

D1-1

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 岡本 弘之

情報Ⅰでプログラミングが必須となり、共通テスト対策をにらんでも実習させておくことが必要である。その一方いきなりコードを使って生徒が創意工夫することは難しく、本実践では「アルゴリズム→Scratch→Pythonの基本→ロボットプログラミング→Pythonでシミュレーション」とステップを踏んで授業実践を行った。発表ではこの授業の目的や授業の流れ、生徒の反応について紹介したい。

1. はじめに

情報Ⅰの教科の目標から考えると、プログラミングの授業では問題解決まで体験させることが理想だが、コードを使ったプログラミングで問題解決を行うことはハードルが高い。本実践ではコードプログラミングの基本も学習しつつ、創意工夫や問題解決をScratchやロボット教材を使ったプログラミングに委ね、教員も生徒もハードルが低い授業を目指してみた。

2. 授業の流れ

本授業は、高校2年生情報Ⅰ（2単位）の2学期の11時間を用いて、学習指導要領の「(3) コンピュータとプログラミング」の単元の「イ. アルゴリズムとプログラミング」「ウ. モデル化とシミュレーション」の部分までを授業した。

授業案の検討にあたっては目標として

- ・体験や実習を通じて学ばせたい
- ・書き写す「写経」だけでなく工夫させたい
- ・成功失敗も含め試行錯誤し問題解決させたい
- ・共通テスト対策にも役に立つ授業としたい

の4点を考え、具体的には次の手順で実施した。

2.1 アルゴリズムで学ぶ（2時間）

アルゴリズムの説明では、体験的に理解できるようJEITA「アルゴリズム2」の実習を行った。

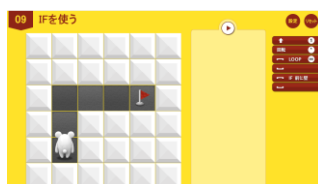


図1 アルゴリズムの画面

(<https://algo.jeita.or.jp/prm/2/index.html>)

画面上のロボットが旗をとるための動きの手順を考え、ステージをクリアしていくゲームであり、順次処理、繰り返し、条件分岐とステージが進むにつれ複雑な手順が求められる。実習後の解説で、この処理手順がアルゴリズムであり最適な手順を考えることがプログラミングで大切なこと、順次・分岐・反復の基本的な処理があることを説明した。

その後身近な作業の手順を考える課題を行い、教科書の内容を実習を通じて体験的に学習した。

2.2 Scratchでプログラミング（2時間）

最初はプログラミングに慣れてもらうため、Scratchを使って簡単なゲームを作る実習を行った。教科書に掲載されている「あっち向いてホイ」、「暗算ゲーム」を乱数や条件分岐を使ってサンプル通り制作した後、発展課題として「じゃんけんゲーム」を自分でプログラムも考えて作ることとした。Scratchであれば命令を理解しやすく、生徒は発展課題についても少しのヒントで制作することができ達成感を感じやすい課題であった。

2.3 Pythonの基本を実習で学ぶ（2時間）

次にコードによるプログラミングでは、オンライン上のプログラム環境「paiza.IO」上でPythonを使い、演算・変数・条件分岐・繰り返し・配列について6行程度の短いプログラムをテキスト通りに入力し、結果を確認する授業を行った。

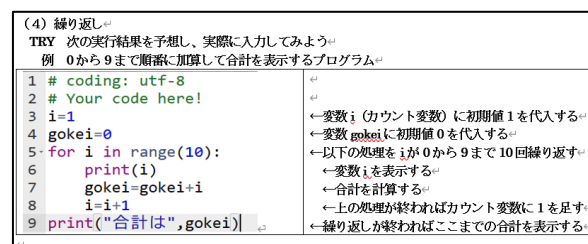


図2 ワークシートの抜粋

テキスト通りの入力で結果を確認した後、自分で計算式や数値を変更するなど、一部を変えて結果を確認することまでは取り組めた。生徒は初めてのコード入力でありがちな、入力ミスによるエラーが続出し、授業者はその対応に追われる結果となり、コードをそのまま入力するだけで手いっぱいな生徒も多く、この段階では「そのまま入力する」「一部を変える」段階までが妥当と感じた。

2.4 ロボットを動かそう（3時間）

ここまでのプログラミングでは問題解決のレベルまで取り組むことはできなかったもので、LEGOマインドストームをプログラムで動かして、グループで与えられた課題を解決する実習を行った。

