei >= 60 ならば：
(05)     daikin = 1200
(06) そうでなくもし nenrei > 18 ならば：
(07)     daikin = 1800
(08) そうでなければ：
(09)     daikin = 1000
(10)
(11) 表示する(daikin)
```

分岐構造

年齢別の映画料金
を表示するプログラム

題材紹介（難易度★☆☆）

Python

```
1  num = int(input('自然数を入力'))
2
3  total = 0
4  for i in range(1, num+1):
5      total = total + i
6
7  print(total)
```

DNCL

```
(01) num = 【外部からの入力】
(02)
(03) total = 0
(04) i を 1 から num まで 1 ずつ増やしながら繰り返す:
(05)     total = total + i
(06)
(07) 表示する(total)
```

反復構造

1 から指定された数までの和を表示するプログラム

題材紹介（難易度★☆☆）

Python

```python
meibo = ['山田', '林', '川井', '森']
print(meibo)
print(len(meibo), '名在籍')
print(meibo[2])
```

DNCL

```
(01) Meibo = {"山田", "林", "川井", "森"}
(02) 表示する(Meibo)
(03) 表示する(長さ(Meibo), "名在籍")
(04) 表示する(Meibo[2])
```

リスト

名簿の名前や人数
を表示するプログラム

題材紹介（難易度★☆☆）

Python

```
1 kion = [32, 28, 26, 37, 29, 30]
2 m = kion[0]
3 for i in range(6):
4     if kion[i] > m:
5         m = kion[i]
6 print('気温は', m, '度')
```

DNCL

```
(01) Kion = [32, 28, 26, 37, 29, 30]
(02) m = Kion[0]
(03) i を 0 から 5 まで 1 ずつ増やしながら繰り返す:
(04)     もし Kion[i] > m ならば:
(05)         m = Kion[i]
(06) 表示する("気温は", m, "度")
```

3 行目を `for x in kion:` とすると，リスト kion から要素を 1 つずつ取り出して x に代入しながら繰り返す → 簡潔に表現

最大値・
最小値

気温のリストから最高気温を
求めて表示するプログラム

題材紹介（難易度★★☆）

Python

```python
num = int(input('整数を入力'))
flag = False
for i in range(2, num):
    if num % i == 0:
        flag = True
        break
if flag == True:
    print('Not P')
else:
    print('P')
```

DNCL

```
(01) num = 【外部からの入力】
(02) flag = False
(03) i を 2 から num-1 まで 1 ずつ増やしながら繰り返す：
(04)     もし 余り(num/i) == 0 ならば：
(05)         flag = True
(06)         繰り返しを中断する
(07) もし flag == True ならば：
(08)     表示する("Not P")
(09) そうでなければ：
(10)     表示する("P")
```

反復・分岐・真偽値

素数判定プログラム

題材紹介（難易度★★☆）

Python

```
1  num = int(input('自然数を入力'))
2
3  total = 0
4  for i in range(1, num + 1):
5      if num % i == 0:
6          total = total + i
7
8  print('合計', total)
```

DNCL

```
(01) num = 【外部からの入力】
(02)
(03) total = 0
(04) i を 1 から num まで 1 ずつ増やしながら繰り返す:
(05)     もし 余り(num/i) == 0 ならば :
(06)         total = total + i
(07)
(08) 表示する("合計", total)
```

反復・分岐・
割り算余り

指定された数の
約数の和を求めるプログラム

題材紹介（難易度★★☆）

Python

```python
import random

num = int(input('回数を入力'))
atari = 0
for j in range(num):
    x = random.randint(1, 6)
    if x == 1:
        atari = atari + 1
print('理論値', 1/6)
print('試行結果', atari/num)
```

DNCL

```
(01) random モジュールのインポート
(02)
(03) num = 【外部からの入力】
(04) atari = 0
(05) num 回繰り返す：
(06)     x = 乱数(1, 6)
(07)     もし x == 1 ならば：
(08)         atari = atari + 1
(09) 表示する("理論値", 1/6)
(10) 表示する("試行結果",
```

シミュレーション

サイコロシで１の目が出る
確率のシミュレーション

題材紹介（難易度★★★）

Python

```
 1  import random
 2
 3  kosu = 1000
 4  kosuIN = 0
 5  for i in range(kosu):
 6      x = random.random()
 7      y = random.random()
 8      if x**2 + y**2 <= 1.0:
 9          kosuIN += 1
10  print('値', 4.0 * kosuIN / kosu)
```

DNCL

