

生徒自身のペースで進める プログラミングによる シミュレーションの実践と検討

第17回全国高等学校情報教育研究会全国大会（愛知大会）

神奈川県立横浜国際高等学校

情報科 鎌田高徳

2024/8/4 B3-5



International
Baccalaureate
Organization

自己紹介

氏名：鎌田 高德（かまだ たかなり）

教職歴（教科情報採用15年目）

- 神奈川県立川崎高等学校（5年）
- 神奈川県立茅ヶ崎西浜高等学校（7年） **プログラミング教育研究推進指定校**
- **神奈川県立横浜国際高等学校（3年目）** **学力向上進学重点校・IB校**
（国際科・国際バカロレアコース）



活動等

- 文科省 情報Ⅰ・情報Ⅱ解説動画
- 神奈川県情報部会実践事例報告会運営
- 活動の詳細はWebサイトに<https://sites.google.com/site/johoeportfolio/home>

文科省が一昨年度、生徒向けの授業動画を初制作

第3章のプログラミングの動画→問題解決のためのプログラミング

文部科学省「情報Ⅰ」学習動画
 授業、教材研究、自学自習で使えます！（本人談）
 神奈川県立横浜翠嵐高等学校 三井栄慶

概要

令和4年11月15日付で「高等学校情報科に係る指導体制の一層の充実について」が通知がなされた。この通知内「2. 施策パッケージについて (2) 文部科学省が授業動画の作成及びNHK高校講座「情報Ⅰ」へ協力を行った趣旨」において「教師の研修、授業の一部における放映及び生徒の予習・復習・自学自習等に幅広く活用できる動画教材を国の責任で提供・充実させることにより、教師が安心して教えることができ、生徒が楽しみながら学べる環境を整備することにあります。」とある。そこで本稿では文部科学省「情報Ⅰ」学習動画を活用事例を紹介し、広く活用していただけることを目的とする。

「情報Ⅰ」学習動画の位置づけ



教材研究でも使えます！



授業の一場で使えます！



生徒の自学自習でも使えます！



「情報Ⅰ」学習動画 コンピュータとプログラミング 動画リスト

- [0] 「コンピュータとプログラミングについて」
- [1] 「センサーライトを作ろう！」
- [2] 「100連ガチャをプログラムして作ろう！」
- [3] 「公平な方法で発表の順番を決めよう！」
- [4] 「天気予報表示マシンを作ろう！」

令和4年度神奈川県情報部会主催「情報科実践事例報告会」

続いて文科省が昨年度、**情報Ⅱ**の授業動画を制作 **問題解決**のための**コンテンツ**制作や**情報システム**の開発

高等学校情報科「情報Ⅱ」授業・研修用コンテンツ

コミュニケーションとコンテンツ



コミュニケーションとコンテンツについて



コースを考えたコンテンツ!動画制作
学校紹介の動画を作ろう



ユーザー中心のコンテンツ!Webサイト制作
文化祭のWebサイトを作ろう



メタバースを楽しもう!
バーチャルリアリティーの可能性

情報システムとプログラミング



情報システムとプログラミングについて



Bit Arrowの使い方①
教員アカウント等の登録



Bit Arrowの使い方②
グループチャットの作り方



情報システムってなに?
身近な情報システム開発から学ぶ



情報システムを設計しよう!



ドキドキワクワク制作と動作テスト!



統合テストと評価をまして改善へ!

今年は12月26日（木）に神奈川県情報部会 実践事例報告会オンライン開催します。

「交流は授業づくりの問題解決（柴田）」

情報部会でぜひ検索！

神奈川県高等学校教科研究会情報部会主催
神奈川県教育委員会共催・神奈川県立総合教育センター選択研修講座

情報科実践事例報告会2023

【日時】令和5年12月26日(火) 9:00～17:00 【形式】ZOOM+YouTube+Slack

オンデマンド発表の部 (YouTube)

9:00～ 9:15 オープニング (ZOOM)
9:15～11:00 **オンデマンド発表(YouTube)**
10:30～11:25 オンデマンド座談会 (Zoom)
11:25～13:00 ランチ交流タイム (Zoom)
13:00～13:10 午後の部開会 (ZOOM)
13:10～15:40 オンラインポスター発表(ZOOM)
15:40～16:40 講演・指導講評(ZOOM)
16:10～17:00 指導講評・クロージング

- A-01 完全ペーパーレス授業～その時、生徒は？教員は？～
岡山県立岡山操山高等学校 太田 重成
- A-02 すべての高校生に「基礎情報学」のエッセンスを！
—3つの情報概念をめぐって—
京都市立日吉ヶ丘高等学校 藤岡 健史
- A-03 ChatGPTを活用した「情報Ⅰ」大学入学共通テスト対策
問題制作の実践報告
帝京大学大学院 理工学研究科 修士課程学生 植垣 新一
- A-04 県立高校でプログラミングワークショップ！
7月の授業実践&訪問授業の受付始動の報告
教育団体：やりたいこと支援サークル ideal
- A-05 生体情報に基づいた評価方法の提案
神奈川県立横浜修徳館高等学校 堀内 宏基
- A-06 PC教室で一人一台の活用を前提とした情報Ⅰの授業実践
報告2023冬
千葉県立船橋啓明高等学校 谷川 佳隆
- A-07 「情報科」の授業で3Dプリンタを使ってみた
神奈川県立相模原城山高等学校 浅井 雄大
- A-08 「1年後を見据えた」教材リンク集の作成
富山県立高岡高等学校 多賀 誠志
- A-09 専門教科「商業」の情報教育
川崎市立幸高等学校 天明 大輔
- C-01 技術に寄せないシミュレーション：現実により近い
感染症シミュレーションを作りパラメータを変えて
考察した1学期の試み
相模女子大学中部・高等部 堺 和貴子
- C-02 PythonとJavaScriptに取り組んでみて
和歌山県立向陽高等学校 西林 諒
- C-03 Monaco Educationを活用したPythonの授業実践例2023
千葉県立船橋啓明高等学校 谷川 佳隆
- C-04 Pictogrammingによるプログラミングの学習2
横浜市立戸塚高等学校定時制 杉山 大海
- C-05 楽しく学ぶ！プログラミングやってみました
神奈川県立希望ヶ丘高等学校 海沼 絵里
- C-06 「アルゴリズムとプログラミング」実践報告
東京都立町田高等学校 小原 格
- C-07 生徒の疑問へのチャレンジ精神を引き出す取り組み
～情報オリンピック (JOI) 対策夏期講習を通して～
東京都立新宿山吹高等学校 中山 享司
モノグサ株式会社・株式会社NTTデータ数理システム大槻 兼貴
- C-08 「プログラミングわからん！」のパターン調査
大阪府立八尾高等学校 阿多 悠生
- C-09 中学生とmicro:bitでやってきたこと10選
樟蔭中学校・高等学校 川浪 隆之

