

# 超小型記号言語 取扱説明書(第11版)

## ■はじめに

「超小型記号言語」は数値計算とパズル的なプログラミングを行うために作った独自仕様のプログラミング言語です。  
言語処理系は非常にコンパクトで、数値は32ビット符号付整数のみ、エラーチェックは皆無に近く、文法的に誤っていてもインタプリタは実行を継続します。  
文法はFORTHと類似していて、逆ポーランド記法、ディクショナリの考えを取り入れています。  
記号言語というその名の通り、1文字の記号が命令として動作します。

例えば、記号(!)はスタックから2つのデータをPULLして、これを「書込データ」と「書込アドレス」と見なしてメモリにデータを書き込む命令です。次のプログラム  
#123 #2 !  
は、123と2をスタックにPUSHした後、スタックから2つのデータをPULLして、123を2番地のメモリに書き込みます。

なお、超小型記号言語では文字A～Zを数値0～25と同等に扱うため、Cは2を意味し  
#123 C !  
と書いても同じ動作を行います。あたかも変数Cに123を代入しているように見えます。これが超小型記号言語における変数への代入の仕組みです。  
同様に、記号(@)を使うと  
C @  
で変数C(実際は2番地)を読み出すことができます。

## ■動作環境、インストール、起動、削除

動作環境	日本語版Windows11で動作します。Windows11以外でも動作するはずですが、確認はしていません。
インストール	配布ファイルを適当なZIP解凍ツールで解凍し、解凍したファイル全体を1つのフォルダに入れます。
起動	フォルダの中のvts.exeを実行することで「超小型記号言語」が起動します。(*1) コマンドプロンプト画面に「?」を表示したら、超小型記号言語で記述されたプログラム(*2)のファイル名(hello.vtsなど)を入力すると実行します。これをプログラムモードと呼びます。 ファイル名の代わりに先頭に記号>を付けて1行プログラムを入力すると、その1行のプログラムを実行します。これをダイレクトモードと呼びます。
削除	削除する場合は、フォルダごと削除してください。

(\*1) 1回目の起動で「PC が危険にさらされる可能性があります」と警告が出たら「詳細情報」から「実行」をクリックしてください。不安な方は、ダウンロードしたファイルをWindows Defenderでスキャンして、ファイルの安全性をチェックしてください。  
(\*2) あらかじめメモ帳などのエディタでプログラムを作って、ファイル保存しておく必要があります。

## ■使用条件、保証範囲

本ソフト、ドキュメントは無保証です。複製、配布(ソースを改造しての配布を含む)は自由に行ってください。

## ■その他

- ・この取扱説明書(第11版)は、超小型記号言語Ver1.9に対応しています。
- ・本文書で使用する製品名は、各社、各組織の登録商標、または商標です。
- ・この取扱説明書では、システムディクショナリ(sys.dic)で定義されたワードの解説はしていません。必要に応じて、テキストエディタで開いて解読するか、同梱の別資料をご覧ください。

# 本編 文法解説

## ■数値、数式、数値出力

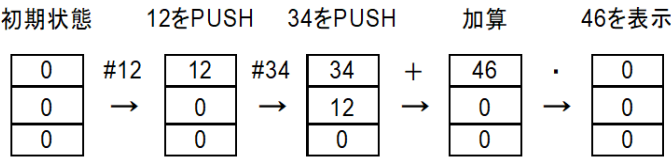
数値は32ビット符号付き整数(-2147483648～2147483647)のみで、数値の前には記号( # )を付けます。  
数式は逆ポーランド記法で、1文字以上のスペースで区切って

<数値1> <数値2> <演算子>

とします。たとえば

#12 #34 + .

は、12と34をスタックにPUSHした後、記号(+ )で12+34を計算し、記号(. )により計算結果46を画面に表示します。(\*1)



実行途中に数値をキーボードから入力する場合は、数値の代わりに記号(?)を使います。

?? + .

この場合、記号(#)は不要です。

負数は扱えるのですが、定数として記述できません。代わりに

#0 #1234 -

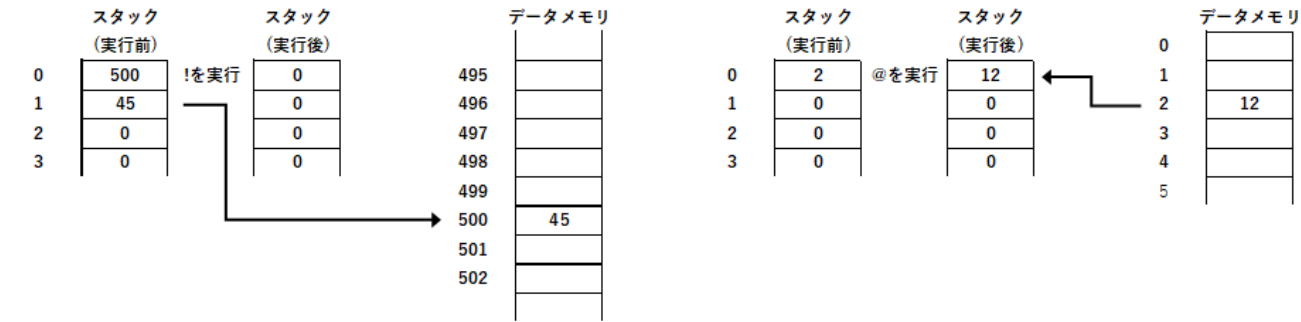
のように0から正数の引算を行ってください。これで-1234がスタックにPUSHされた状態になります。  
スタックは、0番地～65535番地(1番地あたり32ビット)の容量があります。

