

Tiny Basic for Windows について

竹内 照雄

Tiny Basic for Windows は Windows XP, Windows Vista, Windows 7 上で動く BASIC インタプリターです。Windows 上で手軽に、簡単に使えるフリーソフトウェアの BASIC として開発を進めています。Tiny の名の通り、余り大袈裟なことは出来ませんが、言語やアルゴリズムの勉強や個人的に一寸した計算のためには十分の機能が揃っています。フリーソフトウェアとして公開していますので、自由にコピーして使用して下さい。ここではその概要を紹介します。尚、本文の記述は 2010 年 1 月の Ver.1.2 に基づいています。Tiny Basic for Windows の最新版は

<http://www2.cc.niigata-u.ac.jp/~takeuchi/tbasic/index.html>

にありますので、必要な場合はそこからダウンロードして下さい。

目次

1. Tiny Basic 開発の考え方
2. Tiny Basic の特徴
3. Tiny Basic の機能と使い方
4. Tiny Basic のプログラム例
5. 最後に

1. Tiny Basic for Windows 開発の考え方

Tiny Basic for Windows 開発の基本的考え方を説明します。以下 TBasic と略記します。

TBasic は初心者向けの教育用 BASIC としてスタートしました。BASIC は古い歴史を持つ言語で、少し時代遅れになった感じもしますが、コンピューターに本格的に触れようとする人にとって、まだ学ぶ価値があります。Visual Basic は別として、現在ではこのような Basic で応用プログラムを作ることは少ないかも知れませんが、種々のマクロや、データベース等の簡易言語でのプログラミングと共通するものが沢山あります。また学生の立場からすると、数学や物理等の数値実験を行うためのプログラミング言語としては手軽なものです。このようなことから、今でも多くの人々にとって BASIC は計算機言語の第 1 歩としての役割を果たすと考えられます。

多くの人々に手軽に BASIC を使って見て貰うためには、簡単で使いやすい、しかも無料で使えるものがが必要です。TBasic はこの役割の一端を担うものを目指して開発しています。また、Help を活用することで、BASIC が自習できるようにと考えています。

TBasic は元々 PC9801 の MS-DOS 上で動く、N88-BASIC のサブセットをフリーで提供するために開発した BASIC が始まりです。

現在の TBasic はそれを Windows 用に移植したものが元になっています。しかし、元々 DOS 用に使われた N88-BASIC を Windows 用に忠実に移植するのは、不自然です。そこで移植に当たっては、Windows の特徴を生かした言語になるように仕様を変更しています。その変更は現在使われている Windows の標準的な使用法に沿ったものです。その結果、N88-BASIC とはいくつかの面で異なったものになりましたが、Windows のユーザーとしては、むしろ使いやすくなったと思います。また、TBasic は入門用の言語として位置付けていますから、ある程度進んだプログラミングを行うようになると、当然、別の更に本格的言語（Visual Basic 等）を使うようになるでしょう。この場合、TBasic を学んだことがプログラムの勉強として無駄にならないよう考慮しています。

現在のコンピューター環境は急激に進歩していますが、TBasic は、入門用言語としてその流れに沿って柔軟に対応していきと思っています。

2. Tiny Basic for Windows の特徴

Windows 上で動作するフリーの BASIC としては、(仮称)十進 BASIC や N88 互換 BASIC など優れたものが既にあります。しかし、TBasic はそれらと異なる視点で作られていて、次のような特徴を持っています。

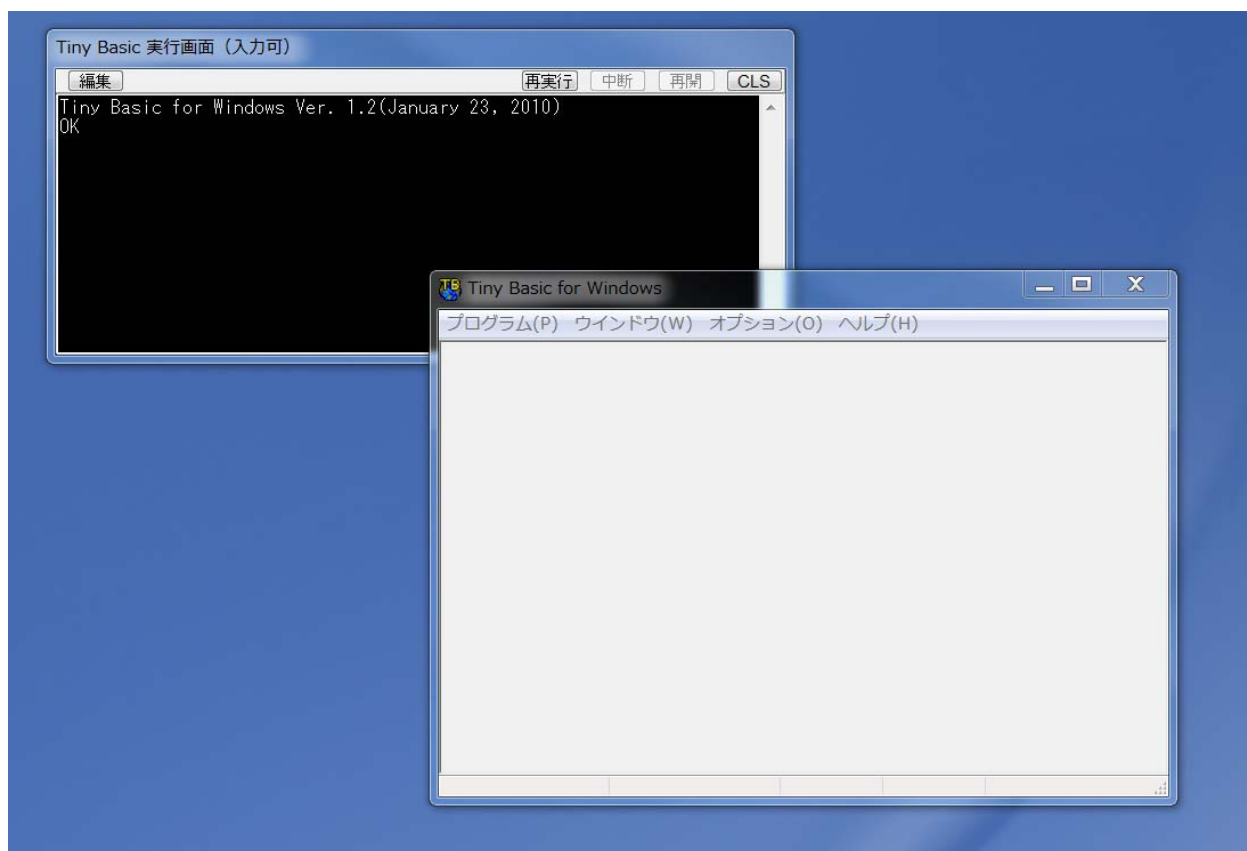
1. Tiny Basic for Windows はフリーソフトです。
 - ・無料で使用できます。
2. インストール, アンインストールが簡単です。
 - ・配布ファイルを適当な場所に解凍するだけで使用できます。
 - ・PC から完全にアンインストールが簡単に可能です。
3. 色々な使用形態 に合わせて環境設定が可能です。
 - ・個人保有の PC での環境 (レジストリ登録可能な環境)
 - ・学校などでの共有 PC での一般ユーザーや制限ユーザー環境
4. USB フラッシュメモリに保存しても使えます。
 - ・実行ファイル TBasic.exe のみでも動作し, 他のファイルは必要ありません。
5. Windows の標準に沿っています。
 - ・Windows の標準に沿っていますので, 使い方を覚えるのが簡単です。
6. Microsoft 系の BASIC 言語です。
 - ・N88-BASIC, Quick Basic 等からの移行が簡単です。
 - ・Visual Basic 等へもスムーズに進めます。
7. BASIC の機能が 1 通り揃っています
 - ・比較的高速に動作しますので, 簡単に使える Tool として活用することができます。
 - ・グラフィック機能も比較的充実しています。
8. Visual Basic への機能を意識しています。
 - ・Visual Basic を意識した機能を取り入れています。
9. ダイレクトモードも使えます。
 - ・本格的なプログラムを書かないで電卓風に使用することも出来ます。
10. ユニコード対応です。
 - ・JIS, Shift-JIS, EUC などの日本語や UTF-16 などユニコードの文書処理が可能です。

