

2025年8月8日千葉工業大学
全国高等学校情報教育研究会
分科会D-1発表

プログラミングの授業 段階を踏んでやってみた

アサンプション国際高等学校
教諭 岡本弘之
okamoto@assumption.ed.jp
<https://okamon.jp>

1. はじめに

最初に反省から・・・

情報A/C・社会と情報では

- ・授業であまりプログラミングをやったこなかった・・・



情報 I が始まって

- ・コードを使ったプログラミングは必須
- ・プログラミングによる問題解決も体験させたい



今まで取り組んできたことと他の実践を組み合わせ
教員も生徒もハードルの低い授業を考えてみた

具体的には

今まで(5h)

アルゴリズム(2h)



LEGOマインストームによる問題解決(3h)

改善後(11h)

アルゴリズム(2h)



Scratch(2h)



コードの基本実習(2h)



シミュレーション(2h)

工夫としては

- ・Scratchから始めてみた
→プログラミングって楽しいと思ってほしい
- ・コードで短いプログラムで基本を実習した
→変数・配列、分岐・繰り返し処理は実習で学ぶ
- ・プログラミングによる問題解決はモノを動かした
→画面だけでは難しい、物が動く方が興味を引く
- ・シミュレーションするためにコードを使った
→目的を明確にしてコードを使ってみた

2. 授業の流れ

授業の範囲と目標

授業の範囲

- 「コンピュータとプログラミング」
 - 「イ. アルゴリズムとプログラミング」
 - 「モデル化とシミュレーション」

授業の目標

- 体験や実習を通じて学ばせたい
- 写経だけでなく工夫させたい
- 試行錯誤して問題解決させたい
- 共通テストでも役立つ内容に

目次	
目次	コンピュータとプログラミング
1	コンピュータのしくみ
2	コンピュータの発展的発展
3	ソフトウェアのしくみ
4	CPUとメモリ
5	CPUによる演算のしくみ
6	入出力のしくみ
7	コンピュータの発展的発展
8	コンピュータの発展的発展
9	コンピュータの発展的発展
10	コンピュータの発展的発展
11	コンピュータの発展的発展
12	コンピュータの発展的発展
13	コンピュータの発展的発展
14	コンピュータの発展的発展
15	コンピュータの発展的発展
16	コンピュータの発展的発展
17	コンピュータの発展的発展
18	コンピュータの発展的発展
19	コンピュータの発展的発展
20	コンピュータの発展的発展
21	コンピュータの発展的発展
22	コンピュータの発展的発展
23	コンピュータの発展的発展
24	コンピュータの発展的発展
25	コンピュータの発展的発展
26	コンピュータの発展的発展
27	コンピュータの発展的発展
28	コンピュータの発展的発展
29	コンピュータの発展的発展
30	コンピュータの発展的発展
31	コンピュータの発展的発展
32	コンピュータの発展的発展
33	コンピュータの発展的発展
34	コンピュータの発展的発展
35	コンピュータの発展的発展
36	コンピュータの発展的発展
37	コンピュータの発展的発展
38	コンピュータの発展的発展
39	コンピュータの発展的発展
40	コンピュータの発展的発展
41	コンピュータの発展的発展
42	コンピュータの発展的発展
43	コンピュータの発展的発展
44	コンピュータの発展的発展
45	コンピュータの発展的発展
46	コンピュータの発展的発展
47	コンピュータの発展的発展
48	コンピュータの発展的発展
49	コンピュータの発展的発展
50	コンピュータの発展的発展
51	コンピュータの発展的発展
52	コンピュータの発展的発展
53	コンピュータの発展的発展
54	コンピュータの発展的発展
55	コンピュータの発展的発展
56	コンピュータの発展的発展
57	コンピュータの発展的発展
58	コンピュータの発展的発展
59	コンピュータの発展的発展
60	コンピュータの発展的発展
61	コンピュータの発展的発展
62	コンピュータの発展的発展
63	コンピュータの発展的発展
64	コンピュータの発展的発展
65	コンピュータの発展的発展
66	コンピュータの発展的発展
67	コンピュータの発展的発展
68	コンピュータの発展的発展
69	コンピュータの発展的発展
70	コンピュータの発展的発展
71	コンピュータの発展的発展
72	コンピュータの発展的発展
73	コンピュータの発展的発展
74	コンピュータの発展的発展
75	コンピュータの発展的発展
76	コンピュータの発展的発展
77	コンピュータの発展的発展
78	コンピュータの発展的発展
79	コンピュータの発展的発展
80	コンピュータの発展的発展
81	コンピュータの発展的発展
82	コンピュータの発展的発展
83	コンピュータの発展的発展
84	コンピュータの発展的発展
85	コンピュータの発展的発展
86	コンピュータの発展的発展
87	コンピュータの発展的発展
88	コンピュータの発展的発展
89	コンピュータの発展的発展
90	コンピュータの発展的発展
91	コンピュータの発展的発展
92	コンピュータの発展的発展
93	コンピュータの発展的発展
94	コンピュータの発展的発展
95	コンピュータの発展的発展
96	コンピュータの発展的発展
97	コンピュータの発展的発展
98	コンピュータの発展的発展
99	コンピュータの発展的発展
100	コンピュータの発展的発展