スタックを初期化(\*2)するには、ダイレクトモードの先頭でワードinitを実行します。  
?>:init  
このワードinitはダイレクトモードの先頭以外では動作しません。

(\*1)ここでは、プログラムモードの場合で説明しています。ダイレクトモードの場合は、先頭に>を付けてキー入力します。  
(\*2)ワードinitは、本図で説明しているスタックと、言語処理系が内部で使用するシステムスタックの両方を初期化します。なお、ここでいう「初期化」とはスタックポインタの初期化のことです。

■データメモリ  
スタックとは別に、0番地～65535番地(1番地あたり32ビット)のデータメモリがあります。  
<数値1> <数値2> !  
で、<数値1>を<数値2>が示す番地のデータメモリに書込みます。たとえば  
#45 #500 !  
は、数値45が500番地のデータメモリに書き込まれます。(図左参照)

逆に、データメモリから数値を読む場合は  
<数値1> @  
とします。これで<数値1>が示す番地のデータメモリを読み出し、その値をスタックにPUSHします。たとえば  
#2 @  
は、2番地のデータメモリの値がスタックにPUSHされます。(図右参照)



■変数、配列  
アルファベット大文字(A～Z)を数値(0～25)と等価に扱います。この仕組みを利用することで  
<数値> <変数名> !  
という書式で、変数に数値を代入することができます。たとえば、変数Cに数値1000を代入するには  
#1000 C !  
としますが、実際にはデータメモリの2番地に数値1000を書き込んでいて、次の式と等価です。  
#1000 #2 !

変数の値を読む場合は  
<変数名> @  
とします。次のプログラムは、変数Aと変数Bの乗算結果を画面に表示します。  
A@ B@ \* .

変数Z(25番地)以後のデータメモリ(26～60999番地)は配列として使用可能です。たとえば、配列Z(10)に5を代入するには  
#5 Z #10 + !  
とします。  
一見複雑に見えますが、変数Zの番地番号(数値25)に数値10を加算した数値(35)が示す35番地をアクセスしているのがわかります。  
なお、データメモリ61000～65535番地はシステムディクショナリで使用するので、通常のプログラミングでは使用しないでください。

■演算子  
次の演算子が使用できます。

演算の種類	超小型記号言語での記号
算術演算子の加減乗除と剰余	+ - * / %
比較演算子の<、>、=、≠	< > = ≠
論理演算子のAND、OR、NOT	&   ^

NOT以外は2項演算子で、スタックからPULLした2個のデータを演算し、結果をスタックに1個PUSHします。  
比較演算は、比較の結果が真なら全ビット1(符号付10進数で-1)、偽なら全ビット0(符号付10進数で 0)の値になります。

NOT(^)のみ単項演算子で、スタックの増減はありません。

■ワード(サブルーチン)定義

超小型記号言語ではサブルーチンを「ワード」と呼びます。

プログラム中で記号(：)に続いて1～16文字のワード名を記述すると、そのワードが実行されます。

:<ワード名>

ワードの定義は、メインプログラムの後に中括弧を使って

{<ワード名> <そのワードの処理内容を表すプログラム> }

と記述します。(\*1)このとき、記号( { )の直後に空白を入れず、逆に記号( } )の前に空白または改行が必要です。たとえば、2個の数値の平均を計算するワードAVGを定義するには

{AVG + #2 / }

とします。(\*2)

ワード呼び出し前後で引数や戻り値が必要な場合、スタックまたはデータメモリを使用します。

(\*1)あらかじめシステムディクショナリやユーザディクショナリで定義してあるワードを使用する場合は不要です。

(\*2)試しに画面で?表示の直後に、ダイレクトモードで

>#100 #200 :AVG . {AVG + #2 / }

と1行入力すると、150が表示されます。

■プログラム

超小型記号言語のプログラムはメモ帳などの汎用エディタで作成する必要があります。

プログラムは、先頭で実行させたい処理 (メインプログラム)を超小型記号言語の命令またはワードの組み合わせで記述し、その後に必要に応じてワード(サブルーチン)定義を記述します。

プログラムファイル 名の拡張子の推奨はvtsですが、他の拡張子でも特に問題ありません。

プログラムファイル(□.vts)

メインプログラム
ワード定義 ...