- A-11 [純]人工知能と人間との関わり方を考える
～AI(人工知能)が人を救く未来～
田園調布学園中等部・高等部 村山達哉
- A-12 高校版GIGAスクールで何を創造するか
神奈川県立希望ヶ丘高等学校 柴田 功
- A-13 情報Ⅰの導入を工夫してみよう
(生徒にとって身近で切実な題材で)
神奈川県立横浜国際高等学校 鎌田 高徳
- A-14 今年度、こんな授業を実施しました2023
(授業実践の紹介)
神奈川県立上鶴間高等学校 相馬 臣彦
- A-15 「総合的な探究の時間」と「情報Ⅰ」との連携について
横浜市立横浜総合高等学校 兵 桂一
- A-16 共通テストを意識した情報Ⅰの定期テスト問題作成
土浦日本大学中等教育学校 瀧見 友章
- A-17 超進学校！！横浜翠嵐高等学校！年間の「情報Ⅰ」
カリキュラム
神奈川県立横浜翠嵐高等学校 山崎 優佳里
- A-18 サイバーポリスゲームの問題作成を利用した情報モラル
教育
大阪府立門真なみはや高等学校 野郎 緑
- ▲B-01 色に関するいろイロ～配色はセンスじゃない
近江兄弟社高等学校 長谷川友彦
- C-11 身近な題材を例に繰り返し学ぶ プログラミング
～2023年度改訂版～
世田谷学園高等学校 神藤 健朗
- ★D-01 データの読み取りおよび活用を考える授業
～個人情報 編～
東大寺学園中学校・高等学校 吉田 拓也
- ★D-02 ネットワーク七転八倒：実習や体験を盛り込んで挑む
「苦手な単元」の記録
相模女子大学高等部 堺和貴子
- ★D-03 レッスンスタディを通じた統計授業フレームワーク
及びWebアプリケーション教材の開発
大分県立大分舞鶴高等学校 伊藤 大貴
- ★D-04 情報Ⅰの授業で仮説検定を扱った！
神奈川県立横浜翠嵐高等学校 三井 栄慶

●問題解決・情報科全般・ICT活用
▲情報デザイン・コミュニケーション・コンテンツ
■プログラミング・アルゴリズム
★データ活用
※発表者の敬称は省略しています
参加申込は情報部会ホームページ <http://www.johobukai.net/>

本校のサマープログラムで外部機関 (やりたいこと支援サークル) と連携



昨年度 (Python) → ライフゲームなど制作
プロのプログラマーからプログラミングを学ぼう
けんちゃんさん、uchanさん



今年度 (Unity) → FPS制作
プログラミングでゲームを作るぜ!
もりくらげさん



やりたいこと支援サークル
<https://yaritaikotoshien.com/stories/>

伝えたいこと

先生がすべて学習活動をコントロールするのではなく
生徒自身のペースで学ぶ情報の時間を増やしませんか？

- ①自ら問題を発見し
- ②自ら解決方法を計画し
- ③自ら実際に実行し
- ④自ら評価を行い振り返る

問題解決能力を育成する授業をやってみませんか？

発表の流れ

(1) 情報Ⅰと情報Ⅱにおけるプログラミングによる
シミュレーション

(2) シミュレーションに関する事例

(3) 授業の実践と検討

- ①生徒自身のベースで学習を進めるプログラミング学習教材の作成
- ②プログラミングの題材を身近な事象に
- ③情報Ⅰのカリキュラム再編

学習指導要領においては具体的なシミュレーションの手法の記載はない

(3) コンピュータとプログラミング

コンピュータで情報が処理される仕組みに着目し、プログラミングやシミュレーションによって問題を発見・解決する活動を通して、次の事項を身に付けることができるよう指導する。

ア 次のような知識及び技能を身に付けること。

(ア) コンピュータや外部装置の仕組みや特徴、コンピュータでの情報の内部表現と計算に関する限界について理解すること。

(イ) 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。

(ウ) 社会や自然などにおける事象をモデル化する方法、シミュレーションを通してモデルを評価し改善する方法について理解すること。

イ 次のような思考力、判断力、表現力等を身に付けること。

(ア) コンピュータで扱われる情報の特徴とコンピュータの能力との関係について考察すること。

(イ) 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。

(ウ) 目的に応じたモデル化やシミュレーションを適切に行うとともに、その結果を踏まえて問題の適切な解決方法を考えること。

学習指導要領解説ではいくつか事例があり 問題解決のためのシミュレーションを行う必要がある

例えば、モデル化とシミュレーションに関する学習活動としては、平面図等を利用した家具の配置等の単純なモデルによるシミュレーションやシミュレーションソフトウェアを利用した体験を通して、事象をどのようにモデル化しているのかを調べたり、生徒自らがモデル化を行ったり、モデル化の長所と短所を調べたりする学習活動などが考えられる。

また、数式等を利用しない単純なモデルを利用したシミュレーションなどの後に、コンピュータを活用した金利計算や通信に関する料金プランのモデル化とシミュレーションを行ったり、シミュレーションの仕組みを考えたりする学習活動が考えられる。更に、関係する変数が少ない事象を数式で表す技能を身に付け、変数に代入する値を変えるなどしながらシミュレーションを繰り返し、適切な解決方法を発見したり選択したりする学習活動が考えられる。