日本文教出版「情報Ⅰ」教科書目次

授業の流れ(11時間)

教科書の項目	授業での実習
アルゴリズム	2.1 アルゴリズムで学ぶ(2時間) ・ゲーム形式で学んだあとアルゴリズムについて学ぶ
プログラミング①	2.2 Scratchでプログラミング(2時間) ・Scratchを使って簡単なゲームを作ってみる
プログラミング②	2.3 Pythonの基本を学ぶ(2時間) ・演算・変数・配列・条件分岐・繰り返しなど基本を実習
プログラミング③	2.4 ロボットを動かそう(3時間) ・プログラミングで問題解決をする
モデル化とシミュレーション	2.5 Pythonでシミュレーション(2時間) ・ガチャの確立、釣銭のシミュレーションをやってみる

2.1 アルゴリズムで学ぶ

ここは実習を通じて学ばせたい

教科書「情報Ⅰ」第13頁のスクリーンショット。左側には「1 アルゴリズム」の定義と例が示されています。右側には「アルゴリズムの効率性」についての説明と、迷路の例が示されています。

2.1 アルゴリズムで学ぶ

・アルゴリズム (<https://algo.jeita.or.jp/prm/2/index.html>)

迷路ゲームのスクリーンショット。迷路のスタートとゴールが示されています。右側には操作ボタンとルールの説明が示されています。

2.1 アルゴリズムで学ぶ

授業の組み立て

- ①実習:アルゴリズムをやってみる(25分)
- ②説明:教科書の内容を説明(下スライド)

①(アルゴリズム)＝何か目的を達成させるための処理手順
↓
(プログラム)＝アルゴリズムをコンピュータで処理できるようにプログラム言語で書いた

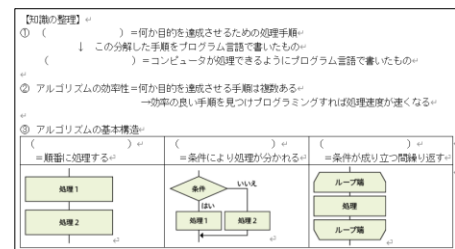
(例)アルゴリズム体系(NHKビギンズイッチ)「アルゴリズム」とは問題を解決する手順や計算方法を表す。アルゴリズム体系では、決まった動きを繰り返し、関の人とぶつからずに体検できる。(同説明より)

②アルゴリズムの効率性

・同じ目標でも複数の経路がある
→短いアルゴリズムを見つけたら効率が良い

2.1 アルゴリズムで学ぶ

②教科書の内容も説明しておく



授業プリントより

2.1 アルゴロジックで学ぶ

③課題:身近なことのアルゴリズムを考える

歯磨きをする手順を分解しよう

- ①歯ブラシと歯磨き粉を用意する
- ②歯ブラシを左手に持つ
- ③歯磨き粉を右手に持ちふたを取る
-
- ⇒10手順以上に分解する

何に役立つ?

- 例えば歯磨きをするロボットを作るとき、どのように動かすといいか、手順に分解する→この手順をもとに、コンピュータが理解できるようプログラミング言語でプログラミングする

授業プリントより

2.1 アルゴロジックで学ぶ

・生徒の振り返り

- ・スマホのアプリは膨大なプログラムで成り立っている
- ・コンピュータに手順を教えるには労力がかかる
- ・小さな工夫でも手順に差が出るといった
- ・自由な動きが必要なゲームは複雑なプログラムだろう
- ・人間のように経験や勘で動けない分、コンピュータには全ての動きをサポートしないといけない
- ・AIは賢いが、そのプログラムを作る人間はもっと賢い

・授業者の振り返り

- ・アルゴロジックの実習から入るとアルゴリズムの説明が理解させやすい

2.2 Scratchでプログラミング

①簡単なプログラミングを写経で作ってみる

【プログラムの意味】

- ①「あつち」を1から20までのランダムな数とする
- ②「わいて」を1から20までのランダムな数とする
- ③「あつち」と「わいて」の計算結果を表示し、入力待つ
- ④入力された値と「あつち」+「わいて」の計算結果が等しいか調べ、一致すれば「正解」、不一致なら「不正解」という