3. Tiny Basic for Windows の機能

Tiny Basic for Windows の機能とその使い方を簡単に説明します。

3. 1 起動

TBasic は編集画面、実行画面、グラフ画面、コントロール画面の4つの画面から構成されますが、起動時は編集画面、実行画面のみが立ち上がります。次は起動直後の画面です。

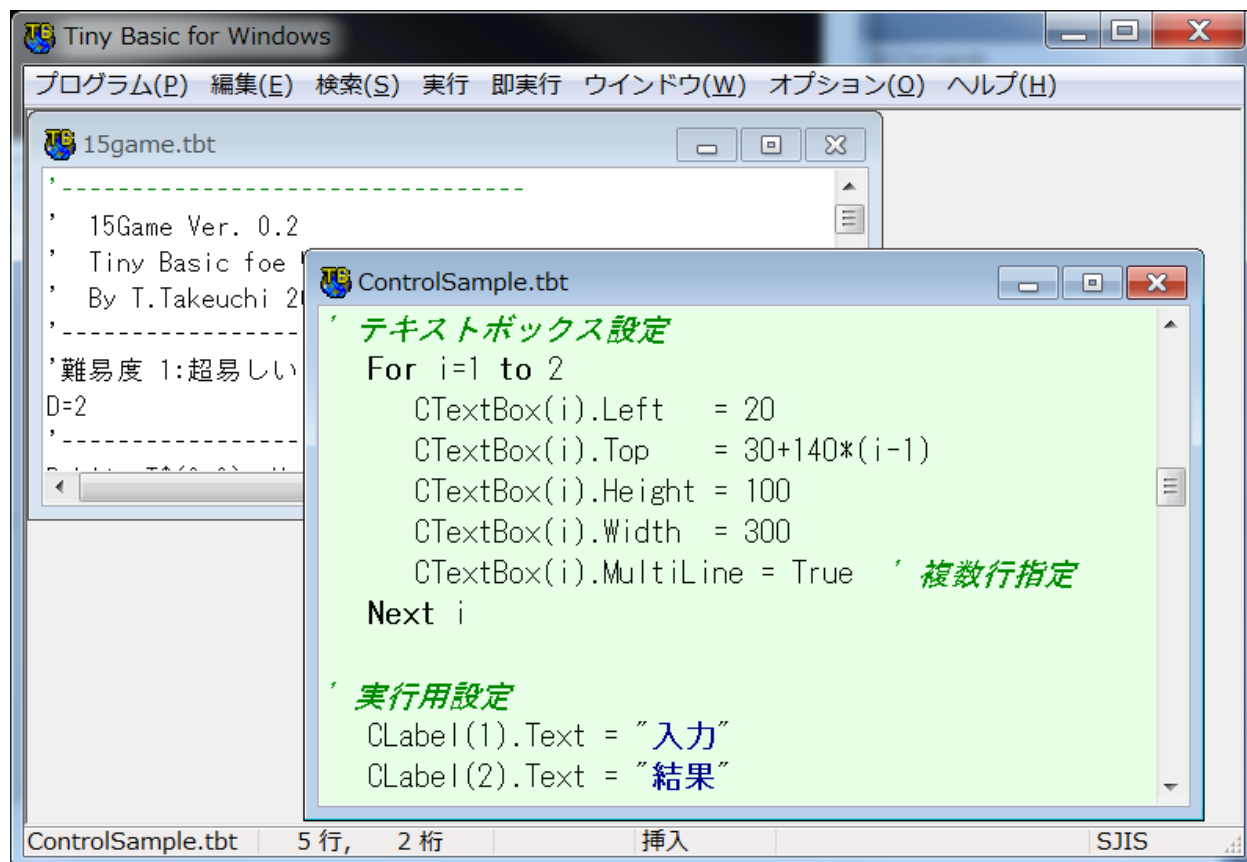


中央にあるのが編集画面、左上が実行画面です。編集画面は Tiny Basic の主画面で、この画面は、プログラムの作成、編集、保存など行います。

実行画面は計算結果の表示など、実行結果を文字で表示する画面です。またダイレクトモードも実行画面で実行します。グラフ画面は、必要に応じて表示されますが、線や点などを使ってグラフィックを表示するために使います。コントロール画面も必要に応じて表示されますが、ボタンや、ラベル、入力ボックスなどを使って処理を行います。

3. 2 プログラムの作成

次は編集画面で2つのファイルを編集しているところです。

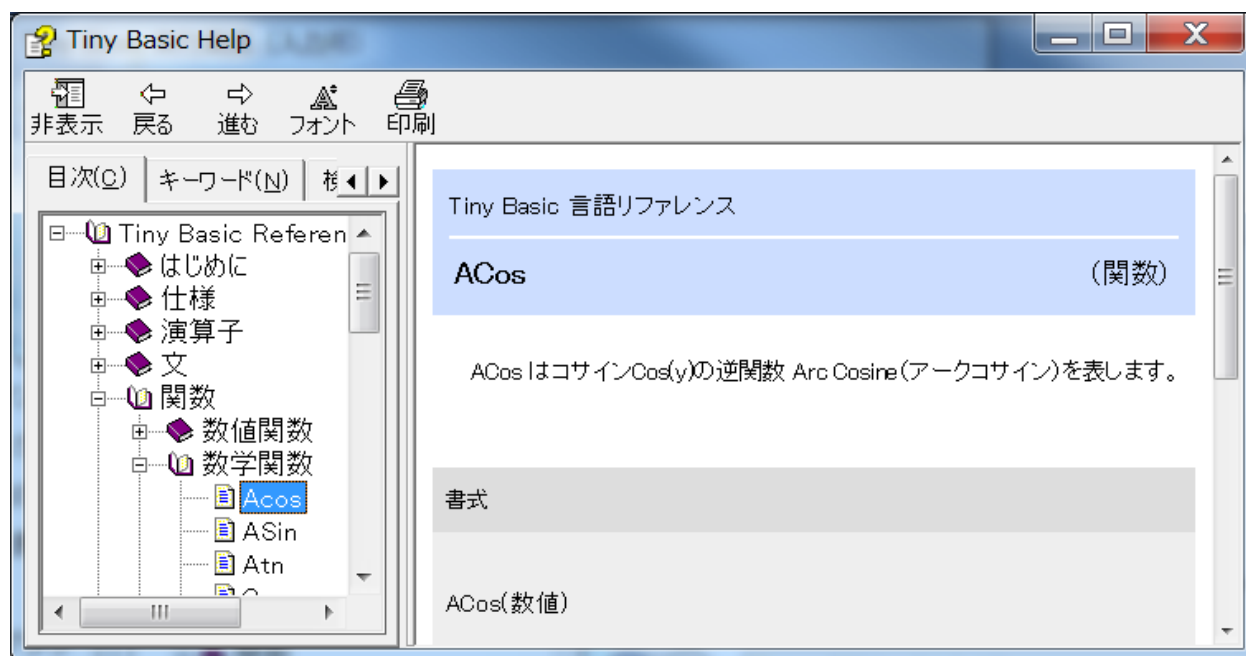


編集画面では内蔵のマルチウィンドウのエディターを使ってプログラムの作成を行います。基本的には Windows での標準的な操作に従っています。ですから、メモ帳等の経験があれば直ぐに使えます。