課題としては「目標の位置でぴったりと止める」,
「センサーを使って障害物をよけてゴールへ到達する」という2つの課題を与えた。ここで体験させたかったのはプログラミング→実行→評価→改善という過程を繰り返し問題解決することであり、一人ではなく相談して課題をクリアすることであった。この過程をワークシートに記録させ、考察・振り返りとともに評価材料とした。

2.5 Pythonでシミュレーション（2時間）

モデル化とシミュレーションの單元では、シミュレーションをするためにプログラミングを使うという目的を明確にしたプログラミングを行った。

前時と同様に「paiza.IO」上でPythonを使い、「当選確率1%のガチャを100回ひいてレアキャラをゲットできる確率」、「模擬店で必要な釣金を調べる」という2つのシミュレーションを行った。

基本となるプログラミングはそのまま入力する形だが、数値を自分で変えたり条件を変化させる課題にも取り組んだ。2回目のコードによるプログラミングで生徒も教員も慣れてきて、今回は思った以上にスムーズに進むことができた。

3. 授業の評価

3.1 生徒の反応から

アルゴリズム、Scratchまではスムーズで、Scratchでプログラミングしたことがうまく動いたことに生徒は達成感を感じていた。Scratchでの成功経験をもとに、「これをコードでやってみよう」と授業をつなげ、生徒のモチベーションは高いままPythonの授業につなげたが、全角半角・スペースが入るなどありがちな入力ミスが相次ぎ、授業者はエラー対応に追われることになった。しかしここで生徒も教員も学習した結果、最後のシミュレーションでは入力ミスやエラー対応が激減した。生徒はそのままコードを入力するいわゆる「写経」であっても、うまく結果が出たときには達成感を感じている様子だった。ロボットを動かす授業では、画面上ではなく実際にモノが動くので、その難しさも含めて積極的に試行錯誤する姿が見られた。プログラミングしたモノが実際に動くというのは生徒の興味を高めると実感した。

3.2 生徒の振り返りから

①Scratchでプログラミング

生徒の振り返りには「思ったより簡単にゲームを作ることができた」という達成感や「がんばればもう少し複雑なゲームを作れそう」「勉強すれば自分が望むアプリを作れそう」という「自分でもゲームやアプリを作れるかもしれない」という期待感が書かれていた。プログラミングの授業の最初に「できた」、「作れそう」という経験をさせる

ことは大切であり、いきなりコードのプログラミングをさせるより良かった。

②Pythonの基本を実習で学ぶ

「少しでも間違えたらうまくいかないのでもややした」など難しいという意見もあったが、「うまくいけば楽しい」、「もう少し複雑なものもやってみたい」という意見が多くみられた。「昔習ったときには意味が分からなかったが、習って意味が分かった」と意味がわかったということや、「席替えのプログラムなら作れそう」など期待感を持ったコメントもあった。最初にありがちな入力の誤りについて注意喚起すれば、エラーも減少し、生徒の達成感が上がると考えた。

③ロボットを動かそう

「他のグループも観察し改善に生かすのがコツだ」、「ほんの少し数値を変えるだけで結果が違う」など生徒がPDCAサイクルで試行錯誤している様子が見られた。また「実際にものを動かすのは要素が多く難しい」「配膳ロボットはセンサーを生かした複雑なプログラミングがされている」と経験から、日常目にするものの仕組みについて考える振り返りも見られた。

④Pythonでシミュレーション

「手計算で行うよりも素早くシミュレーションできる」、「条件を変えることが簡単にできる」とプログラミングによるシミュレーションの利点を評価する意見や、「回数を重ねるごとにプログラミングが楽しくなった」とプログラミングに繰り返し取り組み、慣れたと感じるコメントも見られた。

4. まとめ

今回の授業実践の結果についてまとめると

- ①いきなりコードによるプログラミングを行うよりも、Scratchなど簡単なプログラミングから始める方が生徒は取り組みやすい。
 - ②コードによるプログラミングで、いわゆる「写経」でもうまくいけば達成感を感じ、それをもとにした「改善」までは簡単に取り組める。
 - ③画面上で結果が出るよりも、モノを動かす方が生徒の興味関心を高めることができる。
 - ④仕組みを体験的に知ること、身の回りの製品の仕組みについても興味を持つことができる。
- の4点にある。授業で使用した教材はWeb「情報科の授業アイデア」で公開している。→



参考文献

- (1) 黒上・堀田・村井編「情報Ⅰ図解と実習」日本文教出版、(2022)
- (2) 鎌田高德著「高校の情報Ⅰが1冊でしっかりわかる本」かんき出版(2022)