中野先生 ありがとう

情報Ⅰ 図解と実習

日本語教出版「情報Ⅰ 図解と実習」教科書

2.2 Scratchでプログラミング

②簡単なプログラミングを写経で作ってみる

【プログラムの意味】

- ①「あつち」を1から20までのランダムな数とする
- ②「わいて」を1から20までのランダムな数とする
- ③「あつち」と「わいて」の計算結果を表示し、入力待つ
- ④入力された値と「あつち」+「わいて」の計算結果が等しいか調べ、一致すれば「正解」、不一致なら「不正解」という

参考: 日本語教出版「情報Ⅰ 図解と実習」教科書

2.2 Scratchでプログラミング

③習ったことから改善してみる

- Ⅲ. 実習2のプログラムをベースに下のような工夫を加えよう。方法も考えよう。
- 発展例1: 正解・不正解の時に音を鳴らす
- 発展例2: 3つの数字を足す計算問題にする

④習ったことをもとに自分でプログラミングしてみる

- 【発展課題】作ってみよう!
- ① 実習1のプログラミングを参考に、「じゃんけん」のプログラムを作成してみよう。
 - ② コードは基本的には実習1と全く同じもの。※5行目のみ「2から4までの乱数」
 - ③ プログラムのスクリーンショットをClassroomの「プログラミング3」に提出する

Scratchなら写経だけでなく改善・制作も可能

2.2 Scratchでプログラミング

・生徒の振り返り

- ・思ったより簡単に自分でもゲームが作れてうれしい
- ・乱数を利用しただけで、いろいろなゲームが作れそう
- ・プログラミングは難しいと思っていたが思ったより簡単
- ・プログラミングができれば、ほしいアプリを作れそう
- ・アプリを使うときに、条件分岐・乱数と考えるようになった
- ・ゲームのガチャの「乱数調整」の意味が分かった

・授業者の振り返り

- ・Scratchは生徒にとってもとりくみやすい「かんたん」
- ・命令の理解がしやすく、間違いも発見しやすい
- ・写経だけでなく、改善・制作までできた
- ・達成感を感じやすい課題であった。「できた」「うれしい」

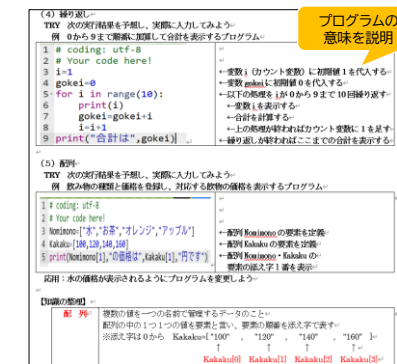
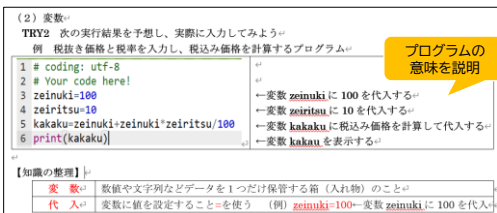
2.3 コードの基本実習

- 環境: Paiza.ioを利用 ←ブラウザ上でできる
- 言語: Python →コードを学ばせたい



2.3 コードの基本実習

- 変数・配列、分岐・繰返し処理など、短いコードを入力するプログラミングの実習を通して学ぶ



適宜スクリーンショットの提出を求めて評価の対象に

2.3 コードの基本実習

- 生徒の振り返り
 - プログラミングはすごく複雑で難しい
 - 少しでも間違えようまくいかず、頭が混乱した
 - エラーが出るともやもやした
 - プログラミングコードの意味がわかった
 - ゲームでランダムに武器が出てくる仕組みが分かった
 - Scratch よりもっと複雑なプログラムができる
 - プログラミングは難しいが、案外行けた
- 授業者の振り返り
 - あちらこちらでエラー→一人での指導に限界を感じた
 - 短いプログラムの実習にしておいてよかった..

