

ブラウザ上で動作する DNCL 処理系「Tetra」の開発

大門 巧^{1,a)} 大西 建輔¹

概要：近年では教育現場における ICT の活用が推進されているという現状がある．一方で特定のソフトウェアの導入には困難を伴う場合がある．そのため，Web ブラウザ上で動作するプログラミングの実行環境が作成されている．我々は，DNCL の実行環境に着目した．DNCL の既存の実行環境には入力支援の機能が充実していないという問題がある．そこで筆者は充実した入力支援の仕組みを取り入れたブラウザ上で動作する DNCL の処理系「Tetra」を開発した．Tetra では DNCL の実行，停止，変数リストと実行ログによるプログラムの追跡ができる．

キーワード：プログラミング，DNCL，入力支援

Development of browser-based programming environment "Tetra"

TAKUMI DAIMON^{1,a)} KENSUKE ONISHI¹

Abstract: In recent years, the use of ICT in the education field has been promoted. On the other hand, it may be difficult to install specific software in schools. Therefore, programming environment on a Web browser has been developed. We focus on DNCL of National Center Test for University Admissions. Existing programming environment for DNCL does not provide sufficient input assistance. We developed the new programming environment for DNCL "Tetra": it is able to run and stop program, track program with variable lists and logs.

Keywords: Programming, DNCL, Input support

1. はじめに

学習指導要領の改訂に伴い，2022 年度より高等学校でのプログラミング教育が必修になる．そのため，大学入試に情報の科目が設定されることが予想され，高等学校におけるプログラミング教育を如何にして行うかという問題に注目が集まっている．従来の大学入試センター試験には，情報関係基礎という科目がある．この科目のプログラミングの問題で，手順やアルゴリズムを表現する場合には，DNCL[1] という日本語で記述する疑似コードが使用されている．DNCL の処理系には，PEN という環境 [2] がある．文献 [3]，[4] では，Web ブラウザ上での実行環境となっている．これらのブラウザ上で動作する既存の処理系には，

入力支援の機能が充実していないという問題がある．また，DNCL の場合は，日本語で記述をおこなうことから，変換ミスなども多発する．中等教育の現場で利用するには，スマートフォンやタブレットにおいても利用しやすい入力支援の仕組みが求められている．そこで筆者は新たな入力支援の仕組みを取り入れた DNCL の処理系である「Tetra」を開発した．本稿では Tetra の実装とその詳細について報告する．

2. 関連研究

2.1 中村らの研究

中村ら [2] は，Java 仮想マシン上で動作する「プログラミング入門用学習環境 PEN」を開発した．PEN は，DNCL に変数の型宣言やグラフィックス用の組み込み関数などの拡張を施した xDNCL の実行環境となっている．PEN はブ

¹ 東海大学

Tokai University, Kanagawa, 259-1292, Japan

^{a)} 6bss2115@mail.u-tokai.ac.jp

ログラムの入力を支援するための入力支援ボタンやステップ実行の機能も備えている。パーサはパーサジェネレータにより自動生成されたものを使用している。

2.2 中西らの研究

中西 [3] は、Web ブラウザ上で動作するプログラミング学習環境である「WaPEN」を開発した。WaPEN は、記述したソースコードをフローチャートに変換することが可能である。また、問題の採点機能やステップ実行の機能も備えている。パーサはパーサジェネレータにより自動生成されたものを使用している。

2.3 本多らの研究

本多ら [4] は、Web ブラウザ上で動作する DNCL のプログラミング学習環境である「どんぐり」を開発した。どんぐりのサイトには、大学入試センター試験の情報関係基礎の科目の解答例が備えられている。プログラムは正規表現により JavaScript にトランスパイルしてから実行される。

3. 拡張 DNCL

3.1 概要

センター試験用手順記述標準言語 (DNCL) は、情報関係基礎の出題において使用される疑似言語である。DNCL には、代入、演算 (算術、比較、論理)、関数、表示、制御 (条件分岐、条件繰返し、順次繰返し) などが定義されている。DNCL は日本語で記述する言語であり、その言語仕様は独立行政法人大学入試センターにより Web 上で公開されている [1]。