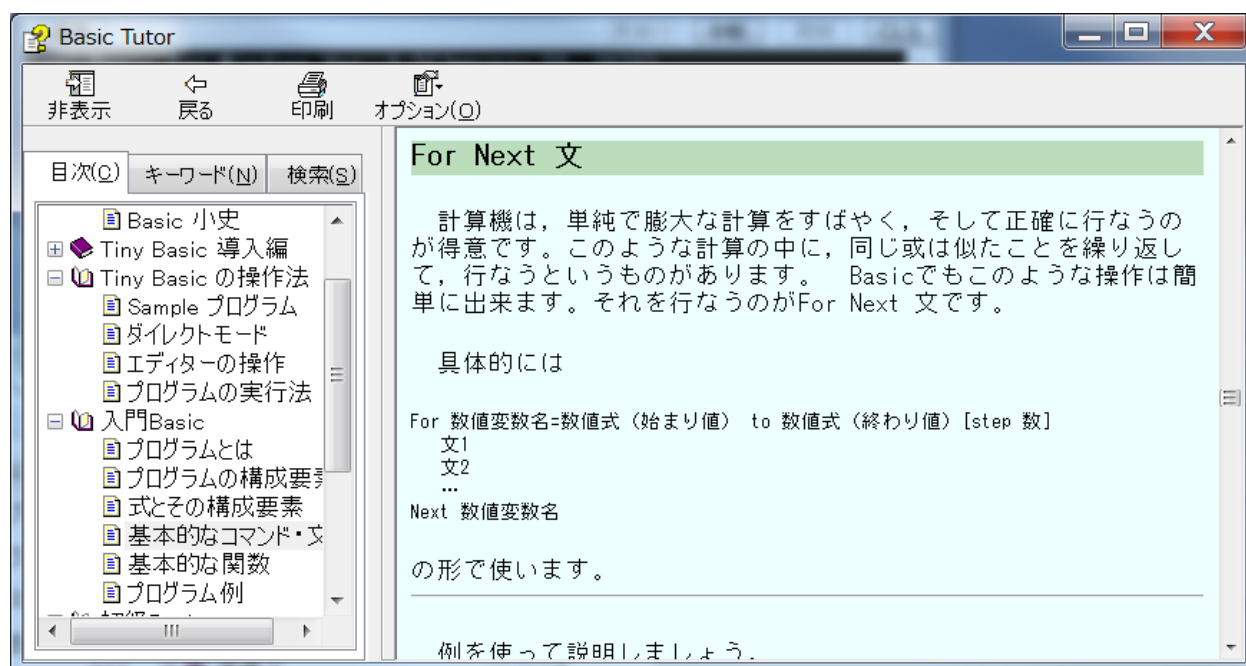
標準の機能のほか、プログラム入力の補助として、キーワードやコメント文などの色指定などの機能もあります。また、編集画面のフォントや種々の色のも好みにより色々な設定も可能です。編集集中のファイルのうち左上のものは、普通の表示ですが、右のものは、エディターの背景を変更したもので、更にプログラムの書式整形オプション使ったものです。

3. 3 ヘルプと独習

ヘルプボタンを押すことで、TBasic ヘルプを起動できます。ヘルプを利用することで Tiny Basic についての多くの情報が得られます。下はヘルプを起動した画面です。



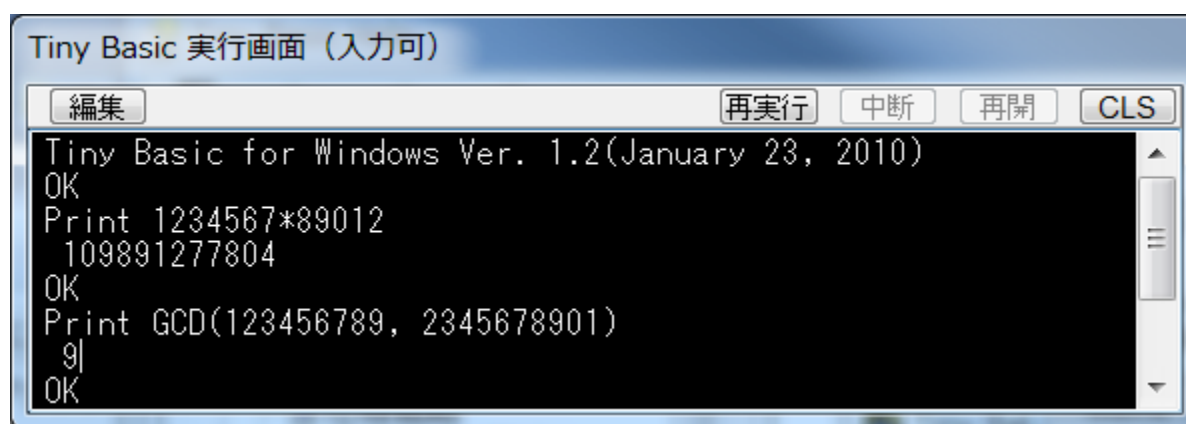
また Basic 入門用文書を利用して、Basic の独習もできます。下は Basic 入門を起動した画面です。



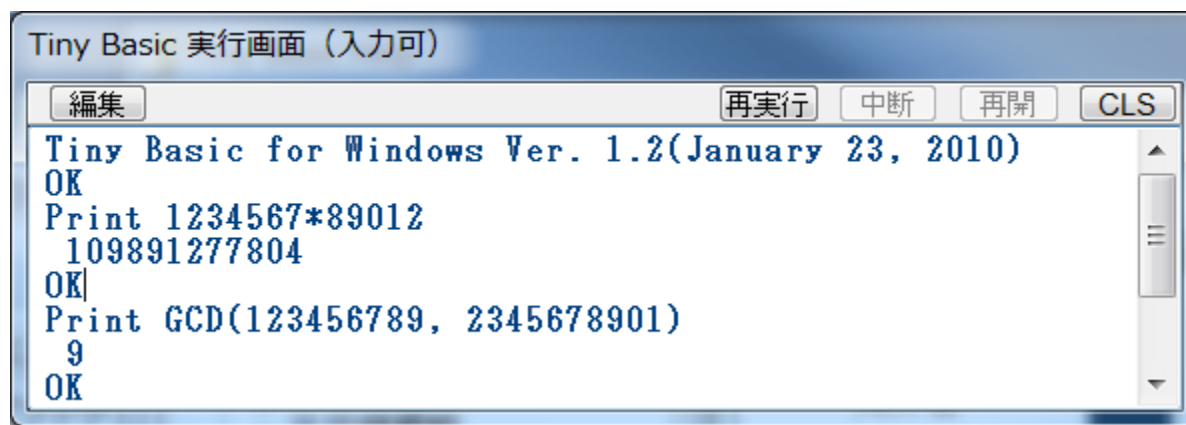
3. 4 実行画面

作成したプログラムは、実行ボタンを押すことで実行されます。テキストの実行結果は実行画面に表示されます。実行画面には一度に最大 1000000 行の結果を表示することができます。表示画面の結果をマウスで編集画面等にコピーすることも出来ます。実行画面の大きさと位置は、実行後変更することも、プログラムの中から指定することも出来ます。

また、実行画面はダイレクトモード機能を持っていて、キー入力することで、コマンドを直接実行することも出来ます。このダイレクトモード機能は Dos 窓風のヒストリ機能も持っていて、簡単な計算は電卓風にプログラムを書かないで実行することも出来ます。次の画面はダイレクトモードで実行している様子です。

