

思考力・判断力・表現力を意識 した「プログラミング」の授業

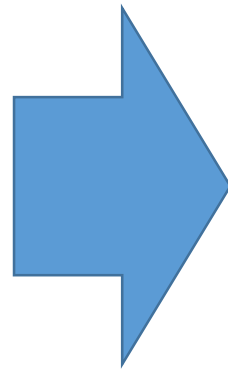
市原中央高等学校 篠田 剛史

目次

- 思考力・判断力・表現力を身につけさせるために
- プログラミングの授業の目標
- プログラミングの授業の流れ
- 大学入試に向けた対応は？

今までの情報の授業

- 教科書にある用語解説
- 2進数などの計算
- 実習
 - タイピング
 - 文書作成
 - 表計算
 - プレゼンテーション
 - プログラミング



知識・技能中心

思考力・判断力・表現力を
身につけさせるために

- 前提として知識・技術が必要
- 適切な学習形態で行う（個人で？グループで？）
- 最終的に思考力・判断力・表現力を連携させるような展開
- 継続する（”ある科目だけ”，”ある単元だけ”にならないように）
- 改善点の検討（授業の実施結果をほかの単元に活かす）

グループで行う場合の判断基準

- 個人では時間がかかってしまう場合
- 多くの意見や解答が出るのが望ましい場合
- 他人と比較することで理解につながる場合

授業の目標

プログラミングの授業の目標

- アルゴリズムを考え，表現することができる。
- 理想的なアルゴリズムは何か考えることができる。
- 実際にプログラムを記述し，評価・改善をすることができる。

授業の流れ

指導のながれ（理想）

① トレース[1~2h]（個人で）

- 流れ図の記号について意味の説明
- 選択，繰り返しの動きの確認

② アルゴリズム[3h]（グループで）

- 合計・平均・最大値・最小値を求める，探索，順位付け，ソート

③ コードを書く[3h]（個人・グループで）

- 順次構造，選択，繰り返しの表現についての説明・実践

④ 演習[4h]（個人・グループで）

- オリジナルのプログラムを作成する⇒時間があればプレゼンテーション
例) おみくじ，計算機，クイズゲーム など

① トレース

- 目的

- アルゴリズム指導のための知識・技能の習得
 - 流れ図記号のみかた
 - 構造化定理

- ゴール

- 書かれている流れ図を読むことができる
 - ↓
 - 変数や配列の値の変化を追うことができる

① トレース

- 指導方法

- あらかじめ書かれている流れ図を使って，入力から出力までの処理手順に従って結果を求める

- 主な内容

- ① 順次構造
- ② 順次構造 + 選択構造
- ③ 順次構造 + 繰り返し構造

② アルゴリズム

- 目的

- アルゴリズムとは何か知る
- アルゴリズムを考え、表現する

- ゴール

- 同じ結果を導く手順は1つではないことを体験する
- 最適な処理手順はどのようなものか見つけることができる
- 処理手順を自分で考えることができる

② アルゴリズム

- 指導方法

- 流れ図を与え，問題に答えていく形式
 - 何をする流れ図か言わない
 - 初期値や条件の理由を考えさせる
 - 流れ図を書き換えてみる
 - 最後にどのようなアルゴリズムが理想か考えさせる

- 主な内容

- 合計・平均・最大値・最小値を求める
- 線形探索，二分探索
- 順位づけ
- 並べ替え（バブルソート，セレクションソート，インサージョンソート）

③コードを書く

- 目的

- 実際に処理手順を記述して，コンピュータ上で動かしてみる。

- ゴール

- 正確に処理手順を記述することができる
- エラーに対応することができる

③コードを書く

- 指導方法

- 言語の解説 → 簡単なプログラムを提示し，そのまま記述させる
- 練習問題 → 完成されていないプログラムを完成させる

- 内容

- 言語の解説
 - 変数の宣言，表示，代入，演算，条件分岐，繰り返しについての記述方法の解説
- 練習問題
 - 流れ図や処理内容から記述方法を考える
 - 主に代入，演算，条件分岐や繰り返しの条件について問う

④演習

- 目的
 - これまでのまとめ
 - プログラム開発の流れを実践してみる
- ゴール
 - 処理手順を流れ図で表現することができる
 - 正確に処理手順を記述することができる
 - エラーや処理手順の変更に対応することができる

④演習

- 指導方法

- テーマ例をいくつか与え，オリジナルのプログラムを作成させる
- 可能であれば生徒自身に言語を選択させてもよい

- 内容

- 流れ図の作成

- 処理手順を箇条書きでまとめさせ，流れ図で表現させる

- コードを書く

- 処理手順通りに書いていく中で，エラーや処理手順の変更に対応する

- 評価

- プログラムの作成を通してわかったこと，思ったことなどまとめる
- 他の人に実行してもらって，フィードバックをもらう→時間があれば修正

大学入試に向けた対応

大学入試（主に共通テスト）に向けて

- ①コードを書くことよりもアルゴリズムを考える。
 - ⇒書き方にこだわる必要はない。
 - 実行さえできれば，どのような言語を扱っていても問題ない。

- ②大学入試（DNCL）のために授業内では特別な対応は必要ない。
 - ⇒生徒の実態，環境にあった言語を選べば良いと考える。
 - （大学入試のためのプログラミングではない。）



処理手順を自ら考え，様々な形で表現することができれば
プログラミングに関しては対応可能であると考える。

ご清聴ありがとうございました