学校や地域の実態及び生徒の状況に応じて乱数を用いたシミュレーションなどを題材とするとともに、インフルエンザが爆発的に増える理由、感染を抑えるための方法について考えるような題材を基にモデル化とシミュレーションを行う学習活動などが考えられる。また、必要に応じて天体シミュレーション、物理シミュレーションや流体シミュレーションなどの専用のシミュレーションソフトウェアの利用やプログラミングによるシミュレーションを行う学習活動も考えられる。

情報Ⅰの解説動画

- 「学業時間と睡眠時間」の単回帰分析

情報Ⅱの解説動画

- 「学業・通勤通学・休養くつろぎ・趣味娯楽と睡眠時間」の重回帰分析

既存の調査結果で問題解決
オープンデータの活用

単回帰分析

2	30
3	-10
4	20
5	9.2

決定係数「 R^2 」は、「1」に近いほど
点に近寄って、線が引かれているという意味になる

※決定係数は0から1の値を取り、1に近いほど当てはまりがよいと言える

重回帰分析のモデルで予測する

重回帰分析を用いた予測
睡眠時間を他の行動時間から予測しよう

```
3 model2.fit(X2, y)
4 print(model2.intercept_, model2.coef_)
[612.39863457] [[-0.23889623 -0.54526488 -0.16213081 -0.16213081 -0.16213081 -0.16213081]]
```

1 X2_yosoku = pd.DataFrame([[400, 70, 135, 40]])

Two-dimensional, size-mutable, mutable container of objects that can be indexed by row and column labels. This structure is similar to the other data structures in Pandas, but the output of arithmetic operations align on both the row and column labels. This makes it useful to have mixed-integer or integer indexes aligned to a particular type of axis, thought of as a dict-like container for objects.

X2_yosoku = pd.DataFrame([[400, 70, 135, 40]],
columns = ['学業', '通勤・通学', '休養・くつろぎ', '趣味・娯楽'])

問題の発見・解決に向けて主体的に情報と情報技術を活用し、自ら評価し改善しようとする人材育成

共通教科情報科の評価の観点及びその趣旨

【学習指導要領 第2章 第10節 情報「第1款 目標」】

(1)	(2)	(3)
情報と情報技術及びこれらを活用して問題を発見・解決する方法について理解を深め技能を習得するとともに、情報社会と人との関わりについての理解を深めるようにする。	様々な事象を情報とその結び付きとして捉え、問題の発見・解決に向けて情報と情報技術を適切かつ効果的に活用する力を養う。	情報と情報技術を適切に活用するとともに、情報社会に主体的に参画する態度を養う。

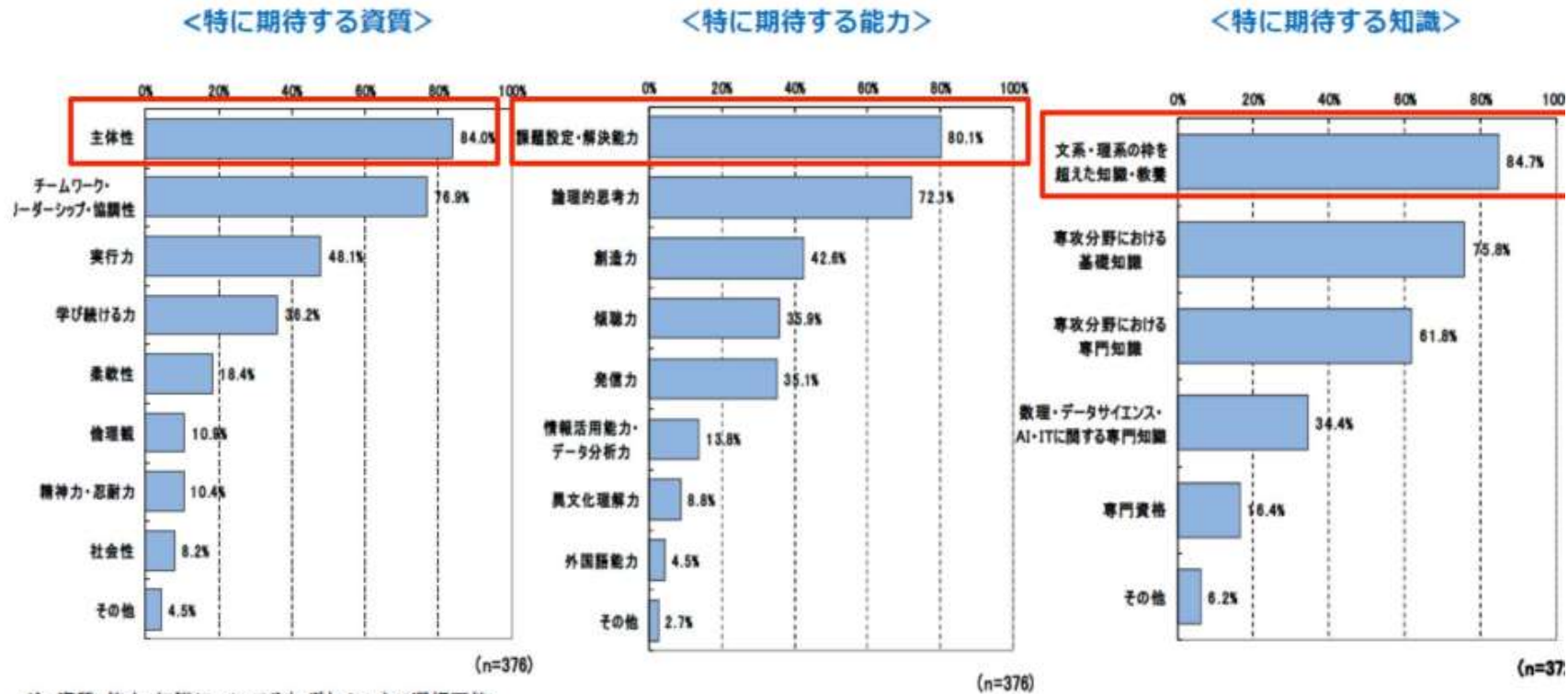
【改善等通知 別紙5 各教科等の評価の観点及びその趣旨 <情報>】

知識・技能	思考・判断・表現	主体的に学習に取り組む態度
情報と情報技術を問題の発見・解決に活用するための知識について理解し、技能を身に付けているとともに、情報化の進展する社会の特質及びそのような社会と人間との関わりについて理解している。	事象を情報とその結び付きの視点から捉え、問題の発見・解決に向けて情報と情報技術を適切かつ効果的に用いている。	情報社会との関わりについて考えながら、問題の発見・解決に向けて主体的に情報と情報技術を活用し、自ら評価し改善しようとしている。

