

特集：各科目で育む情報の科学的見方・考え方②

情報Bにおける情報の科学的見方・考え方

～問題解決能力を育む授業実践～

立教池袋中学校・高等学校教諭 山口 弘泰

『044新版情報B』使用

1. はじめに

本校では現在、中2から高1までの合計5単位で情報の授業を展開しています。生徒の進路等の事情から構成を次の2つに分けています。

- ① 基本的なアプリケーションの操作の習得
- ② コンピュータのしくみの理解と問題解決への役立て方の習得

①は、系列大学で補講を受けることなく、授業でコンピュータを使える力を養うことが狙いです。本校では高校、大学で卒業研究・論文の作成および口頭発表があり、ワープロ・図形描画・表計算・プレゼンテーションソフトが必要になります。中学校の段階でキーボードでの文字入力からはじめて表計算で簡単なマクロを組むまでを終えます。授業時間内ですべてを教えることは不可能なので、昼休み・放課後とチューターを置いてコンピュータ室を生徒に開放し課題等は授業外で行わせます。

②は、学年が進むにつれ重要性が増してきます。学習内容が複雑になるとソフトの操作を覚えただけでは解決できない課題が出てきます。単純な例としては大きな数の素因数分解などです。このような問題の解決にコンピュータを用いるにはプログラムを組む必要があります。プログラムが正しく組めれば正しい答が出

力されるとしてしまいがちですが、間違った答が出力されることもあります。

これはコンピュータのしくみに起因する現象であり、それらを学ぶことでコンピュータの限界を知ることができます。そうすれば間違いを回避するようなアプローチを取れると同時に、出力結果を検証する姿勢が身につき、これこそが科学的な態度ではないかと考えます。

そこで、この分野を扱うために、高校では情報Bを選択しました。

表1 授業カリキュラム

単元	項目	内容	教材・課題
2進数	変換	2進数⇄10進数・2進数⇄16進数	2進数の小数表示
	小数の扱い	変換方法(誤差の発生)	2進数の8ビットの補数表示と計算(10進数と対比)
	補数	足し算で引き算を行う	
コンピュータとは	数値の表現	固定小数点表示・浮動小数点表示	
	ビットとバイト	組み合わせ数	
コンピュータのしくみとデータの数値化	5大機能	入力⇄処理⇄出力の繰り返し	
	発色のしくみ	RGB・ビット数と発色数の関係	
	数値化のメリット	アナログとデジタルデータ(誤差の発生)	
	データの数値化と加工	ビット数と再現性	
数値化	・文字	演算(足し算)によるデータ変化	シーザ暗号
	・画像	加工の手順(標本化・量子化・符号化)(誤差の発生)	
	・音	加工の手順(標本化・量子化・符号化)	白黒画像の明るさの変化
			MIDIによる音階の変化
演算のしくみ	2進数を用いる利点	データの重みと補正・伝達性	
	四則演算と足し算の関係	足し算のみで四則演算を行う	
	2進数の足し算	1桁の足し算⇄4通りの組み合わせ	
	機械による足し算	ON/OFFと2進数との関係	
	論理回路	基本論理回路	論理回路シミュレータ
		組み合わせ回路(NAND・NOR)	
	半加算器	二桁目(和)の出力	
		二桁目(繰り上がり)の出力	
	実際の回路	基本論理回路のリレーによる置き換え	リレーで作成した論理回路
		リレーによる半加算器	リレーによる半加算器
	基本論理回路のトランジスタによる置き換え	トランジスタで作成した論理回路	
全加算器	繰り上がりの考慮		
	メモリのしくみ	順序回路(フリップフロップ)	論理回路シミュレータ
コンピュータ発達史	リレー⇄真空管⇄半導体⇄集積回路	論理回路シミュレータ	
コンピュータの動作	逐次実行型コンピュータ	スタアドプログラム方式 プログラム動作の詳細	
アルゴリズムとプログラミング(シミュレーション)	アルゴリズムの表記法	フローチャート	
	アルゴリズムの基本形	順次構造・選択構造・繰り返し構造	
	構造化プログラミング	処理のモジュール化(入り口1つ・出口1つ)	
	プログラムを構成する要素	定数・変数・演算子・関数	Tiny BASIC
	変数への値の入力	変数定義とメモリ内での動作	
	順次構造	式の記述と値の出力(print文)	方程式
	選択構造	プログラム外からの値入力(input文)	
		#文の記述について	ABS関数
		#文のネスト(elseif)	判別式と解の個数
	繰り返し構造	while文 とカウンタ	数列・剰余・INT関数・米粒
数の制限がない繰り返し 回数に制限がある繰り返し	for文	関数の描画・ネイピア数 モンテカルロ法	
反復の入れ子(ネスト)	反復を用いた値の自動入力	素数判定・素因数分解	
	while文 と for文の組み合わせ	根号の括り出し	
		ソート	
モジュール化	作成した処理単位のモジュール化		