```
(01)  random モジュールのインポート
(02)
(03)  kosu = 1000
(04)  kosuIN = 0
(05)  kosu 回繰り返す：
(06)      x = 乱数(0～1)
(07)      y = 乱数(0～1)
(08)      もし x**2 + y
(09)          kosu
(10)  表示する("
```

モンテカルロ
シミュレーション

ランダムな点が円に入る確率
シミュレーション（円周率）

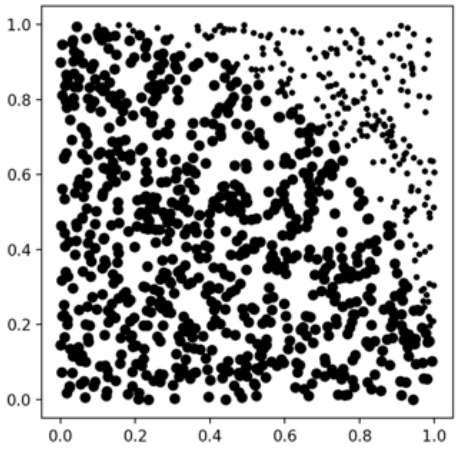
題材紹介（難易度★★★）

【参考】グラフ描画ライブラリ matplotlib の利用

① Windows のコマンドプロンプトを起動
② pip install matplotlib

【4－2例題】を視覚的にシミュレーション