<https://www.nier.go.jp/kaihatsu/shidousiryou.html>

大学卒業者に特に期待する資質・能力・知識

○ 大学卒業者に期待される資質・能力・知識として、特に期待する資質は「**主体性**」、特に期待する能力は「**課題設定・解決能力**」、特に期待する知識は「**文系・理系の枠を超えた知識・教養**」と回答した企業が最多であり、探究的な学びや文理横断型のリベラルアーツ教育が求められている。



注: 資質・能力・知識についてそれぞれ3つまで選択可能

(出典) 一般社団法人 日本経済団体連合会「採用と大学改革への期待に関するアンケート結果」

R5.2.20個別最適な学びと協働的な学びの一体的な充実に向けた学校教育の在り方に関する特別部会(第3回)・義務教育の在り方ワーキンググループ(第6回)・高等学校教育の在り方ワーキンググループ(第6回)合同会議参考資料2より

発表の流れ

(1) 情報Ⅰと情報Ⅱにおけるプログラミングによる
シミュレーション

(2) シミュレーションに関する事例

(3) 授業の実践と検討

- ①生徒自身のベースで学習を進めるプログラミング学習教材の作成
- ②プログラミングの題材を身近な事象に
- ③情報Ⅰのカリキュラム再編

感染者数をシミュレーションした実践事例（小松） 表計算ソフトとPythonの比較

生徒の反応（毎時間の振り返り）

理解度

- 1:よく理解できた
- 2:理解できた
- 3:まあまあ理解できた
- 4:イマイチわからない
- 5:わからない

満足度

5点満点

	2020年度	2021年度
理解度	2.19	3.28
満足度	4.11	3.58



```
import matplotlib.pyplot as plt

kansen=[556,392,563,748,888,884,949,708,481,856,944,1337,783,814,816,884,1278]
heikin=["-.-.-.-.-"]
r=list()

for i in range(0,9,1):
    goukei=kansen[i]+kansen[i+1]+kansen[i+2]+kansen[i+3]
    +kansen[i+4]+kansen[i+5]+kansen[i+6]
    heikin.append(goukei/77)
    r.append(kansen[i+8]/heikin[i+3])

print(r)
plt.plot(r)
plt.show()
```

Pythonの基礎をやって、表計算ソフトでシミュレーションをした後に、そのシミュレーションをPythonでやってみた

→データの活用を学んだ後にPythonをしてみてもは？

表計算はオートフィルで直感的に理解できるが、プログラミングは配列とループの理解が必須

→配列・繰り返しの組み合わせが鬼門

創造的写経プログラミング（春日井）

生徒がオリジナリティを考えプログラムを発展させる

創造的写経プログラミングとは

創造的：授業で学習したプログラムをもとに
それを発展させて新しいプログラムを作り出すこと

写経：もともになるプログラムを模倣する
プログラムを模倣することによる
プログラミングの理解の深化を目的とする

要するに、学習したプログラムを発展させて
生徒自身のオリジナルのプログラムを作成すること

授業で学習したプログラムをもとに、それを発展させて新しいプログラムを作り出す
→いきなりプログラムを作りなさいは難しい
いくつかプログラムのモデルが必要

授業で生徒に示したプログラムの問題②

【問題1】

西武新宿線の本川越駅から所沢駅までの区間について、2021年度の1日平均の乗降人員は次の表1のとおりであった。2000人を■印1個としてグラフを作成したい。空欄をうめてプログラムを完成させなさい。

表1 本川越から所沢までに2021年度1日平均乗降人員

駅名	1日平均乗降人員	駅名	1日平均乗降人員
本川越	43048	入管	14168
南大塚	14620	新所沢	42857
新狭山	15973	航空公園	20480
狭山市	32966	所沢	86613



生徒が作成したプログラム⑤

【プログラム】惑星の直径を比較するプログラム

```
def graph(number):
    star = round(number / 3000)
    return star

planet = ['水星', '金星', '地球', '火星', '木星', '土星', '天王星', '海王星']
chokkei = [4879, 12104, 12756, 6702, 142984, 120536, 51118, 49528]

for i in range(0, 8, 1):
    star = graph(chokkei[i])
    print(planet[i], end=':')
    for j in range(0, star, 1):
        print('★', end='')
    print(chokkei[i])
```




成人のプログラミング学習のハードルとその越え方（井澤）

ITエンジニアの視点から

まとめ

1. プログラミング学習のハードル
 - プログラミング環境をどう準備したらよいか分からない
 - どこまで学ばよいか分からない
 - 学習をどう進めたらよいか分からない
 - 「失敗」が怖い
2. 最初のハードルの越え方
 - 「PCの管理者」
 - 学習指導要領/教科書/指導計画を見直す
 - スモールステップで少しずつ取り組む
 - エラーメッセージと付き合う
3. 次のハードルはどこにある？




学習指導要領/教科書/指導計画を見直す

- 学ぶ範囲は、自分で決めてよい
- プロのエンジニアでも、「とりあえず自分が今必要としている範囲（+アルファ）」を目標とし、必要に応じて学び足していくことがほとんど