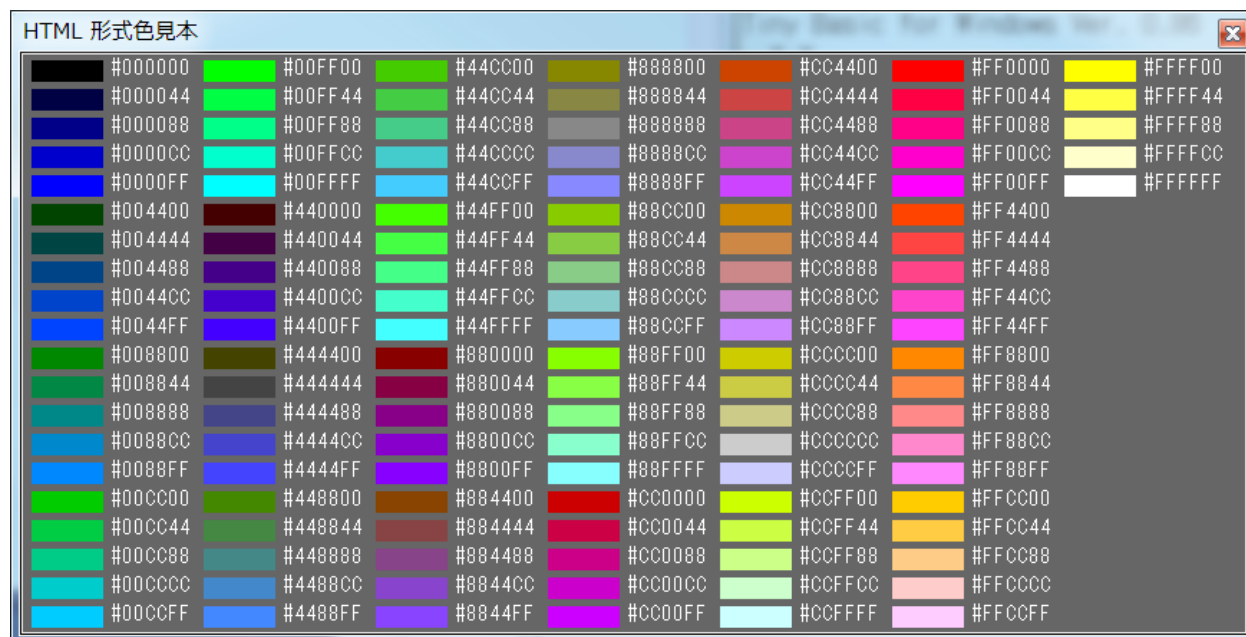


下の画面は同じものですが、実行画面の色やフォントを設定して使っている様子です。このように好みに合わせてカスタマイズすることも出来ます。

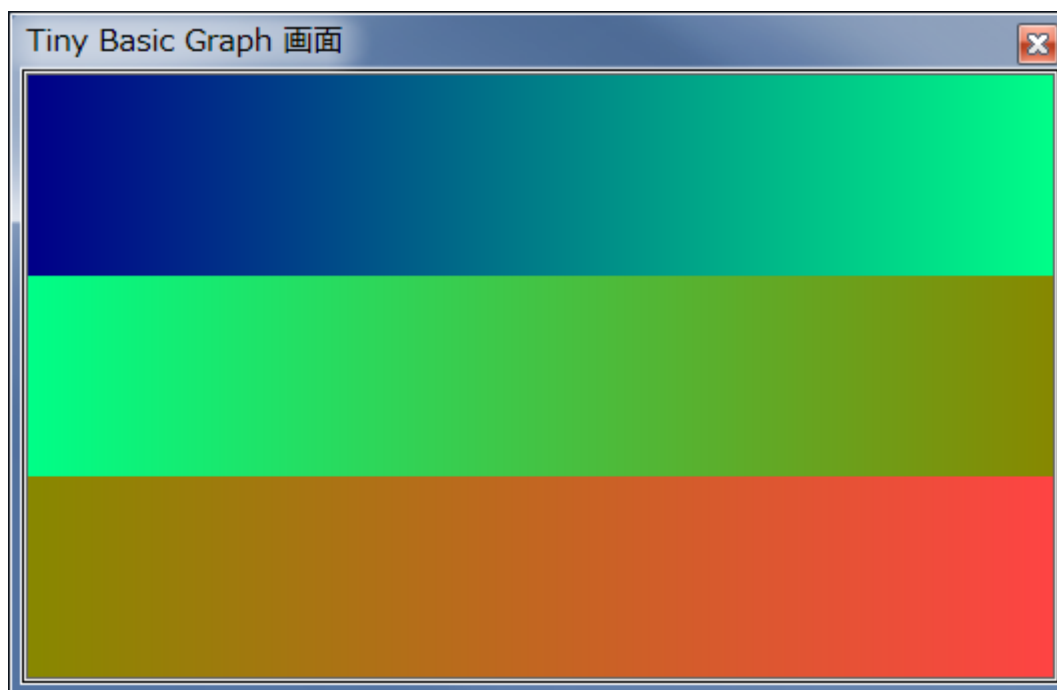


3. 5 グラフ画面

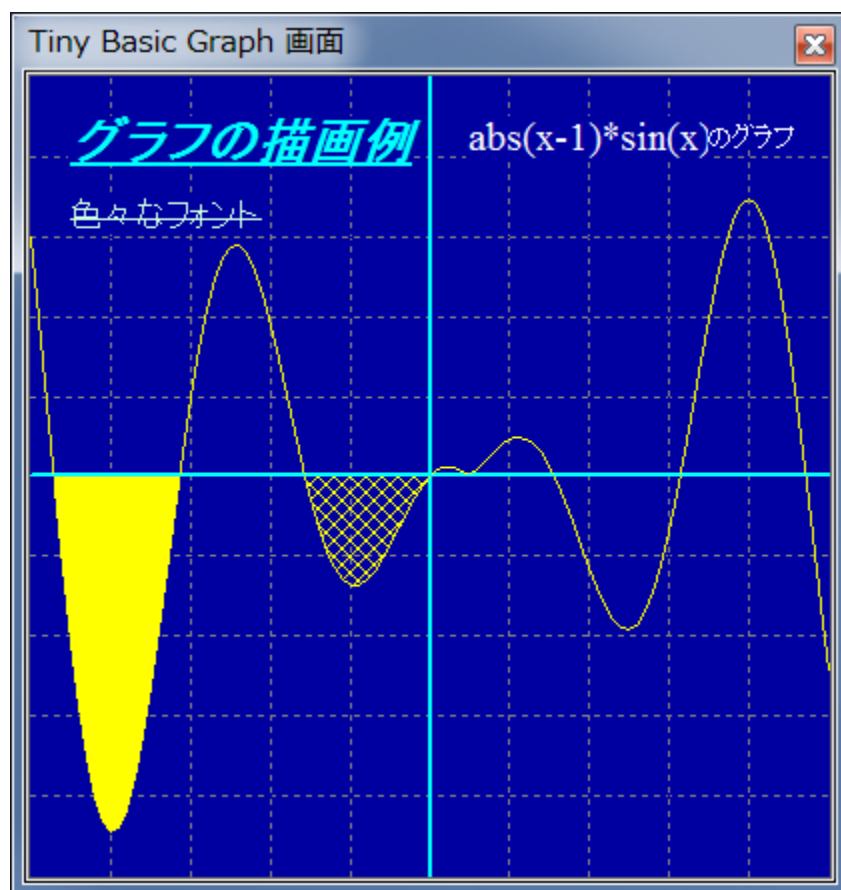
グラフ画面は起動時には表示されず、オープンすることで表示されます。この画面には種々のグラフを表示することが出来ます。大きさはディスプレイの解像度に従って設定することが出来ます。下の図は、TBasic で使える色の見本を表示するプログラムを実行したときの、グラフ画面です。



