

# 百マス計算ソフトの開発と実践

せたな町立若松小学校

教頭 佐々木 朗

## 1. はじめに

道研での研修の成果により、私は簡単なものであれば、エクセルの VBA (visual basic for application) を使って、プログラムを作成することができるようになった。今回は、百マス計算ソフトの開発とその実践について述べる。

(図 1)

## 2. 百マスの効用

今年度、3・4 年生の複式学級 (3 年生男子 4 名・4 年生女子 1 名) を担任している。百マス計算について、足し算、かけ算、引き算について毎日順番に、算数の時間に実施した。また毎日家庭用に一枚ずつ持たせて家での課題とした。

百マス計算の効用については、様々な論議があるようであるが、私は前任校でも実践し、その効果は非常に高いと思っている。

一つは集中力の育成である。一日 1 分～3 分程度であるが、全ての神経を集中する時間を持つことができる。もう一つは、成果が自分ではっきりわかることである。このことについては、もう少し詳しく触れる。

子どもたちにとっては、「今日もやるの？」

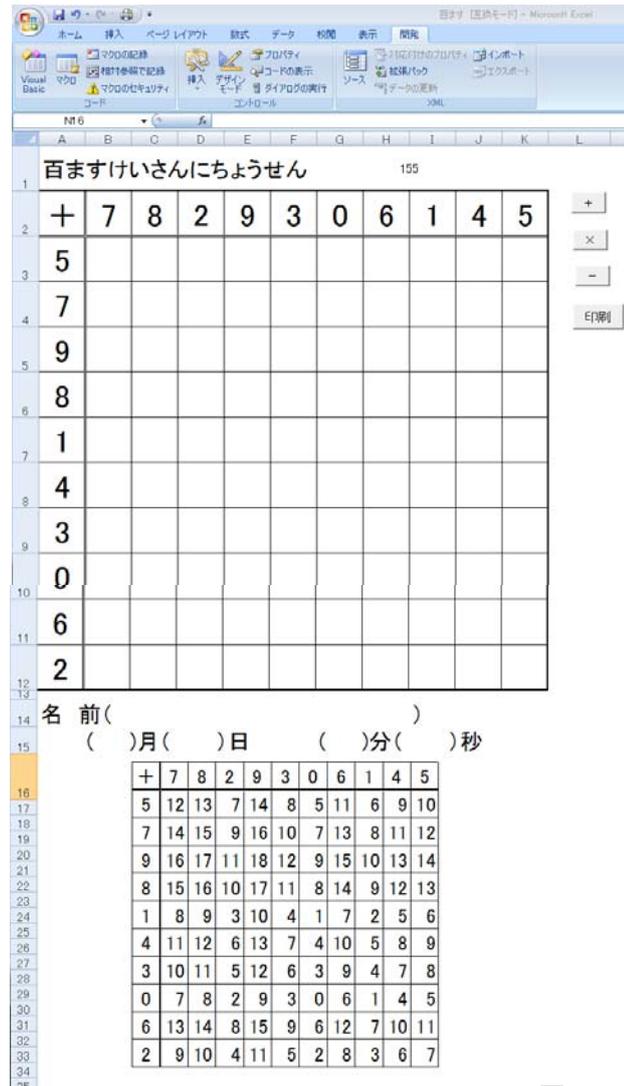


図 1

という声もあるが、ストップウォッチを押すと真剣に取り組んでいる毎日である。

次ページグラフは、3 年生の児童の「たし算」この 1 年間のかかった時間の推移である。太い線の区切りが 1 分となる。(図 2, 図 3)

### 3. 百マスの計算効果について

私の実践では、  
 ①全ての百マス計算において、時間を計測している。  
 ②数字は読める程度であれば、可とした。  
 ③できあがったら、「できました。」と言い、そのタイムを告げる。  
 ④私が解答をチェックし、間違いがあれば、1問につき5秒を加える。  
 ⑤結果については、子どもたち自らがグラフにつける。

私は、前任校と、本校で実践して、一番大切にしているのは、⑤の自らの結果を記し、その上達を知ることができることである。どの子も最初は5分近くかかっていたが、初頭効果でグッとタイムが速くなり、その後も、少しずつ時間を縮めていくことができる。