学校の場合

- 学習指導要領の「プログラミング」
 - 小学校・中学校・高校
- 教科書
- 教科書会社・教材会社の示す指導計画
- 入試の範囲



高校『情報Ⅰ』の教科書の
“プログラミングの基礎”の範囲

- 制御構造
 - 順次
 - 分岐
 - 繰返し
- データ構造
 - 変数
 - 配列/リスト
- 関数

これらのハードルをどう超えるか

→ スモールステップで生徒が失敗を恐れずに学べる
教材と指導計画を検討する

プロのエンジニアでも+アルファ

→ 学ぶ範囲を生徒たちに決めさせる工夫と
モデルとなるプログラムを自分たちの手で
改変させて学ぶ授業を

全高情研2022年 井澤哲也

成人のプログラミング学習のハードルとその越え方

https://www.zenkojoken.jp/wp-content/uploads/2022/08/R2-1_izawa.pdf

発表の流れ

(1) 情報Ⅰと情報Ⅱにおけるプログラミングによる
シミュレーション

(2) シミュレーションに関する事例

(3) 授業の実践と検討

①生徒自身のベースで学習を進めるプログラミング学習教材の作成

②プログラミングの題材を身近な事象に

③情報Ⅰのカリキュラム再編

プログラミングにおける 指導の個別化と学習の個性化

「個別最適な学び」と「協働的な学び」

個別最適な学び

子ども一人ひとりの特性や興味関心に応じて、
子ども自身が学習を進めていく学び

指導の個別化



子ども一人ひとりに応じて、
学習方法や教材、
学習時間等を柔軟に設定する

学習の個性化



子ども一人ひとりに応じて、
学習活動や学習課題に
取り組む

協働的な学び

多様な他者との交流を通して、
必要な資質・能力を育成する
ための学び

地域の人 異学年の子ども



クラスメイト



教員、クラスメイト、地域の人など
多様な人との交流を通して
よりよい学びを生み出す

問題解決のための情報 I の授業の工夫 自由に歩き回って聞きたい人に聞いてよい

・授業構成

- ① 12分説明
- ② 15分問題解決 1
- ③ 15分問題解決 2
- ④ 8分振り返り

・教室のレイアウト

4人1グループに変更
グループ内で協力して
問題に取り組む



プログラミングを学ぶ環境を生徒たちに選択させる (作業用BGMを生徒から集約して流す)



ペーパーを見ながら一人で
プログラミング



PDFをスマホを見ながら一人で
プログラミング



席を移動して2人で相談しながら
プログラミング



どうしようもないデバッグは先生と
プログラミング

Pythonのプログラミングの課題を12の課題をSmall Stepで設定して基本課題と応用課題に分け、各自のペースで進める 50分×2コマで最後までできた人はオリジナルプログラム制作

ステップの内容	
3-1	変数aに1~100の数を代入し、aの値を表示させる
3-2	3-1で作ったプログラムを、1が出たら'Atari'、それ以外は'hazure'が表示されるプログラムに書き換える
3-3	3-2のプログラムを100回繰り返し、100連ガチャを作る
3-4	さらに当たった回数(変数atarikai)と外れた回数(変数hazurekai)を表示させる
3-5	実行結果が縦に長すぎるので、当たった回数(変数atarikai)と外れた回数(変数hazurekai)のみを表示させるようにする
3-6	<ul style="list-style-type: none"> 前回の実行結果をatarikaiのみ表示させるようにして100回繰り返し、実行結果をスプレッドシートに貼り付けて、100連ガチャをしたら何%の確率でゲットできるか明らかにする プログラムが完成したら実行を押し、課題にあるスプレッドシートを開き、実行結果をスプレッドシートに貼り付けてシミュレーション結果から分かったことをまとめる ガチャのシミュレーション結果が表示されるので、グラフから読み取れた内容を書き込む

**50分でほぼすべての生徒が3-6まで到達
ガチャのシミュレーションの結果から考察を書くところまでできた**

ステップの内容	
3-7	While文を使って1%のガチャが当たるまで100回引く、当たったらガチャ終了のプログラムを作る
3-8	3-7の結果の表示が縦に長すぎるので、何回目に当たったかだけの結果を表示させる
3-9	当選確率3%のガチャを100回行うプログラムを作る
3-10	SSRが3%、SRが18%、Rが79%で排出されるガチャを100回分実行し、SSR・SR・Rがそれぞれ何回出るか表示されるプログラムを作る
演習課題 3-1	3-10は、あるゲームの実施のガチャの当選確率を参考に作られている。あなたもスマホやPCの検索で実際にスマホ上のガチャの当選確率をモデルに100連ガチャを作り、プログラミングで実装しなさい。
演習課題 3-2	演習課題3-1であなたが実装したガチャのモデルにおいて、 1番当てたいガチャは100連ガチャをしたら何%の確率で1個以上はゲットできるか 、ガチャの結果をスプレッドシートに貼り付け、シミュレーションをして求めなさい（3-6を参考にしてみよう）

**100分でほぼすべての生徒が最後まで到達
自分で探したモデルを元にガチャのシミュレーションの結果から考察を書くところまでできた**

事例325 キミのミライ発見

生徒自身のペースで進める「100連ガチャのプログラミングによるシミュレーション」～

「アルゴリズムとプログラミング」の全体設計～ <https://www.wakuwaku-catch.net/jirei24325/>

1 回目のプログラムのみ自分ですべて写経する
2 回目以降は 1 つ前のプログラムをコピペして
資料を見ながら書き換えてプログラムを改良していく
(資料には **7割答えが載っている** ように調整した)

3-1. 第1段階「変数aに1~100の乱数を代入し、aの値を表示させよう」

```
1 import random
2 a=random.randint(1,100)
3 print(a)
```

乱数の最小値 乱数の最大値

69

1~100の乱数が表示される

3-2 第2段階「先ほどのプログラムを1がでたら'atari' それ以外は'hazure'が表示されるプログラムにしよう」 (前のプログラムをコピペして書き換えよう)

```
1 import random
2 a=random.randint(1,100)
3 if(a==1):
4     print(a,'atari')
5 else:
6     print(a,'hazure')
```

69

2 hazure

ここでif文の分岐構造を使います

字下げ(インデント)に注意しよう
字下げ1はTabキー1回押し、もしくは半角スペース2回
このプログラムは4行目と6行目に字下げ1です

3-3 第3段階「先ほどのプログラムを100回繰り返し、100連ガチャを作ろう」 (前のプログラムをコピペして書き換えよう)

```
1 import random
2 a=random.randint(1,100)
3 for i in range(100):
4     a=random.randint(1,100)
5     if(a==1):
6         print(a,'atari')
7     else:
8         print(a,'hazure')
```