また下の図は TBasic でのグラデーションの例プログラムを実行したときの、グラフ画面です。



また下の図はグラフの例プログラムを実行したときの、グラフ画面です。

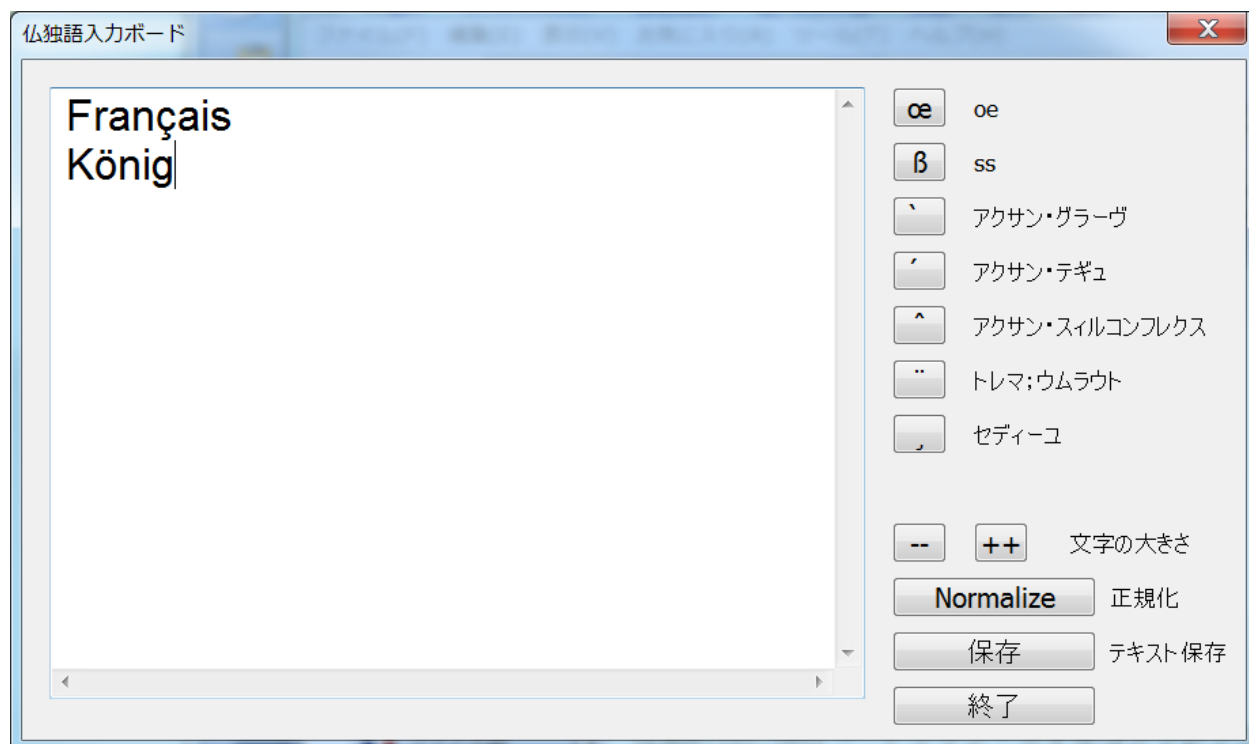


グラフ画面では、色々なフォントや、線の種類を使うことができます。グラフ画面の例は以下にもあります。

3. 6 コントロール画面

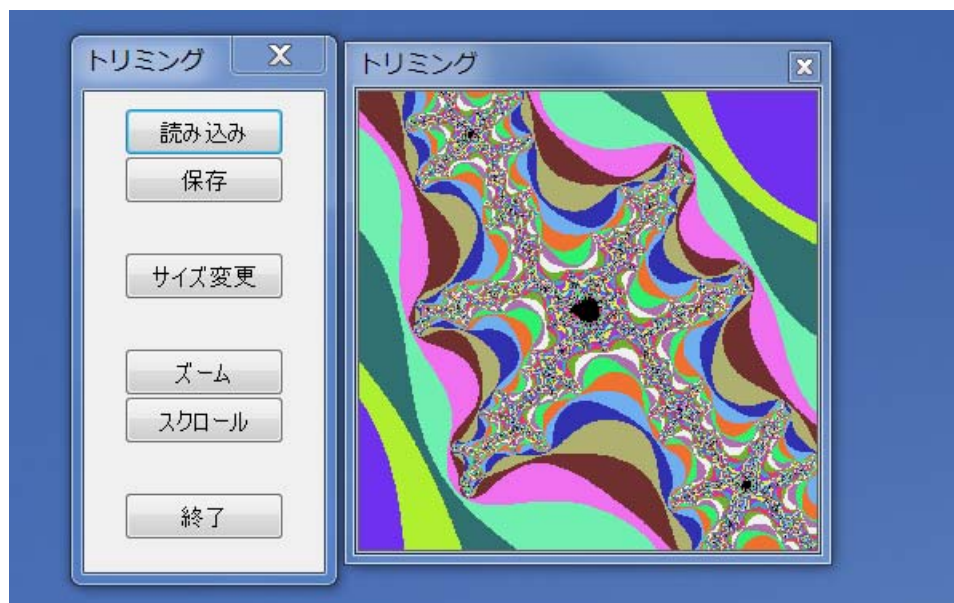
コントロール画面も起動時には表示されません。コントロール画面をオープンすることで表示されます。この画面ではボタン、ラベル、テキストボックスを使って Visual Basic 風のプログラムを書くことが出来ます。下は、Sample プログラムにある仏独語の入力プログラムを実行している画面です。

TBasic は Ver.1.2 よりユニコード対応になり、UTF-16 等を扱えるようになりました。このプログラムは、その機能を使った例で、アクサンや œ などの文字を簡単に入力できます。



またグラフ画面と、コントロール画面を組み合わせることで、更に色々なプログラムを書くことが出来ます。

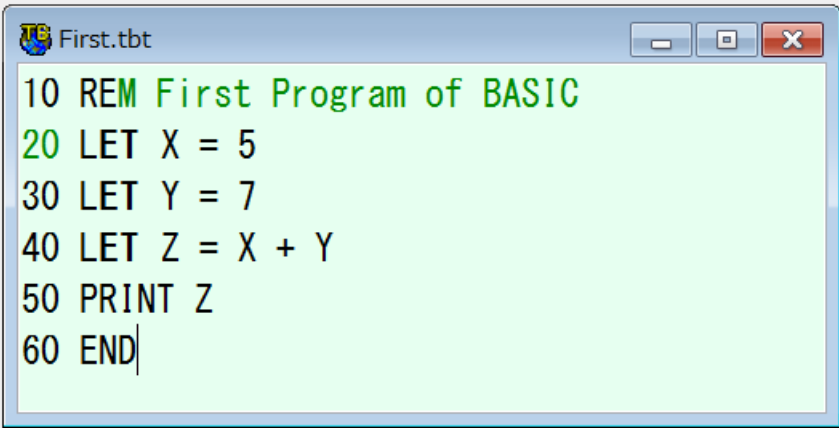
下図はコントロール画面とグラフ画面を組み合わせ使って、画像のトリミングをするプログラムを実行しているところです。右の画面は、マンデルブローットの例プログラムで作成した図です。



3.7 プログラム機能

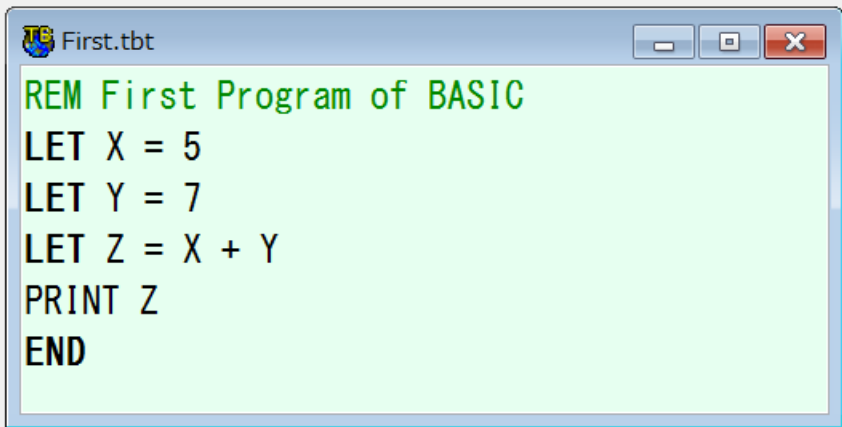
プログラムは基本的に行番号無しの普通のテキストファイルです。高校の教科書などでよく使われる N88-Basic 風 BASIC は行番号付ですが、行番号は古いラインエディタの名残で、現在積極的に使う意味はありません。現在 Windows 上での Basic や C など標準的プログラミングでは行番号はありませんので、TBasic はそれに従いました。しかし、行番号もラベルとして使うことができますので、N88-Basic 風のプログラムを使うことも出来ます。

次のプログラムはメニューとカーツが最初に BASIC の教育に使ったプログラムの例です。



```
10 REM First Program of BASIC
20 LET X = 5
30 LET Y = 7
40 LET Z = X + Y
50 PRINT Z
60 END
```

このようなプログラムも TBasic で動作させることもできます。もちろん現在の TBasic の標準的書き方では、行番号は使いませんから、上のプログラムは、次のようになります。

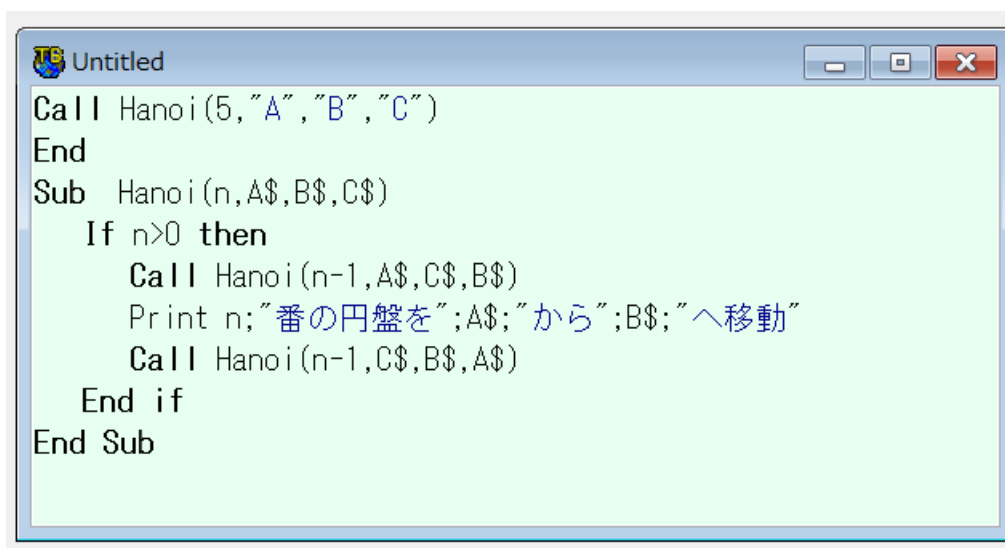


```
REM First Program of BASIC
LET X = 5
LET Y = 7
LET Z = X + Y
PRINT Z
END
```