Tetra では、DNCL に関数定義や連想配列などの拡張要素を加えた「拡張 DNCL」でプログラムを記述する。拡張 DNCL の言語仕様は仕様書 [1] に準拠しているが、条件分岐文の記述方法が一部異なる。また、関数の定義や文字列操作などの拡張機能がある。Tetra における拡張 DNCL には、試験問題のプログラムを原文に近い形で記述できるという特徴がある。図 1 に拡張 DNCL のプログラム例を示す。

```
/*DP と再帰で実装したフィボナッチ数列の第 n 項を求めるプログラム*/
N ← 10
fib ← {1,1}
n を 3 から N まで 1 ずつ増やしなが、
    fib[n] ← fib[n-1]+fib[n-2]//配列は数値以外でもマッピング可能
    を繰り返す
"fib("+N+") = "+fib[N] を表示する

//フィボナッチ数列の第 n 項を返す関数
関数 数列 (n) は
    もし n<2 ならば、n を返す//公式では「もし n<2 ならば n を返す」
    数列 (n-1)+数列 (n-2) を返す
    を実行する
```

図 1 拡張 DNCL のプログラム例

3.2 条件分岐文

仕様書 [1] では、条件分岐文の内部でおこなう処理を 1 行で書ける場合は、全体を 1 行で書くことができると定義されている。1 行で書く条件分岐文は DNCL では図 2 の 1 行目のように、拡張 DNCL では、図 2 の 2 行目のように記述する。拡張 DNCL の場合は、「ならば」の後にカンマ「,」が必要である。

```
もし x<3 ならば x を表示する//DNCL の場合
もし x<3 ならば, x を表示する//拡張 DNCL の場合
```

図 2 条件分岐文

3.3 関数の定義

関数の定義方法を図 3、図 4 に示す。Tetra ではすべての関数が戻り値を持つ。戻り値は「<値>を返す」で定義する (図 3)。「<値>を返す」を明示的に記述しない場合 (図 4) は、非数値「NaN」が戻り値となる。関数の定義は<処理>に含まれるため、関数の中に関数を定義することもできる。関数の多重定義を許していないため、同一の名称の関数を複数定義した場合は、最後に定義した関数が有効になる。

```
関数 <関数名> (パラメータ 1, パラメータ 2,...) は
    <処理>
    <値>を返す
    を実行する
```

図 3 値を返す関数の定義方法

```
関数 <関数名> (パラメータ 1, パラメータ 2,...) は
    <処理>
    を実行する
```

図 4 値を返さない関数の定義方法

3.4 組み込み関数

Tetra では表 1 に示す組み込み関数を提供する。これらの組み込み関数に使用されている名前をユーザー定義の関数に使用することはできない。

表 1 組み込み関数

大きさ (a)	文字列 a の大きさや配列 a の要素数を返す
配列初期化 (params...)	指定された容量で初期化された配列を返す
平方根 (x)	x の平方根を返す
乱数 ()	0 以上 1 未満の実数値乱数を返す
sin(x)	sin x の値を返す (x はラジアン)
cos(x)	cos x の値を返す (x はラジアン)
tan(x)	tan x の値を返す (x はラジアン)
log10(x)	log ₁₀ x の値を返す
log(x)	log x の値を返す
数値に変換 (x)	x を数値に変換した値を返す
切り捨て (x)	数値 x の小数点以下を切り捨てた値を返す

3.5 連想配列

Tetra における配列は連想配列として実装されている。そのため、数値だけでなく、文字列や実数値によるマッピングも可能である (図 5)。配列において何もマッピングされていない番地を参照した場合は、非数値「NaN」が返される。

```
a[1] ← 100
a[1.5] ← 200
a["あいいうえお"] ← 300
a[1] を表示する//100 が表示される
a[1.5] を表示する//200 が表示される
a["あいいうえお"] を表示する//300 が表示される
```

図 5 連想配列の例

4. DNCL 処理系 Tetra

4.1 処理系の概要

Tetra はソースコードを処理する字句解析器、抽象構文木を構成するノード、構文解析をおこなうパーサから成る言語処理系と、図 6 に示すユーザーインターフェースで構成される。言語処理系は JavaScript、ユーザーインターフェースは HTML で記述されているため、ブラウザ上で動作する。詳細なエラーレポートを提供するために、パーサとエディタはすべて自作したものを使用している。現在は Google Chrome, Mozilla Firefox, Safari, Opera において正常に動作することを確認している。なお、Internet Explorer では動作せず、Microsoft Edge では一部の機能に制限がある。スマートフォンやタブレットには、Google Chrome や Safari があるため、Tetra を動作させることができる。