私は、子どもたちに2つの目標を持たせた。一つは、2分を切る壁、もう一つは、1分40秒の壁（100秒、つまり、1秒1問）である。一年間を通して、実践し、

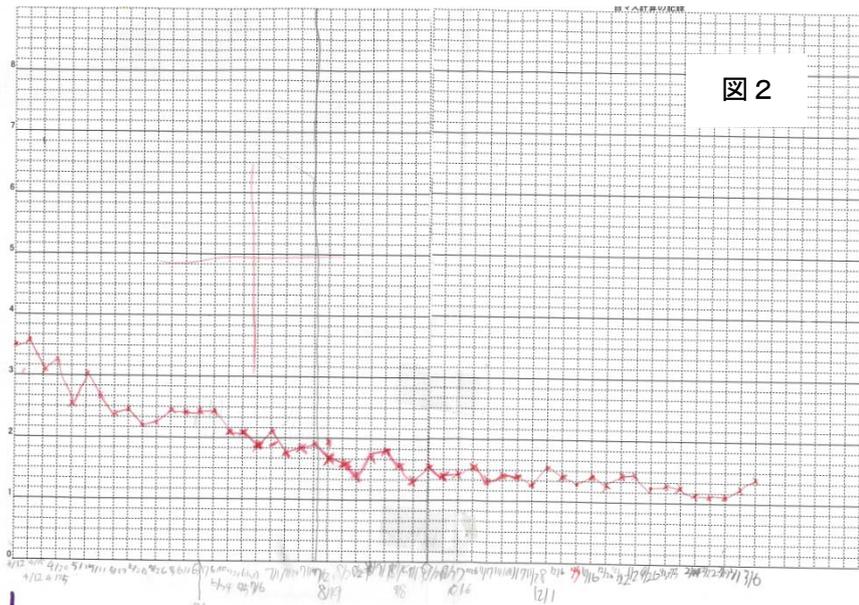


図 2



図 3

この2つの壁をクリアした児童も出てきた。

もう少し、継続すると、さらなる時間短縮が期待されるが、もうすぐ3学期も終了を迎えそうな時期になってしまった。

### 4. プログラムについて

プログラムは、エクセルを用いて、乱数発生及び表示にVBAを使っている。(図4)

①ボタンにより、「足し算」、「引き算」、「かけ算」のいずれかを選択する。

②乱数制御ルーチンにより、乱数をシャッ

```

[Module1 (コード)]
入(D) 書式(O) デバッガ(D) 実行(R) ツール(T) アドイン(A) ウィンドウ(W) ヘルプ(H)
11行, 23桁
[General] 足し算
解答.Cells(2, i + 1) = 作業.Cells(i + 10, 2)
Next i
For i = 1 To 10
For j = 1 To 10
解答.Cells(i + 2, j + 1) = 解答.Cells(i + 2, 1) + 解答.Cells(2, j + 1)
Next j
Next i
End Sub
Sub かけ算()
Call 乱数制御
Set 問題 = Worksheets("問題用紙")
Set 解答 = Worksheets("解答")
Set 作業 = Worksheets("作業用")
問題.Cells(1, 8) = 問題.Cells(1, 8) + 1
Dim i As Integer
Dim j As Integer
For i = 1 To 20
作業.Cells(i, 1) = Rnd()
Next i
問題.Cells(2, 1) = "×"
解答.Cells(2, 1) = "×"
For i = 1 To 10
問題.Cells(i + 2, 1) = 作業.Cells(i, 2)
問題.Cells(2, i + 1) = 作業.Cells(i + 10, 2)
解答.Cells(i + 2, 1) = 作業.Cells(i, 2)
解答.Cells(2, i + 1) = 作業.Cells(i + 10, 2)
Next i
For i = 1 To 10
For j = 1 To 10
解答.Cells(i + 2, j + 1) = 解答.Cells(i + 2, 1) * 解答.Cells(2, j + 1)
Next j
Next i
End Sub
Sub 引き算()
Call 乱数制御
Set 問題 = Worksheets("問題用紙")
Set 解答 = Worksheets("解答")
Set 作業 = Worksheets("作業用")
問題.Cells(1, 8) = 問題.Cells(1, 8) + 1
Dim i As Integer
Dim j As Integer
For i = 1 To 20
作業.Cells(i, 1) = Rnd()
Next i
問題.Cells(2, 1) = "-"
解答.Cells(2, 1) = "-"
For i = 1 To 10
問題.Cells(i + 2, 1) = 作業.Cells(i, 2)
問題.Cells(2, i + 1) = 作業.Cells(i + 10, 2) + 10
解答.Cells(i + 2, 1) = 作業.Cells(i, 2)
解答.Cells(2, i + 1) = 作業.Cells(i + 10, 2) + 10
Next i
For i = 1 To 10
For j = 1 To 10
解答.Cells(i + 2, j + 1) = 解答.Cells(2, j + 1) - 解答.Cells(i + 2, 1)
Next j
Next i
End Sub
Sub 印刷()
Set 問題 = Worksheets("問題用紙")
問題.Range("a1:k34").PrintOut
End Sub
Sub 乱数制御()
For i = 1 To Second(Now()) + Minute(Now()) + Day(Now())
e = Rnd()
Next i
End Sub

```

図 4

D23		
	A	B
1	0.401374	5
2	0.27828	7
3	0.160442	9
4	0.162822	8
5	0.646587	1
6	0.410073	4
7	0.412767	3
8	0.71273	0
9	0.326206	6
10	0.633179	2
11	0.207561	7
12	0.186014	8
13	0.583359	2
14	0.080715	9
15	0.457971	3
16	0.90573	0
17	0.261368	6
18	0.785212	1
19	0.378903	4
20	0.289665	5
21		
22		
23		
24		
25		

図 5

数値が表示された。これを回避するために、日付関数から、日付（1～31）、分（0～59）、秒（0～59）を加算し、その回数だけ、FOR NEXT で乱数を発生させて、

ルする。

初期化を行った。

※最初はこのアルゴリズムがなかった。そうするとソフトを立ち上げるたびに、同じ

③作業シートに、乱数を20個発生させる。（図4）

④ 10 個ずつについて RANK 関数を用いて、順位をつける。この 0 から 9 までの 10 個の数値が縦と横の数になる。(引き算については、横の数値に 10 を加えた数とし、10 から 19 までを発生させる)

⑤ 関数により表示シートに④の数値を配置する。

⑥ 解答シートも数値を配置し、関数で解答を作成する。

⑦ 解答を図形として縮小して、問題シートの下につける。

⑧ 印刷ボタンをクリックすると、印刷される。

このプログラムを開発して、毎回違った数値を配置できること(クラスで全員異なる問題とすることもできる)。解答が下にあるので、採点しやすいことなどが工夫したところである。

このプログラムは、

<http://www.edu-hakodate.jp/sasaki/kyozai/100masu/>

から、ダウンロード可能である。是非使ってみてほしい。