2 hazure

71 hazure
22 hazure

2行目にfor文を入れて、3行目から7行目まで範囲選択してTabキーを押すと字下げ(インデント)できます。

字下げ(インデント)に注意しよう) このプログラムは

- ・3、4、6行目字下げ1
- ・5、6行目字下げ2

1 つめは全てコードを書く

2 つ目以降は前のプログラムをコピペして資料を参考に改変する
→ **どこを書き換えたら実行結果がどう変わったかを意識して考えさせる**

毎回、前のプログラムをコピペしていくことで、すべての課題のチェックを教師側でも行える。また生徒も前後のプログラムの比較をしやすくなる。

▼ プログラミングによるシミュレーション

当選確率 1% のガチャ 100 回引くと、何回の確率で当選するだろうか？ プログラミングで第 1 段階から第 6 段階までシミュレーションをしてみよう

▼ 3-1. 第 1 段階「変数 a に 1~100 の乱数を代入し、a の値を表示させよう」

```
[ ] 1 import random
    2 a=random.randint(1,100)
    3 print(a)
```

🔄 90

▼ 3-2 第 2 段階「先ほどのプログラムを 1 がでたら 'atari'、それ以外は 'hazure' が表示されるプログラムにしよう」（前のプログラムをコピーして書き換えよう）

```
[ ] 1 import random
    2 a=random.randint(1,100)
    3 if(a==1):
    4 | print(a,'atari')
    5 else:
    6 | print(a,'hazure')
```

このコードを追加したら

🔄 82 hazure

実行結果がなぜ変わったのかを意識させる

3-6のシミュレーションの結果については スプレッドシートを活用（データの活用との連携）

100連ガチャのプログラミング

当選確率1%の100連ガチャを100回分引くと何%で1個以上当選するか3つの構造のプログラムで作ろう

当選確率1%のガチャを100回引く → 100回引くと何%の確率でレアキャラをゲットできるのか？ 何回引けば必ずレアキャラをゲットできるのか？

3-3 第3段階「先ほどのプログラムを100回繰り返し、100連ガチャを作ろう」（前のプログラムをコピーして書き換えよう）

```

1 import random
2 a=random.randint(1,100)
3 if(a==1):
4     print(a,'atar!')
5 else:
6     print(a,'hazure')

```

2行目にfor文を入れて、3行目から7行目まで範囲選択してTabキーを押すと字下げ（インデント）ができます。

字下げ（インデントに注意しよう）このプログラムは

- ・3、4、6行目字下げ1
- ・5、6行目字下げ2

3-6 第6段階「前日の実行結果をotorikoiのみ表示させるようにして、100回繰り返し、実行結果をスプレッドシートに貼り付けて、100連ガチャをしたら何%の確率でゲットできるか明らかしよう」（前のプログラムをコピーして書き換えよう）

```

1 import random
2 atarikai=0
3 hazurekai=0
4 for i in range(100):
5     a=random.randint(1,100)
6     if(a==1):
7         atarikai=atarikai+1
8     else:
9         hazurekai=hazurekai+1
10 print(atarikai,hazurekai)

```

字下げ（インデントに注意しよう）

字下げ1はTabキー1回押し、もしくは半角スペース2回

3-1. 第1段階「変数aに1~100の乱数を代入し、aの値を表示させよう」

```

1 import random
2 a=random.randint(1,100)
3 print(a)

```

乱数の最小値 乱数の最大値

69

1~100の乱数が表示される

3-4 第4段階「さらに当たった回数（変数atarikai）と外れた回数（変数hazurekai）を表示させるようにしましょう。」（前のプログラムをコピーして書き換えよう）

```

1 import random
2 for i in range(100):
3     a=random.randint(1,100)
4     if(a==1):
5         print(a,'atar!')
6     else:
7         print(a,'hazure')

```

このプログラムの字下げは

- ・5、6、9行目字下げ1
- ・7、8、10、11行目字下げ2

3-6 プログラムが完成したら実行し、課題にあるスプレッドシートを開き、実行結果をスプレッドシートに貼り付けてシミュレーション結果から分かったことをまとめよう

- ①実行結果100回分をすべて範囲選択
- ②範囲選択をしたら右クリックを押して「選択した内容をコピーする」を押す。
- ③A1のセルで右クリック
- ④貼り付けを押す

3-2 第2段階「先ほどのプログラムを1がでたら'atar!'それ以外は'hazure'が表示されるプログラムにしよう」（前のプログラムをコピーして書き換えよう）

```

1 import random
2 a=random.randint(1,100)
3 print(a)

```

ここでif文の分岐構造を使います

字下げ（インデントに注意しよう）

字下げ1はTabキー1回押し、もしくは半角スペース2回

このプログラムは4行目と6行目に字下げ1です

3-5 第5段階「実行結果が縦に長すぎるため、当たった回数（変数atarikai）と外れた回数（変数hazurekai）のみ表示させるようにしましょう。」（前のプログラムをコピーして書き換えよう）

```

1 import random
2 atarikai=0
3 hazurekai=0
4 for i in range(100):
5     a=random.randint(1,100)
6     if(a==1):
7         atarikai=atarikai+1
8     else:
9         hazurekai=hazurekai+1
10 print(atarikai,hazurekai)