2. 実施内容の概略

高校でのカリキュラムは大きく分けて2つあります(表1)。

- (1) コンピュータのしくみとデータの数値化
- (2) アルゴリズムとプログラミング

(1)については、データが数値化され演算(主に足し算)によって加工(処理)されていく方法やしくみの説明に重点を置き、(2)については問題解決の手段としてのパソコン使用法を中心に学びます。ここで、各項目の要点を説明します((p.)は、『044新版情報B』のページ数です)。

(1) コンピュータのしくみとデータの数値化

a) n進数の変換

ここでは2進数⇔10進数や2進数⇔16進数への変換を学びます(p.29)。また、マイナスの数が補数として表現され、引き算を足し算に置き換えることで四則演算すべてが足し算で行われること(p.31)、小数以下の数値は10進数では有限桁でも2進数にすると循環小数になってしまう数があり誤差発生の一因になることを押さえます。また、数値の表現で固定小数点表示と浮動小数点表示があり(p.32)、後者においては正規化に伴い桁落ちや情報落ちの誤差が発生する可能性があることも伝えます。

b) 情報の単位

情報の基本単位のビット・バイトを学び、またnビットでの組み合わせ数を考えさせ(p.25)、ビット数が増えれば表現できる状態の数も増えるが、より細かく表現するにはデータ量が増える<トレードオフの関係>に繋がります(p.18)。

c) 数値化

コンピュータに取り込めるデータは数値なので、元のデータを一定の決まりで数値に置き換える必要がある点、また演算により加工できる点を押さえます。題材として扱うデータは次の3つで、演算によるデータの加工については事例を単純化した自作教材を用いて理解を深めるようにしています。

- ・文字(p.33)……アルファベットを入れたファイルのダンプを取り、文字と文字コードが一致していることを確認します。その後、コードに一定の値を加えて暗号化するシーザ暗号を例に演算でデータが加工されることを押さえます。
- ・画像(p.34)……画像が標本化⇒量子化⇒符号化の手順を経て数値化されていること、その間に誤差が発生することを伝えます。また、教科書の写真でデータ量と再現性の関係について触れます。データの演算による加工については、拡大した45画素の白黒映像を用い、各画素の明度を増減させて画像を変化させます。これがグラフィックソフトの処理と同様であることを確認させます。
- ・音……数値化については画像と同様な手順であることを実例を踏まえ伝えます。データの演算による加工については、MIDI形式で音階のみを取り出し、そこに何音階か加えたときの変化を実際に試聴させ確認させます。

個々の例を学ぶ際に、ファイルのダンプを取ってすべてのデータが数値になっていることや、2進数を用いることで、歪んだデータを修正しやすい利点などにも触れます(p.23)。

以上のようにデータの加工が演算によって行われていることを確認した後、演算(計算)はどのようなしくみにより実行されているかに話を進めます。

d) 論理回路と加算器

1桁の足し算ができれば桁が増えても同じ繰り返しなので、2進数1桁の足し算(4通り)の回路による実現に内容を絞ります(p.27)。

コンピュータを構成する基本的な3つの回路と動作を伝えた後、「論理回路のシミュレーション」(実教出版Webより配布)を利用して、基本回路や組み合わせ回路の入出力を確かめます。