```
1  import random
2  # matplotlib の pyplot モジュールを plt としてインポート
3  import matplotlib.pyplot as plt
4
5  kosu = 1000
6  kosuIN = 0
7  # グラフの軸の縦横比を同じにする
8  plt.axes().set_aspect('equal')
9  for i in range(kosu):
10     x = random.random()
11     y = random.random()
12     if x**2 + y**2 <= 1.0:
13         kosuIN += 1                # 変数 kosuIN に 1 を加算(kosuIN = kosuIN + 1 と同じ)
14         plt.scatter(x, y, c='black', marker='o')  # 座標(x, y)の位置にマーカー'o'を表示
15     else:
16         plt.scatter(x, y, c='black', marker='.')  # 座標(x, y)の位置にマーカー'.'を表示
17 print("円周率:", 4.0 * kosuIN / kosu)
18 plt.show()
```



matplotlib 版

ランダムな点が円に入る確率シミュレーション（円周率）

題材紹介（難易度★★★）

線形探索

バブル
ソート

鋭意制作中

クイック
ソート

ユーザ定義
関数

二分探索

題材紹介（難易度★★☆～★★★）

難易度別

難易度★
西暦和暦変換
１～１０の和
数え上げ

難易度★★
約数とその和
最大・最小
素数判定
ユーザ定義関数

難易度★★★
線形探索
バブルソート
二分探索
クイックソート

難易度★★★
シミュレーション

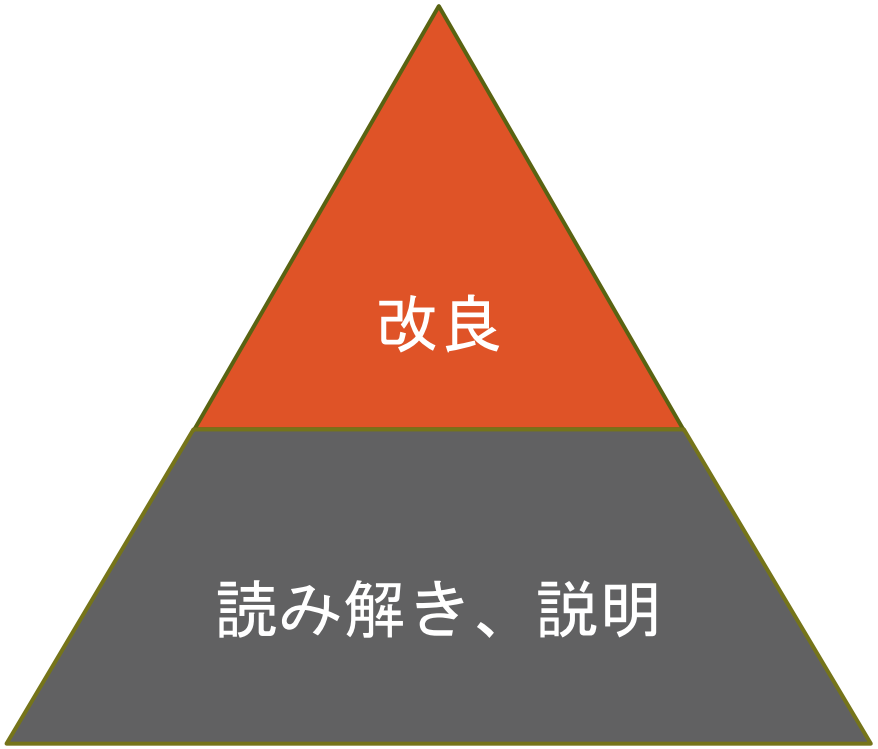
（実際のコードを触る方式は）
ここまででいいかな

3．具体的な方法の提案

3-4 評価

評　価