図 6 Tetra のユーザーインターフェース

4.2 プログラムの実行

Tetra では、ステップ実行と通常実行が可能である。画面上部のステップボタンが押されると 1 ステップだけプログラムが実行される。ステップ実行の実行速度は画面上部のスライダーにより変更可能である。多くの実行環境で

は、ステップ実行は 1 行単位でおこなわれることが多いが、Tetra ではステップ実行を 1 トークン単位での実行としている。これは、プログラミング初学者に対してプログラムの構造を理解させることを目的としているためである。ステップ実行を実現するために、Tetra では抽象構文木のノードの走査を忠実に再現している。

通常実行ではプログラムを停止させることなく実行する。プログラムは画面上部の実行ボタンを押すことで実行され、実行中断ボタンを押すことで実行を中断することができる。

プログラムの実行処理は図 7 のようにおこなわれる。ユーザーが作成したプログラムは、字句解析器によりトークン列に分解される。トークン列に基づき、再帰下降構文解析により、構文解析器が抽象構文木を構築し実行に至る。通常実行の場合は Worker という処理単位を用いてバックグラウンドで実行される。ステップ実行の場合は yield を用いてジェネレータ関数を逐次呼び出すことで実行される。

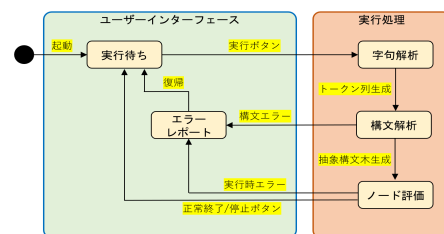


図 7 プログラムの実行処理

4.3 入力支援ボタン

DNCL は日本語で記述する言語であるため、変換ミスが多発する。入力支援ボタンを用いてこの問題を解決する。入力支援ボタンはエディタの左側に配置されており、折りたたみが可能である。また、拡張 DNCL のすべての構文が用意されている。入力支援ボタンを押すと図 9 に示すダイアログが表示される。ダイアログには図 10 のように、定義した変数や関数名の候補をリスト形式で表示する機能がある。ダイアログに必要な事項を入力し決定ボタンを押すと、図 8 のようにエディタのカーソルのある位置にコードが自動生成される。自動生成されたコードが入力され、コードも自動的にインデントが整えられる。

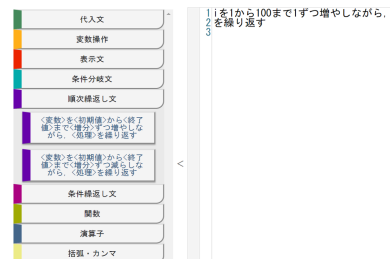


図 8 自動生成されたコード

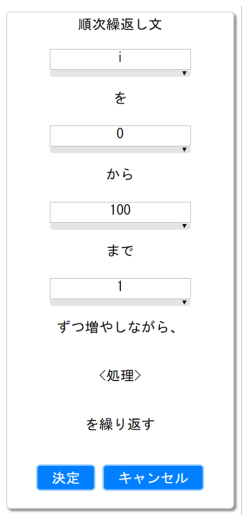


図 9 ダイアログ

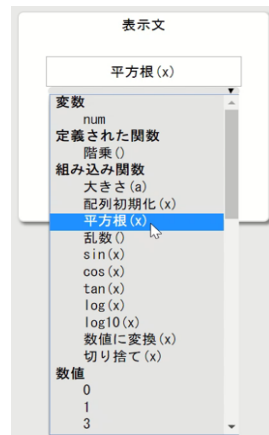


図 10 変数名や関数名のリスト

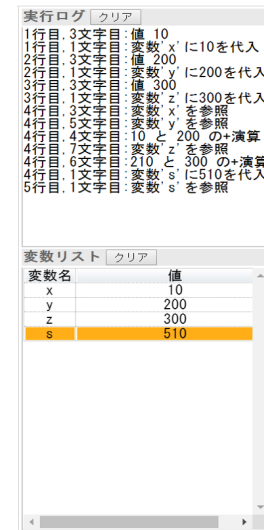


図 12 変数リストと実行ログ

4.4 エラーレポート

構文エラーや実行時エラーが起きた場合は、エラーが起きた行の行番号がハイライトされ、コンソールにエラーメッセージが表示される(図 11)。エラーメッセージにはエラーの詳細情報が記されている。図 11 に 0 での除算がおこなわれた場合のエラーレポートを示す。