標準の拡張子は tbt (Tiny Basic Text)です。しかしこれは関連付け等の便宜のために、実際の動作には関係しません。ですから、テキストファイルであれば、どのような拡張子のものでも使えます。

プログラムで利用できる演算子や制御コマンド、関数等は標準的なものは使えます。行番号は普通使いませんが、行ラベルとして、goto 行番号を使うことはできます。勿論この使い方は非推奨です。

TBasic は基本的に構造化言語ですので、種々の構造化構文を使ってプログラムを書きます。Sub 文や、Function 文も使えます。また、as Real や as String, as Boolean といった型宣言も使えます。そして Function や Sub での再帰的使用も可能です。Quick Basic の縮小 Windows 上版といった位置付けです。次のプログラムはハノイの塔のプログラムで Sub での再帰を使っています。



```

Call Hanoi(5,"A","B","C")
End
Sub Hanoi(n,A$,B$,C$)
  If n>0 then
    Call Hanoi(n-1,A$,C$,B$)
    Print n;"番の円盤を";A$;"から";B$;"へ移動"
    Call Hanoi(n-1,C$,B$,A$)
  End if
End Sub

```

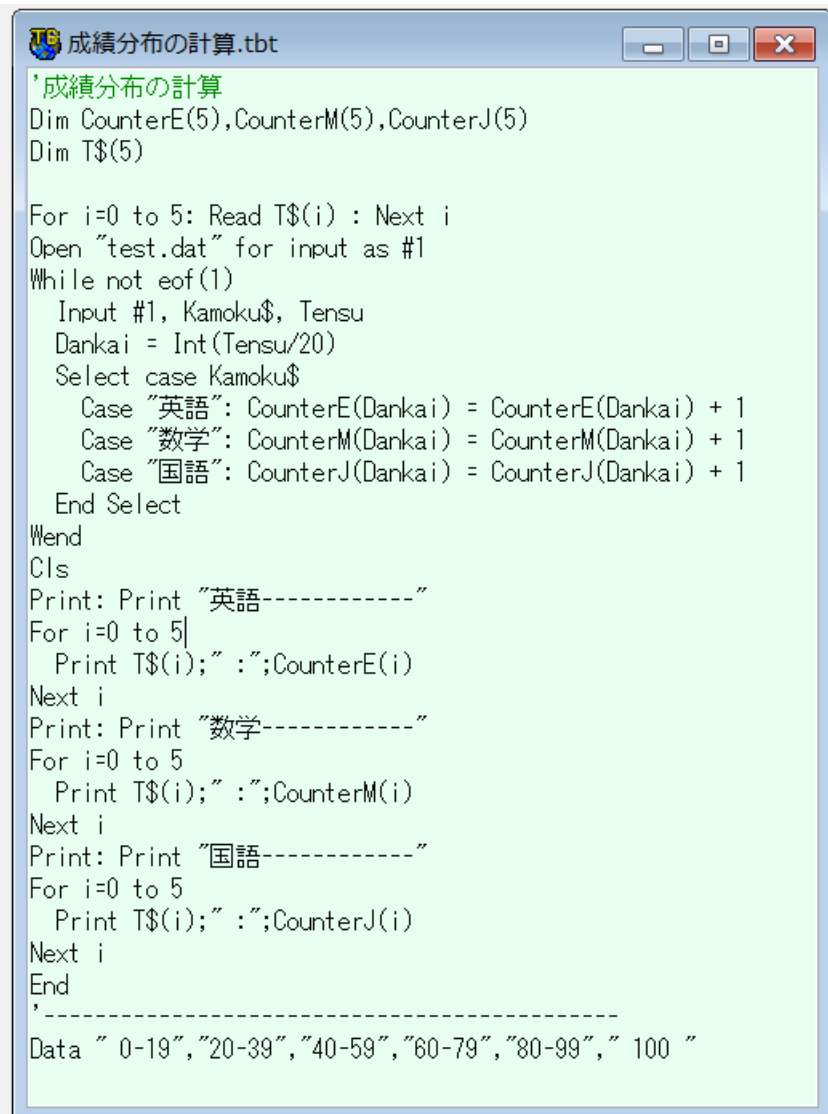
独自のコマンドとしては、文字列を数式に変換する StrToForm, その式の値を解釈する EvalForm もあります。独自の関数はかなりあります。整数論からのものでは最小素因数を返す PrimeFactor, 2項係数を返す Binomial, 最大公約数を返す GCD, 階乗を返す Factorial 等です。ファイル処理系の関数はファイル処理を関数で行うもので、成功失敗を返す関数です。失敗した場合エラーを返しますが、プログラムは停止しません。日付時刻処理関数では 閏年を判定する IsLeapYear や曜日を与える DayOfWeek があります。これらの独自コマンドや関数はあくまでも追加的なもので、教育用ではありません。ある程度慣れた人がツールとしてのプログラムを書くと

きのものです。あると便利と思われるものを適宜付け加えていますが、コマンドや関数を独自に追加する場合、Windows のプログラミングとして標準的なものの範囲を逸脱しないように心がけています。実際、これらの拡張は、現在使われている多くの言語で同様な機能や関数が提供されています。