規準	プログラム**を**読み解き、その手順や仕組みを説明できている**。**
A （十分満足できる状況）	プログラムを**読み解き**、その手順や仕組みを**説明した上で**、**コードを考慮した改良**ができている。
B （望まれる全員に到達してほしい状況）	プログラムを読み解き、その手順や仕組みを説明できている。
C （努力を要する状況）	手順や仕組みの説明が不十分である。



ワークシート・
改良プログラム

4. 学 バイブル 裏付け

学習指導要領

プログラミング教育で育む資質・能力

知識・技能

アルゴリズムを表現する手段，プログラミングによってコンピュータや情報通信ネットワークを活用する方法について理解し技能を身に付けること。

高等学校新学習指導要領より抜粋　～ 1/2 ～

プログラミング教育で育む資質・能力

知識・技能

アルゴリズムを表現する手段

コンピュータを活用する方法

理解し技能を身に付ける

高等学校新学習指導要領より抜粋　～ 1/2 ～

プログラミング教育で育む資質・能力

思考力・判断力・表現力等

目的に応じたアルゴリズムを考え適切な方法で表現し，プログラミングによりコンピュータや情報通信ネットワークを活用するとともに，その過程を評価し改善すること。

高等学校新学習指導要領より抜粋　～ 2/2 ～

プログラミング教育で育む資質・能力

思考力・判断力・表現力等

目的に応じたアルゴリズムを考え表現し，コンピュータを活用するとともに，その過程を評価し改善すること。

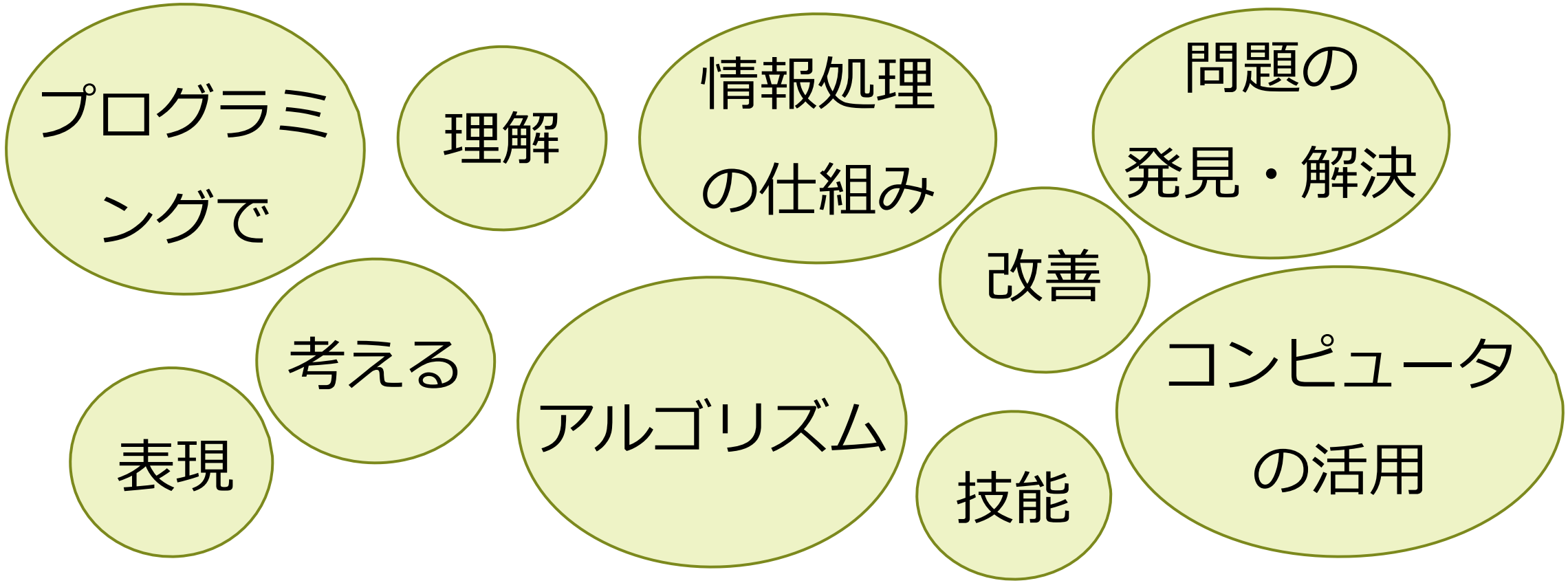
アルゴリズムを考え表現

コンピュータを活用

その過程を評価し改善

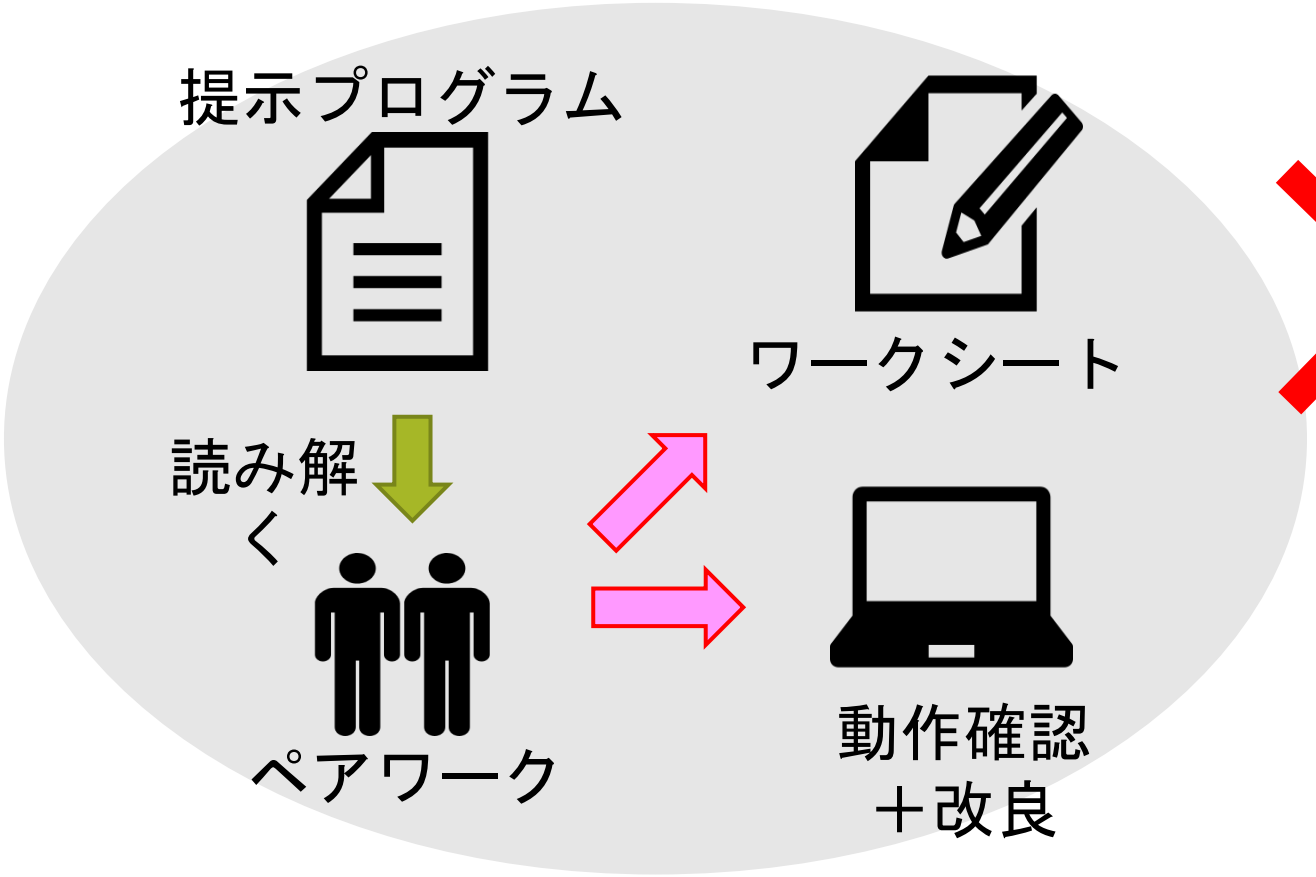
高等学校新学習指導要領より抜粋　～ 2/2 ～

これらを効果的に身に付けさせる
ことができる授業実践



5. おわりに

できるだけたくさん繰り返す



×何回も

・多種多様なアルゴリズム
・さまざまな難易度

でネタを増やしていき、
時間の許す限り実践

教材データ

ワークシート・プログラムファイル等