その後、前述の足し算を1桁目(和)と2桁目(桁上げ)に分け、それぞれの回路を作成し合成します。

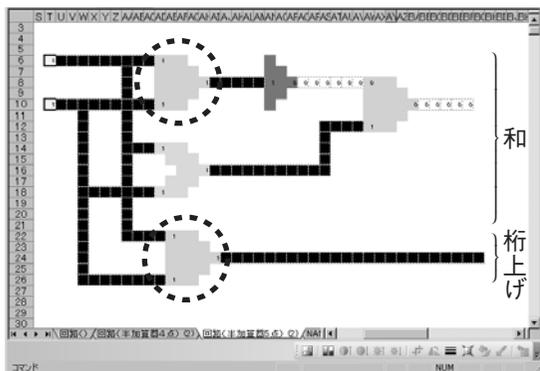


図1 シム回路

次に上図の円内のAND回路が共通に使えることに着目させ、4つの回路で構成される半加算器に繋げます (p.27)。さらに下位からの繰上げを考慮した全加算器を組み、論理回路による足し算のしくみ全体を理解させます。また、順序回路(フリップフロップ回路)がメモリ(記憶素子)として動作する様子を確認します。

e) 実物の回路

半加算器を作り検証します。具体的には素子にリレーを用いたAND・OR・NOT回路を単品で動作させた後、半加算器を組んで1桁の足し算を行わせます。素子をトランジスタに換えても同様に動作することを確認します。この流れからコンピュータ発達の歴史(素子の変遷)を伝えます。

f) コンピュータの動作

これまで見てきた加算器やメモリの動作が他の装置と連携していかにプログラムを実行していくかを学びます (p.39)。加算を例題とし、アセンブラのニーモニックと日本語で書いた命令を対比させ理解を深めます。

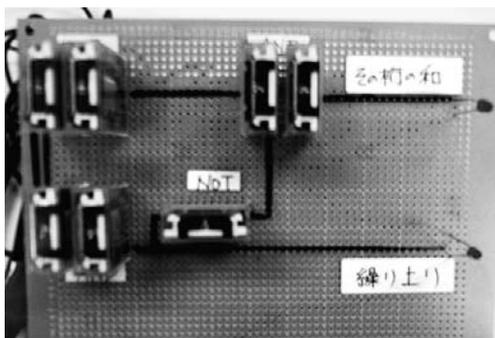


図2 半加算器

(2) アルゴリズムとプログラミング

a) アルゴリズム

最初は日常生活の中の簡単な例を複数出して単純な作業に分解し書き出します。すると順番に行えば終了する単純な処理、条件によって結果が変化処理、同様な手順が繰り返し発生する処理が出てきます。これを元にわれわれが行う作業手順は順次構造・選択構造・繰り返し構造の組み合わせで表現できることを伝えます (p.41)。また、一度組み立てた処理を別のプログラムでも再利用するため、入口1つ出口1つのモジュールとする構造化プログラミングについても触れます。

b) プログラミング・シミュレーション (p.64)

アルゴリズムを簡潔に表現できるフリーソフトのTiny Basic for Windowsを利用します。基本的な手順として流れ図を参考に基礎となる簡潔なプログラムを習得させ、次にそれを多少変化させた発展的な課題を解かせます。実行結果に不備があればデバッグして原因を究明します。また、シミュレーションもプログラミングの課題として解かせています。このようにプログラムを与えるだけでなく、コンピュータを利用した問題解決の方法を自ら摸索し理解させます。以下にプログラム課題の一部を示します(「**内太字**」は基本の例題・下線部付はシミュレーションの課題です)。