4. Tiny Basic for Windows のプログラム例

以下、実際のいくつかのプログラム例を挙げて、使えるプログラムの雰囲気の説明します。

4. 1 成績分布の計算

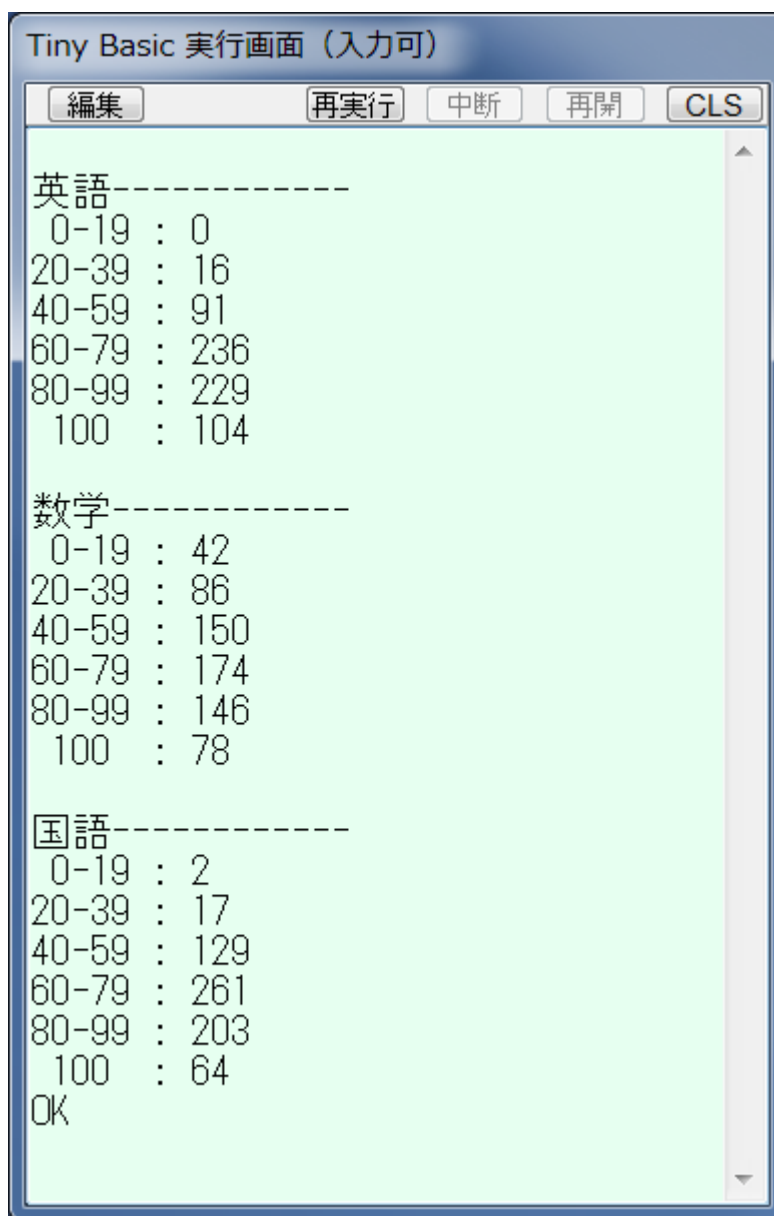


```

'成績分布の計算
Dim CounterE(5),CounterM(5),CounterJ(5)
Dim T$(5)

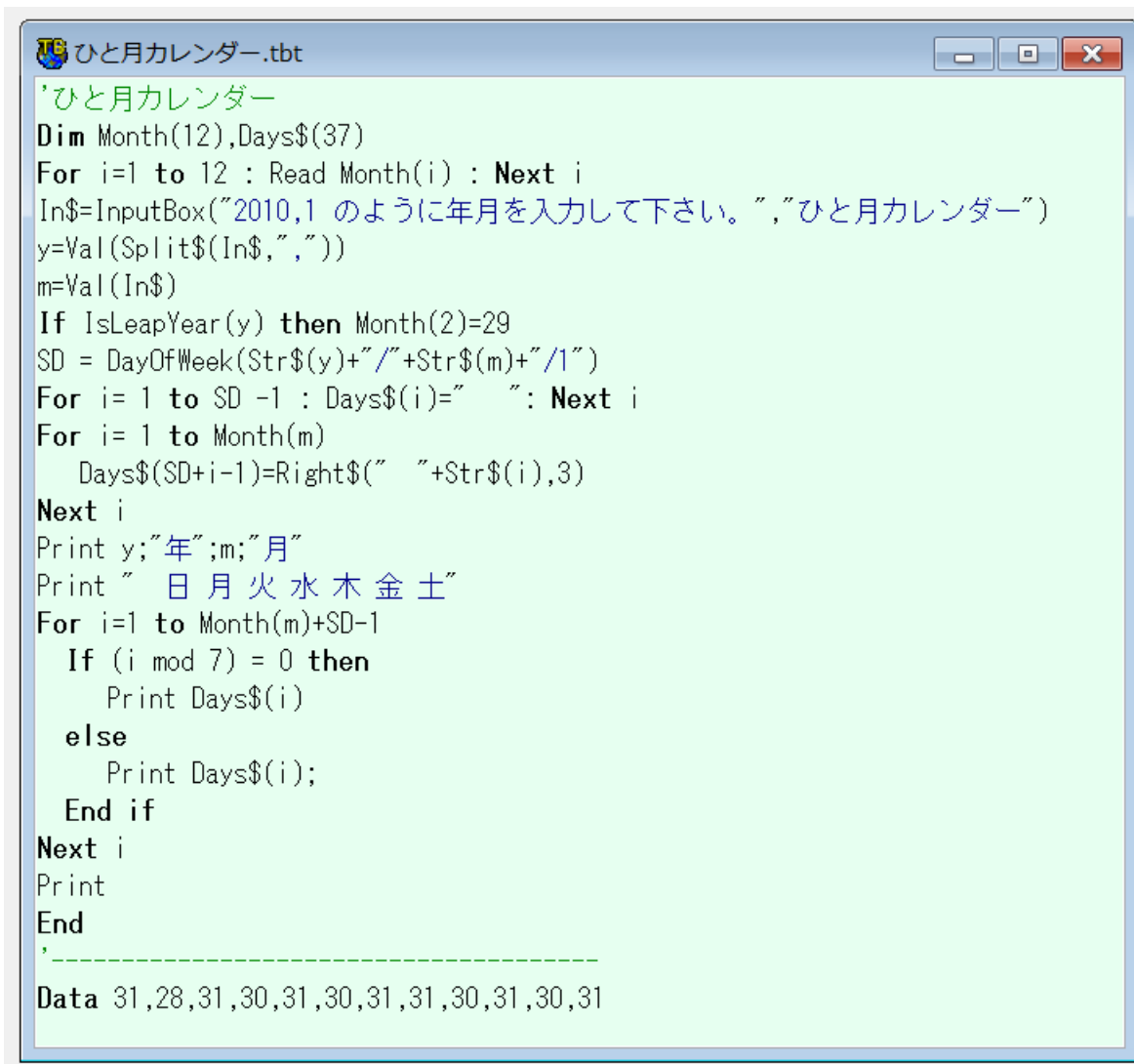
For i=0 to 5: Read T$(i) : Next i
Open "test.dat" for input as #1
While not eof(1)
  Input #1, Kamoku$, Tensu
  Dankai = Int(Tensu/20)
  Select case Kamoku$
    Case "英語": CounterE(Dankai) = CounterE(Dankai) + 1
    Case "数学": CounterM(Dankai) = CounterM(Dankai) + 1
    Case "国語": CounterJ(Dankai) = CounterJ(Dankai) + 1
  End Select
Wend
Cls
Print: Print "英語-----"
For i=0 to 5
  Print T$(i); " : "; CounterE(i)
Next i
Print: Print "数学-----"
For i=0 to 5
  Print T$(i); " : "; CounterM(i)
Next i
Print: Print "国語-----"
For i=0 to 5
  Print T$(i); " : "; CounterJ(i)
Next i
End
'-----
Data " 0-19", "20-39", "40-59", "60-79", "80-99", " 100 "
```

このプログラムは、英語、数学、国語の成績が書かれているファイル test.dat からデータを読み込んで点数分布を計算するものです。結果は実行画面に表示されます。このプログラムは標準的なコマンド使ったもので、Select Case 文を使っています。適当な tst.dat に対して計算させた結果は例えば次のようになります。



4. 2 ひと月カレンダー

次の例は、年、月を入力して、その月のカレンダーを出力するものです。Tiny Basic 特有の関数 Split\$, IsLeapYear, DayOfWeek を使っています。

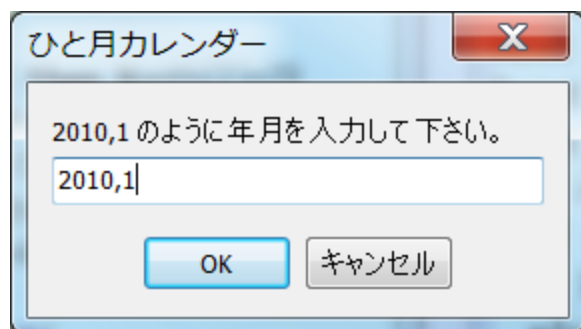


```

ひと月カレンダー
Dim Month(12),Days$(37)
For i=1 to 12 : Read Month(i) : Next i
In$=InputBox("2010,1 のように年月を入力して下さい。","ひと月カレンダー")
y=Val(Split$(In$,""))
m=Val(In$)
If IsLeapYear(y) then Month(2)=29
SD = DayOfWeek(Str$(y)+"/"+Str$(m)+"/1")
For i= 1 to SD -1 : Days$(i)="  ": Next i
For i= 1 to Month(m)
    Days$(SD+i-1)=Right$(" "+Str$(i),3)
Next i
Print y;"年";m;"月"
Print "  日 月 火 水 木 金 土"
For i=1 to Month(m)+SD-1
    If (i mod 7) = 0 then
        Print Days$(i)
    else
        Print Days$(i);
    End if
Next i
Print
End
'-----
Data 31,28,31,30,31,30,31,31,30,31,30,31
  
```

TBasic では Input 文で入力もできますが、ここでは InputBox を使っています。このように Visual Basic 風の入力も可能です。IsLeapYear は閏年かどうか判定する関数、DayOfWeek は曜日を返す関数です。これらを使うとカレンダーが簡単に作成できます。