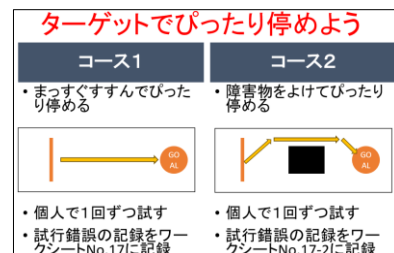
2.4 ロボットを動かそう

- 目標: プログラミングで問題解決する
- 方法: グループでLEGOマインドストームのプログラミングを考え、課題を達成する



2.4 ロボットを動かそう

- 最初は単純なコースから

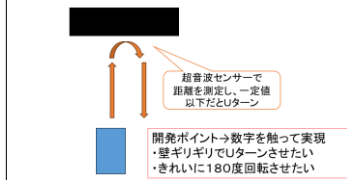


2.4 ロボットを動かそう

- 慣れたらセンサーも使ったプログラムも考える

センサー・条件分岐を使う

- 目標: 壁が近づくリターンして戻るプログラム



2.4 ロボットを動かそう

- 試行錯誤の過程を記録して改善していく

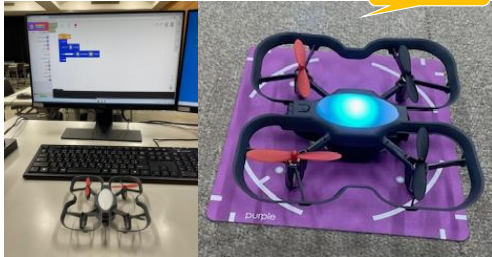
グループでの試行錯誤の記録		
試行回数	試行結果	振り返り・改善の方向性
1回	タンク 50, 50, 50	手前で止まる 数値を増やす
2回	タンク 50, 50, 60	手前で止まる 曲がった 数値を増やす 置き場所を変える
3回		
4回		

グループの他の人の結果も記録して改善案を考える

2.4 ロボットを動かそう

- 今年度はドローンでの実践を予定

DXハイスクール
ありがとう



2.4 ロボットを動かそう

- 生徒の振り返り
 - 課題を達成できたうれしい
 - 課題達成のためには他のグループの観察も重要だ
 - 失敗から試行錯誤を繰り返し、成功には時間がかかる
 - 少し数値を変えるだけで動きが変わり成功につながる
 - プログラミングはトライ＆エラーの繰り返しが重要だ
 - 物を動かすには、いろいろな要素が増え難くなる
 - 今回のようなセンサーはいろいろなところで使われている
- 授業者の振り返り
 - プログラミングによる問題解決まで実習できた
 - 物を動かすのはわかりやすく、おもしろい

2.5 Pythonでシミュレーション

- 最後はPythonでシミュレーション①

目標

- 出現確率(当選確率)1%のガチャを100回引いて、レアキャラをゲットできる確率は何%か?

方法

- シミュレーションするプログラムを作って調べる

アルゴリズム

- 乱数1~100を発生させ、1が出たときだけ「あたり」と表示(確率1/100)するプログラムを作る
- これを100回繰り返し当たりの回数を数える

鎌田先生
ありがとう



2.5 Pythonでシミュレーション

- 手順を追って確認しながら進めていく
→ 共通テストの問題もこんな感じ

STEP1

確率1%のガチャを作る

```

1 # coding: utf-8
2 # Your code here!
3 import random
4 a=random.randint(1,100) ←乱数のモジュールを読み込む
5 print(a)                 ←変数aに1~100の乱数を代入し
6 if a==1:                 変数aを表示する
7     print("あたり")      ←もし変数aの値が1と一致した
8 else:                    →"あたり"を表示する
9     print("はずれ")      →そうでなければ"はずれ"を表示する
  
```

STEP2

100回繰り返し当たりの数を数える

```

1 # coding: utf-8
2 # Your code here!
3 import random
4 b=0                      ←当たりの回数を数える変数bを初期化する
5 for i in range(100):    ←カウント変数iが100になるまで繰り返す
6     a=random.randint(1,100)
7     print(a)
8     if a==1:
9         print("あたり")
10        b=b+1             ←あたりが出たら変数bに1を加える
11    else:
12        print("はずれ")
13 print("当たりの回数は",b) ←変数b=あたりの回数を表示する
  
```

2.5 Pythonでシミュレーション

- アナログなこともあえてやってみた

STEP3 計算しよう

確率1%のガチャを100回引くことを

①10回繰り返したときの当たる確率

②100回繰り返したときの当たる確率

① プログラムを10回実行し、1回ごとに当たりの回数を記録してください。

1回目	2回目	3回目	4回目	5回目	6回目	7回目	8回目	9回目	10回目	確率

※確率のときは、「1回以上当たった回数÷10回」で計算します。

② 自分以外の9人の人に「1回以上当たった回数」を書いて情報交換し、確率を計算してください。

自分	1人目	2人目	3人目	4人目	5人目	6人目	7人目	8人目	9人目	確率

※確率のときは、「1回以上当たった回数÷100回」で計算します。

③ プログラムのスクリーンショットをクラスルームの「シミュレーション1」に提出してください。

アナログな
とり組み

2.5 Pythonでシミュレーション

- 教科書の知識も教えておく

【事前の知識】

①モデル化とシミュレーション

- ・ () ⇒ 実際の現象を抽出し、それ以外の要素や条件を省略し、単純化したモデルを作る。
- ・ { } ⇒ モデルを構える様々な要素や条件が可能な「集合」。
- ・ [] ⇒ モデルを使って現象を再現したり、モデルに要素を加減試してみる。