```

3-6 ガチャのシミュレーション結果が表示されますのでグラフから読み取れた内容を書き込もう

- ⑤当選確率を書き込む
- ⑥グラフの結果を読み取りましょう

⑦1%の100連ガチャを100回実行した結果1%の当たりのガチャが1回以上は当たる確率は何%か？分かったことに書き込もう

発表の流れ

(1) 情報Ⅰと情報Ⅱにおけるプログラミングによる
シミュレーション

(2) シミュレーションに関する事例

(3) 授業の実践と検討

①生徒自身のベースで学習を進めるプログラミング学習教材の作成

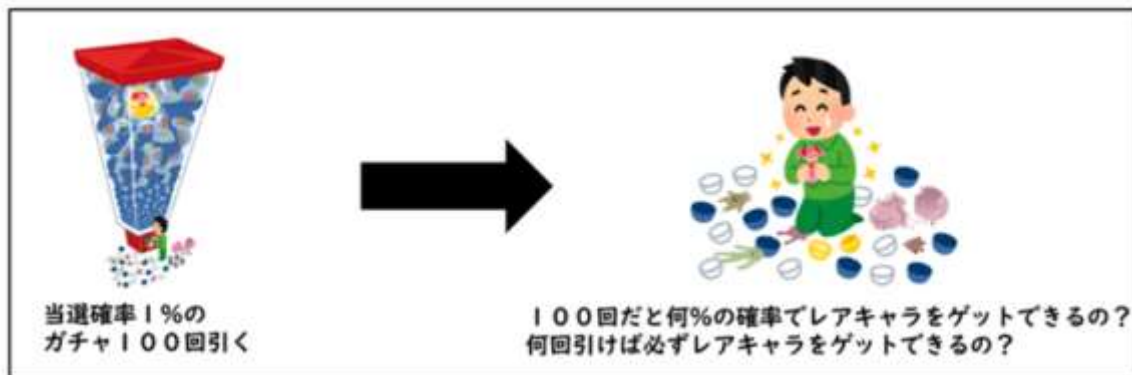
②プログラミングの題材を身近な事象に

③情報Ⅰのカリキュラム再編

生徒たちにとって身近な事象をプログラミング

100連ガチャのプログラミング

当選確率1%の100連ガチャを
100回分引くと何%で1個以上当選するか
3つの構造のプログラムで作ろう



公平な方法で発表の順番を決めよう

公平な発表順番を瞬時に決めたい
→こうしたプログラムにリストを使用する



生徒たちにとって身近な事象をプログラミング micro: bit

1. コンピュータをプログラミングで制御しよう

1-1 コインを鳴らそう

1-2 カウントアップとカウントダウン

1-3 分岐構造

演習課題1-4 (チャレンジ問題)



2. 変数と乱数を使ったプログラミング

復習: カウントアップとカウントダウン

2-1 乱数 (じゃんけんのプログラミング)

2-2 乱数 (サイコロのプログラミング)

演習課題2-3 (チャレンジ問題)

おみくじのプログラミング



3. LEDとBluetoothを使ったプログラミングで制御をしよう

3-1 信号機の制御をしよう

3-2 暗くなったら音を鳴らそう

3-3 LEDのボタンによる制御

演習課題3-4 (チャレンジ問題) BluetoothによるLED通信のプログラミング

3-4で作る通信の様子

4. コードの行数を減らして効率よくプログラミングしてみよう

4-1 自分と相手のじゃんけんを表示しよう

演習課題4-2 (チャレンジ問題) 自分と相手のじゃんけんの勝敗結果を表示しよう

演習課題4-3 (チャレンジ問題) 自分と相手のどちらかが3回勝つまでじゃんけんをしよう



① コンピュータの制御

- ・ 音の制御
- ・ カウントアップ
- ・ カウントダウン
- ・ ライトの点滅

② 変数と乱数

- ・ じゃんけん
- ・ サイコロ
- ・ おみくじ

③ Bluetoothによる制御

- ・ 自動点灯ライト
- ・ 信号機

④ 関数

- ・ じゃんけんの判定
- ・ 勝った数の判定

⑤ 時間があるクラスのみ 自由課題

生徒たちにとって身近な事象をプログラミング Python

3 公平な発表順番を提案しよう (リスト、モジュール)

[並び替えノートブック \(コピーして使用して、最後に提出\)](#)
説明資料

2 ガチャのシミュレーション (乱数、順次・分岐・反復)

[シミュレーションノートブック \(コピーして使用して、最後に提出\)](#)
説明資料
[100回ガチャの構築フォーム \(完成後は実行した結果を入力しよう\)](#)

1 プログラミングの基本を押さえよう (変数、順次・分岐・反復)

[演習ノートブック \(コピーして使用して、最後に提出\)](#)
説明資料

参考資料 (必要に応じてみよう、資料の内容をそのままコピーしてもプログラムの課題はできません)


- [episode01 変数の扱いと画面表示](#)
- [episode02 条件分岐 \(分岐構文\)](#)
- [episode03 繰り返し \(反復構文\)](#)
- [episode04 繰り返しを足していく \(反復構文\)](#)
- [episode05 リスト \(配列\)](#)

⑥⑦⑧⑨

- 発表順番決め、席替え
- ガチャのシミュレーション

5 シミュレーションのプログタイプを作ろう (シミュレーション)

画像のキャプチャの仕方
①左下のWindowsアイコンをクリック



作ったプログラミングのスクショを撮って
スライド4枚目に貼る

参考にするコードはこちら
期限内にできるところまで実装して、提出。

4 問題をモデル化しよう (モデル化)

身近な事象をモデル化しよう



どうせやるならやってみよう。シミュレーションをよう。
モデル化するためのアイデアを共有した後に、数式を真似てみよう。
シミュレーションのサンプルコードは[こちら](#)

- ⑩最後に演習課題として
シミュレーションを行った内容
をスライドにまとめる

シミュレーションする人にとって切実な課題を

発見した問題点

今遊んでいる歴史シミュレーションゲームでガチャチケットが1000枚溜まりガチャを引きたいと思っている。その時、URの3種類がどのようにゲットできるか引く前にシミュレーションしたい。

URのキャラが13枚ゲットできると、最高のスペックを発揮できるようになる。

モデル化してみよう

- ・UR(ウルトラレア)0.575% 3種類
- ・SSR(スーパースペシャルレア)14% 87種類
- ・SR(スペシャルレア)39% 30種類
- ・R(レア)46.425% 27種類

※50回引くとURが確定で1枚ゲットできる

モデルを使ってシミュレーション

```
import random
UR=0
SSR=0
SR=0
R=0
for i in range(1000):
    if (i%50==0):
        UR=UR+1
    else:
        a=random.randint(1,100000)
        if (a<=575):
            UR=UR+1
        elif (a<=14575):
            SSR=SSR+1
        elif (a<=53575):
            SR=SR+1
        else:
            R=R+1
print(UR, SSR, SR, R)
```