入力画面と実行結果は次の通りです。



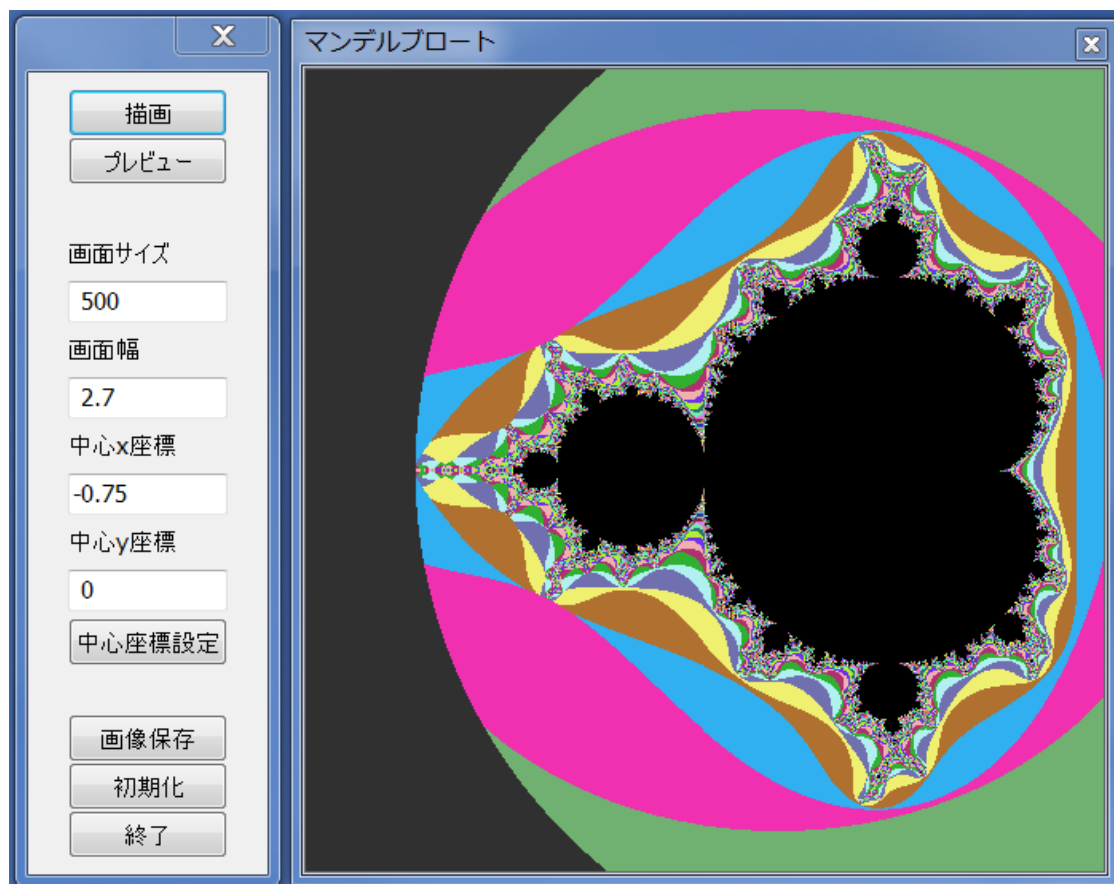
入力画面



実行結果

4.3 マンデルブロー集合 (グラフィックの例)

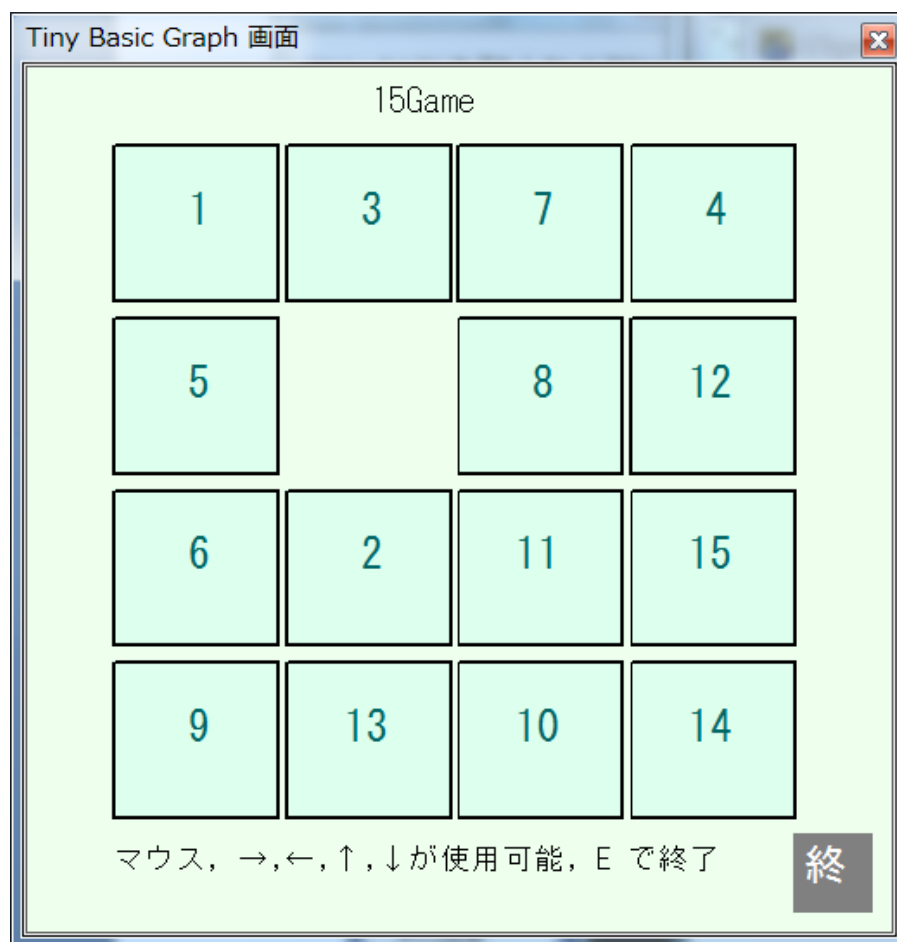
グラフィックの描画方法はグラフ画面の設定以外はマイクロソフト系の Basic と同様です。画面を開き、Window コマンドを使って、座標系を指定し、PSet や Line 等でグラフを描きます。



上の図はグラフィックを使った応用プログラムのマンデルブロー集合を描くプログラムを実行しているところです。このプログラムでは、中心座標の設定がその点でマウスをダブルクリックすることで可能です。このようにグラフィックを描くだけでなく、種々の操作が可能なプログラムを作成することができます。このプログラムは Sample の中にありますが、マンデルブロー集合の色々な部分を色々な大きさで書くことを目的としたものです。トリミングのところにあった図はこのプログラムを使って、上の図の一部を拡大したものを画像保存したものです。

4. 4 ゲームの例

元々TBasic はグラフィックを使ったゲームのようなものの作成は得意ではありません。しかし、動きの少ないものであれば、ゲームも作ることは可能です。以下は配布ファイルに同梱されている 15Game の起動画面です。このようにマウスを使った処理も可能です。



5. 最後に

昔、「コンピューターの勉強イコール BASIC の勉強」という時期がありました。しかし、今の情報リテラシー教育では「BASIC の勉強」は無いのが普通です。これはコンピューターの使い方の変化によるもので、自然なことですが、しかし他方プログラミング教育がコンピューターで必要なくなった訳ではありません。依然として中級以上のコンピューター利用ではプログラミング能力が必要です。これら必要とされるプログラミングは BASIC というわけではありませんが、BASIC の属する手続き型の言語が主なものです。ですから、コンパクトで簡単に使えるインタプリタ型の BASIC はその入門としては優れている言語です。

一方でコンピューターを取り巻く環境の変化はかなり早く、プログラミング手法もそれに対応して変化しています。ですから入門用の言語といえども、それに対応して行くべきでしょう。大きなプログラムを書きその資産を継承していくべき言語は仕様の変更には慎重であるべきですが、そうでないものでは、ある程度時代の趨勢を見て独自の拡張をしても良いと思います。

TBasic の仕様は元々確定したものに従っていませんので、かなり柔軟です。あると便利で、一般的に良く使われる機能は積極的に取り入れていきたいと思います。しかし、その際、Visual Basic との整合性が損なわれないように心がけたいと思います。

また、ヘルプの充実と、エラーメッセージの適切化は終わらない課題です。この他まだまだ不満なところはありますが、徐々に直していきたいと思います。

使用しての意見・要望がありましたら、

ホームページの掲示板、あるいは電子メール(takeuchi@math.sc.niigata-u.ac.jp)でお寄せください。