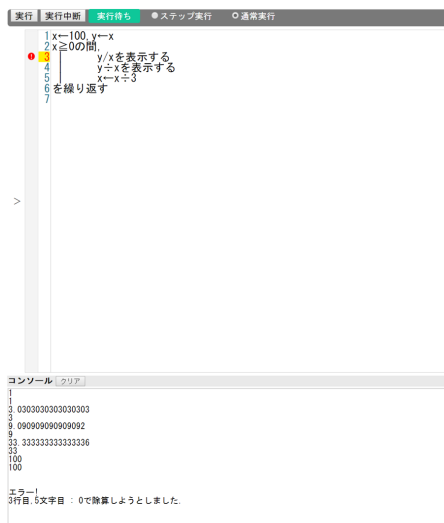


図 11 エラーレポートの様子

4.6 エディタ

Tetra では、入力支援ボタンを使用してプログラムを記述することを前提としているため、エディタにおいて改行した際や実行時に自動的にプログラム全体のインデントが整えられる。また、エディタには図 13, 図 14 のような行番号のハイライト機能があり、実行中の行やエラーの起きた行をユーザーに分かりやすく示すことができる。

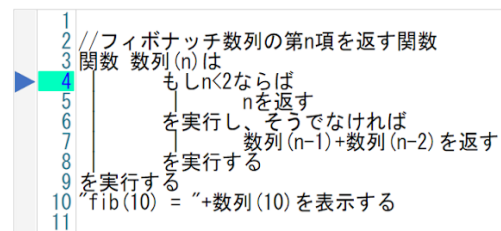


図 13 実行中の行のハイライト

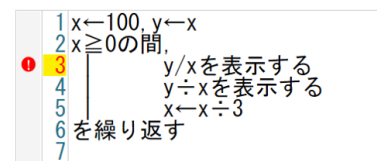


図 14 エラーの起きた行のハイライト

4.5 変数リスト・実行ログ

Tetra では、ステップ実行時に変数の値と実行ログを表示する機能を提供している。変数リストと実行ログはプログラムの 1 トークンが実行される度に更新される。変数リストには、プログラムの実行中に変数の参照がおこなわれると、参照された変数がハイライトされる機能がある。

5. おわりに

現在は処理系の機能の充実をおこなっている。Tetra には既存の DNCL の処理系には見られない新たな入力支援の仕組みが取り入れられている。筆者は、入力支援ボタンはプログラミング初学者に対してプログラム作成のヒントを与えることのできる有用なものであると考えている。また、Tetra はパーサジェネレータなどのツールを使用せずに書かれた言語処理系である。そのため、詳細なエラーレ

ポートを提供する機能を備えている。今後は評価実験を実施し、Tetra でプログラミングを学習した高校生が、プログラムの理解を深めることができるという利用価値を明らかにしていきたい。

参考文献

- [1] 独立行政法人大学入試センター:センター試験用手順記述標準言語 (DNCL) の説明, 入手先 (<https://www.dnc.ac.jp/albums/abm00004841.pdf>) (2019.09.06).
- [2] 中村亮太, 西田知博, 松浦敏雄. プログラミング入門用学習環境 PEN. 情報処理学会 コンピュータと教育研究会, CE(81), pp.65-71, 2005.
- [3] 中西渉. Web ブラウザ上のプログラミング学習環境 WaPEN の改良. 情報処理学会, 情報教育シンポジウム (2019), pp.130-135, 2019.
- [4] 本多佑希, 兼宗進. ブラウザ上で動作する DNCL 学習環境「どんぐり」の開発. 情報処理学会 コンピュータと教育研究会, CE(147), 2018.