■準備
Tiny Basicの扱い、変数の定義、コマンドによる数値・文字の出力、簡単な計算等、基本的な扱いを伝えます。

■順次構造 (p.43)
「方程式」「連立方程式」
一次方程式の解法をプログラミングし、発展課題として連立方程式を解かせます。

■選択構造 (p.44)
◎ 2つの分岐
「正負の判定」「ABS関数」
入力された値により正負を出力するプログラムを応用し、絶対値を返すABS関数を作ります。

◎ 3つ以上の分岐
「じゃんけん」「判別式」
前は2つの処理でしたが、3つ以上の処理に用いるELSE IFの記述方法について伝えます。

■繰り返し構造

条件によりループ内の処理が何回繰り返されるかわからないものと、一定の回数または範囲が最初から決まっているものがあるので、それぞれに適した構文を選択させることは大切です。

◎単純な繰り返し

「カウンタによる数列の出力」

Print文と組み合わせると1, 2, 3など単純な数列から奇数・偶数、等差数列・等比数列と課題を進めます。また、値を加えていくだけでなく引く例も扱います。

「剰余」「INT関数」

2数a, bにおいて $a \div b$ の剰余は、 $a - b$ の値が0以上の間引き続け、最終的にaに残ったものであることを確認します。それを応用してaが有限小数のとき、 $b=1$ とすれば剰余は小数部分になるので、元のaから引けば値は小数切り捨ての整数になることに気づかせます。

「日々倍になる米粒の和」(数列の和)

シミュレーション課題として、1粒からはじめて日々倍の米粒を30日間受け取りすべて貯めた場合の総数を計算させます。日々の米粒数を棒グラフにし、各グラフの合計(面積)が米粒の総数であることを理解させ、積分の学びに繋がります。

「関数の描画」

ある関数の(x, y)座標を計算し、グラフ画面に出力させ関数の概観を出力します。

「ネイピア数」

関数としてネイピア数の公式 $(1 + 1/n)^n$ を与え、nを大きく増やした場合、値がどのように変化するかを確認します。この値が自然対数の底であることも触れておきます。

「1/4円による円周率の算出」

モンテカルロ法を用いて円の面積と円周率を計算させます。これらの計算はコンピュータなら簡単に答えが出せること、結果が予測外になるものもあることから模擬実験の必要性を確認します。

◎繰り返しの入れ子 (p.50)

問題(課題)の中には1重のループでは解決できないものもあります。そこで1重ループのプログラムを土台として2重ループに進みます。

「素数判定」

ループを用いて、与えられた数nを $2 \sim n-1$ で割ったときの余りを求め、0が出なければ素数と判定します。

「1000までの素数の出力」

一定の範囲から素数を抜き出して出力する方法を考えさせます。ここでnを入力する代わりに、先のプログラムの外側を3~1000の自然数を次々と与えるループで囲み、素数と判定された数を順に出力させればよいことに気づかせます。

「素因数分解」

素数判定では同じ数で複数回割ることはしませんが、素因数分解では割り切れる間何度も割ります。外側のループで割る数を1つつ増やしながらかけることは同じですが、内側のループを割り切れる間除算を繰り返すために利用します。

「根号外し」

先のプログラムを応用し、外側のループから与えられた数の二乗で割ればよいことに気づかせます。

3. おわりに

私は授業において、逐次実行型のコンピュータがどんなしくみで何をしているかという基本的な部分と、どうすれば何ができるのかということについてのプログラミングをしっかりと把握させることが大切だと考えています。これらの基礎が身についていけば日々変化する技術にも柔軟に対応でき、この先もコンピュータを自分の課題解決に利用できるでしょう。

情報Bの内容は難しい箇所が多いので生徒の理解を深めるために、以下の点を工夫しています。

- ・教材をなるべく視覚化する
- ・抽象的、難解な例題を避け、理解しやすい身近な題材を用いる
- ・内容をパターンごとに分け、スモールステップで理解させていく
- ・多少ひねれば応用が利くような例題を用いる
- ・将来への学びに繋がるような例題を用いる

手探りの状態から情報の授業をはじめて6年経ちました。自分の中で何を伝えるかがようやく見えはじめ、受け取る生徒側からも情報処理技術者試験に合格する者が出つつあります。

今後多くの先生方のお知恵を拝借しつつ、授業の改善に努めようと思います。