24 148 393 435

シミュレーション結果を用いて問題を解決する

URが24回あたったが、確定ガチャであたったのがそもそも20回なので、欲しいキャラクターの時を狙ってチケットをためておくべきだということが分かった。

しかし、確定URは42.857%と57.142%で2キャラのどちらかが出るということなので、URがあたっても欲しいキャラがたくさんゲットできることではない。

推しのコンサートでアリーナ席が獲得できるか シミュレーション

発見した問題点

この前、あるグループのコンサートを見に行った。

私は毎回アリーナ席を希望しているのだが、当選した時は毎回スタンドになってしまう。次のコンサートではどうしてもアリーナに当たりたい為、どのぐらいの倍率でアリーナ前ブロック席のチケットを獲得できるのか知りたい。

モデル化してみよう

- ・東京ドームの最大収容人数は、55,000人
- ・コンサート自体には当たったものとする
- ・アリーナ席の最大収容人数は10,000人
(前ブロックは約3,800人)
- ・どのぐらい当たったらアリーナ前ブロック席を獲得できるか知りたい。

モデルを使って シミュレーション

```
import random
a=random.randint(1,55000)
b=0
c=0
while a>1:
    a=random.randint(1,3800)
    b=b+1
print('arena')
print(b,'回目で当たりました')

arena
1617 回目で当たりました
```

シミュレーション結果を用いて問題を解決する

何回か作ったプログラミングを試したところ、運が良いと50回目に当たり、平均的に8000回は買わないとアリーナ前ブロック席が当たらないことを知った。コンサート自体当たらないこともあるので、その中で指定した席に当選することはとても難しいことだと感じた。

これからも、席はランダムなため、コンサート自体に当たったことに感謝しようと思う。

告白シミュレーション何人と付き合えるのか

発見した問題点

私の問題は、恋人がいないこと。クリスマス前までに恋人がいるようにするには今、告白したらどれくらいの確率で付き合えるのか、過去のデータをもとにシミュレーションしてみる。いままでの告白経験13回のなかで、振られた、友達のまま、付き合えた（一か月以上）（一か月以内）の4段階で判断し、毎日別のの人に告白をし、何人と付き合えるのか、シミュレーションしてみたいと思った。

モデル化してみよう

- ・振られた 38.5%
- ・友達のまま 38.5%
- ・付き合えた(一か月以内で別れた) 15.4%
- ・付き合えた(一か月以上) 7.6% ←これ希望

※365日別のの人に告白!

数値は実体験による統計です。

モデルを使ってシミュレーション

```
import random
A=0
B=0
C=0
D=0
for i in range(365):
    a=random.randint(1,1000)
    if(a<=76):
        A=A+1
    elif(a<=154):
        B=B+1
    elif(a<=385):
        C=C+1
    else:
        D=D+1
print(A,B,C,D)
```

31 37 85 212

シミュレーション結果を用いて問題を解決する

365日毎日別のの人に告白したとして、付き合える（一か月以上）のは31回となった。つまり毎日告白をしたら、31人と付き合える、1か月未満のお付き合いの場合も合わせると、69人と付き合えることが分かった。しかし振られる、友達のままが大多数を占めており、一年のうち悲しみに浸る日のほうが多そうだ。当たって砕ける。この精神を忘れずにいれば人ができない私の問題は、解決されそうだ。

発表の流れ

(1) 情報Ⅰと情報Ⅱにおけるプログラミングによる
シミュレーション

(2) シミュレーションに関する事例

(3) 授業の実践と検討

①生徒自身のベースで学習を進めるプログラミング学習教材の作成

②プログラミングの題材を身近な事象に

③情報Ⅰのカリキュラム再編

総合的な探究の時間との兼ね合いもあり データの活用をしてからプログラミングを行うようにした



先に「データの活用」を学んで、
シミュレーションした結果をグラフ
で読み取り方を理解

「プログラミング」でシミュレーシ
ョンを行い、シミュレーションの結
果をグラフ化して考察する

前半では**反復構造を使わず**にプログラミングの概念理解 後半では学んだ内容をもとにシミュレーションを行う

第3章 コンピュータとプログラミング

[前半]

	授業のテーマ	内容	使用するもの
1	コンピュータをプログラムで制御しよう	コンピュータの構成要素	micro:bit
2	じゃんけんとサイコロの動きをプログラミング	変数と乱数を使ったプログラミング	
3	交差点における信号の動きをプログラミング	Bluetoothを使ったプログラミング	
4	コードの重複を減らし効率よくプログラミング	関数を使ったプログラミング	
5	身近な問題を解決するマイクロビットの商品開発		

micro:bitでは身近な事象のプログラミングを通して概念理解を行う

[後半]

	授業のテーマ	内容	使用するもの
6	コンピュータの演算の仕組みを開発しよう	論理回路と真理値表	Python
7	アルゴリズムとプログラミングの基本	Pythonによるプログラミング	
8	公平な発表順番を考えよう	リストとモジュール	
9	プログラミングによるシミュレーション	身近な現象をモデル化してプログラミングでシミュレーション	
10	一人暮らしのモデル化とシミュレーション	家具の配置のレイアウト	

Pythonで鬼門となる反復構造や配列を学びシミュレーションに繋げる
→それでも独自のシミュレーションができた生徒は5割程度

事例325 キミのミライ発見

生徒自身のペースで進める「100連ガチャのプログラミングによるシミュレーション」～

「アルゴリズムとプログラミング」の全体設計～<https://www.wakuwaku-catch.net/jirei24325/>

これまでの実践を通して

- ・ シミュレーションは生徒の問題解決のためにする
→ 1つの指標はでき、達成した生徒の満足度は高いが
できなかった生徒の意欲が著しく低下する
- ・ 教材や指導計画の改善を通してオリジナルのシミュレーションを行える割合は増えた
→ しかしまだ6割程度
- ・ シミュレーションの事例の質を上げるには、数多くのモデル化とシミュレーションの事例の積み上げが必要
→ 先輩たちの事例を積み上げ、そのモデルを元に実践させる

伝えたいこと

先生がすべて学習活動をコントロールするのではなく
生徒自身のペースで学ぶ情報の時間を増やしませんか？

- ①自ら問題を発見し
- ②自ら解決方法を計画し
- ③自ら実際に実行し
- ④自ら評価を行い振り返る

問題解決能力を育成する授業をやってみませんか？