■システムディクショナリ

システムディクショナリ(sys.dic)は、実行ファイル(vts.exe)と一緒に提供されます。汎用的なワードは、あらかじめシステムディクショナリで定義してあります。システムディクショナリは、プログラム実行直前に読み込まれます。

システムデクショナリ(sys.dic)

ワード定義 ...
--------------

■ユーザディクショナリ

ユーザディクショナリ(usr.dic)も、実行ファイル(vts.exe)と一緒に提供されますが、空のファイルです。自由に書き換えてワードを定義することができます。ユーザディクショナリも、プログラム実行直前に読み込まれます。

ユーザデクショナリ(usr.dic)

ワード定義 ...
--------------

ユーザ独自のワードは、システムディクショナリではなく、ユーザディクショナリに登録することをお勧めします。

■条件分岐、ラベル

条件分岐命令は次の書式です。

<数値1> <数値2> <比較演算子> <分岐先ラベル名>]

ラベル名はアルファベット小文字(a～z)で、比較結果が真(0以外)なら同じラベル名のラベルへジャンプし、偽(0)ならジャンプしません。

たとえば、N>100のときラベルyへジャンプさせるには

N@ #100 > y]

とします。なお、分岐先のラベルは次の書式です。

[<ラベル名>

先程のプログラムの分岐先であれば

[y

と記述します。

ラベルのスコープはメインプログラム内、または各ワード定義内でのみ有効です。

言い換えると、メインプログラム内と各ワード定義プログラム内で、それぞれが最大26個のラベル名(a～z)を使えることになります。

無条件ジャンプさせるには、初めから比較結果を真(0以外)とし

#1 <分岐先ラベル名>]

のように記述します。

■文字出力、コメント

文字列出力は、文字列を記号(")で囲みます。改行する場合は後に記号(;)を付けます。

"HELLO WORLD";

文字コードによる1文字出力は、文字コードの後に記号(¥)を付けます。

特に文字コードが改頁(12)の場合、画面クリアしカーソルは左上に移動します。

コメント文字列は、文字列を小括弧で囲みます。

(KOREHA COMMENT DESU)

■組み込みワード

処理系(vts.exe)で最初からサポートしているワードを、組み込みワードと呼びます。

次の組み込みワードをサポートしています。

組み込みワード	機能
init	スタックとシステムスタックのスタックポインタを初期化します。
trace	トレース機能を有効(*1)にし、以降のプログラム実行では、実行したワードを括弧付きで表示(*2)します。
troff	トレース機能を無効にします。
sdump	現在のスタックの内容をすべて表示します。 左端の数値がスタックトップになります。

組み込みワードはダイレクトモードの先頭で単独で入力します。例えばtraceワードであれば次のように入力します。

?>:trace

OK (\*3)

(\*1)「trace」実行で有効となったトレース機能は「troff」実行まで継続します。

(\*2)999 ワードを超えると表示を打ち切りますが、プログラムの実行は継続しています。

なお、表示するのは「:」で定義したワードだけで、記号、数字などは表示しません。

(\*3)組み込みワードを実行したのに「OK」を表示しない場合は、実行されていません。先頭に「>:」が無い、小文字でない、などが考えられます。

■実行終了

メインプログラム末尾まで実行すると、実行終了します。

メインプログラムの途中で実行終了させる場合は、メインプログラムの末尾までジャンプさせてください。

超小型記号言語(vts.exe)を起動後の実行例を図に示します。(\*1)

最初にプログラムモードで円周率の計算プログラム(同じフォルダ内にあるpai.vts)を実行し、31416を出力しています。

次にダイレクトモードで>に続けて1行プログラムをキー入力して12+34を計算し、46を出力しています。

```
超小型記号言語(VTS) Version1.9 (System Dictionary Version1.5)
( プログラムファイル名を入力、または先頭に>を付けて1行プログラムを入力してください )

?pai.vts
PAI WO KEISAN CYU!
PAI= 31416
?>#12 #34 + .
46
?
```

この画面で、Contorol+Cを押下すると超小型記号言語を終了します。

(\*1)処理系、システムディクショナリによって、1行目のバージョン表示は異なります。

以上

来歴(超小型記号言語 取扱説明書)
2018/09/17 新規作成、第1版とする
2018/09/17 誤記訂正し、第2版とする
2018/09/23 Ver1.4に対応し、第3版とする
2019/01/02 Ver1.5に対応し、第4版とする
2019/01/12 誤記訂正し、第5版とする
2020/05/30 全面改訂、PDF版とし、第6版とする
2021/04/25 Ver1.6に対応し、第7版とする
2025/07/21 Ver1.7に対応し、第8版とする
2025/07/27 Ver1.8に対応し、第9版とする
2025/08/11 Ver1.9に対応し、第10版とする
2025/08/14 誤記訂正、説明を変更/追記し、第11版とする