②モデルの種類

1) 表現の仕方による分類

{ }	⇒ (実物モデル) ⇒ 実物とほぼ同じ大きさのモデル (例) 航空機模型の再現
[]	⇒ (物理的モデル) ⇒ 物理法則に基づいたモデル (例) 分子運動
{ }	⇒ (数値モデル) ⇒ 数値を基にしたモデル (例) 統計学
[]	⇒ (概念的モデル) ⇒ 現象を捉えたモデル (例) 因果関係
{ }	⇒ (式モデル) ⇒ 現象を数式で表現したもの (例) 速度=距離÷時間

2) 表現するものの特性による分類

時間的	⇒ (動的モデル) ⇒ 時間の経過によって変化するモデル (例) 落下する物体
機能的	⇒ (静的モデル) ⇒ 時間の経過を考慮する必要のないモデル (例) 図解の要領のまとめ

不確定

不確定	⇒ (確率モデル) ⇒ 不確定要素を含むモデル (例) さいころの目、台風の経路
不確定	⇒ (決定モデル) ⇒ 不確定要素のない、論理的な現象のモデル (例) 水曜に水をやる

データ

データ	⇒ (連続モデル) ⇒ データの連続的な変化を表現するモデル (例) 気温の変化
連続性	⇒ (離散モデル) ⇒ データが離散的な状態を表現するモデル (例) 卒業の学年

授業プリントより

2.5 Pythonでシミュレーション

- 最後に、違うシミュレーションもやってみる

【発展課題】

調べよう・考えよう

目的

- 文化祭で200円の商品を60人に販売した場合に必要な100円玉の数をシミュレーションする
- 条件: 支払いは300円ちょうどの人と500円玉で支払う人のみとし割合は同じとする

手順

- 1人から60人めまで、300円と500円を支払う人をランダムに出現させ100円玉の枚数を見る
- 乱数(0~1の数をランダムに発生)を使い、0.5未満の場合は300円ちょうどを支払う人、0.5以上の場合は500円支払う人とし出出現率を50%ずつとする。

```

1 import random
2 coin = 0
3 for i in range(60):
4     r = random.random()
5     if r < 0.5:
6         coin += 1
7     else:
8         coin += 2
9 print("100円玉の枚数は", coin, "枚です")

```

→乱数のモジュールを呼び出す
→変数 coin の初期値を0にする
→変数 coin の初期値を0にする
→ランダムな数値を1から41まで作り出す
→変数 r に乱数 (0~1の間の数値) を代入
→変数 r が0.5未満の場合
→500円を支払う人とする
→変数 coin に3を加えて代入する
→変数 r が0.5以上ならば
→300円を支払う人とする
→変数 coin から2を引いて代入する
→1~60人の結果を表示する
→変数 coin が変数 coin_max より小さい場合は
変数 coin_max に変数 coin の値を代入
→最後に変数 coin_max を表示する

2.5 Pythonでシミュレーション

- 生徒の振り返りより

- 紙やペンで計算するよりも早くにシミュレーションできる
- プログラムを使えば、条件を変えることも簡単にできる
- ゲーム好きなのでガチャの確立を求めるのは楽しかった
- コツをつかんだらコードも早く打てるようになった
- 乱数を使ったシミュレーションは人によって結果が違った
- 活用すれば危険を回避し出費を抑制できる
- シミュレーションでは防災で役に立つ

- 授業者の振り返り

- 2回目なのでコード入力スムーズ→教員も生徒も慣れた
- ～するためにという目的をもってプログラミングできた

2. 共通 評価をどうするか？

- アルゴリズム＝記録＋課題＋振り返り
- プログラミング①＝スキショの提出＋振り返り
- プログラミング②＝スキショの提出＋振り返り
- プログラミング③＝試行錯誤の記録＋振り返り
- シミュレーション＝スキショ提出＋考察＋振り返り



ワークシート＝記録や考察・課題、振り返りの記入
課題の提出＝取り組み内容を確認

3. まとめ

