

# プログラミング初学者の自己学習のための 変数を中心としたプログラム理解支援環境

下田 修平<sup>†</sup> 横濱 拓樹<sup>†</sup> 中小路 久美代<sup>†</sup> 山本 恭裕<sup>†</sup>

公立はこだて未来大学システム システム情報科学部 アーキテクチャ学科<sup>†</sup>

## 1. はじめに

プログラムの内容を理解する際には、変数の値に着目してソースコードを追うことが重要である。実行中の変数の動きを確認する方法の一つとして `print` 文で変数の値をコンソールに出力することが挙げられるが、実行時の変数の値の確認のため挿入した `print` 文を後で削除する必要があり、デバッグエラーにつながる可能性がある[1]。プログラムの内容を理解するための支援ツールとして、Python Tutor[2]や IDE で実装されているブレークポイント機能などがあるが、プログラミング初学者が使うにはツールの操作が複雑で時間や手間がかかる。本研究の目的は、プログラミング初学者がサンプルプログラムの内容を理解することへの支援を行うことである。

本論では、プログラミング初学者のためのプログラム内容理解支援ツールに必要なと考える性質を論じ、その性質を反映させてプロトタイプ中のツールを説明する。

## 2. 提案する初学者支援の枠組み

本節では、プログラミング初学者のためのプログラムの内容理解支援ツールに求められる性質を考察する。

### 2.1 ソースコードを崩さない

元々のサンプルプログラムに対して、`print` 文の挿入と削除など、ユーザが一切の編集を加えずに変数の値を確認できることが望ましい。最近では、ノートブック形式でインタラクティブにプログラミングを行える環境として Jupyter Notebook がある。複数のセルに分割してコードを作成することが可能である[3]。1つのセルで定義された変数は、同じノートブック上であればどのセルからでも扱うことができるため、ユーザはコードを実行した次のセルで変数の値を確認するというサイクルでプログラムの内容の理解を進めることができる。しかし、変数の値

を確認するには元々のソースコードを実行可能な単位で切り分ける必要がある。結果として、必要な情報がノートブック上に散らばってしまい、プログラムの理解も進みにくくなってしまふ。

### 2.2 ステップ実行できる

プログラムを1行もしくは複数行ごとに手動で実行するデバッグ方式であるステップ実行を行えるツールとして Python Tutor[2]が挙げられる。Python Tutor は、Python のプログラムを可視化する Web ベースのツールである。ユーザは Web ブラウザで直接 Python プログラムを書き、実行中のデータ構造の状態を表示するためにソースコードを前後にステップしてプログラムを可視化する。

### 2.3 変数の値の動きを追える

プログラムの内容を理解するには、実行中のプログラム中での変数の値の変化に着目して、ソースコードを追っていけることが望ましい。`for` 文のようなループ文では、繰り返す回数が多ければ多い程、`for` 文の中の変数の値を追うことは難しくなると考えられる。

### 2.4 ピンポイントで即座に調べられる

ユーザが理解できない箇所を発見したとしても、確認するために手間がかかるとその疑問を無視してしまう可能性があり、ソースコードの中で疑問を即座に解消できる必要がある。プログラムが動作しているように見える場合に体系的な欠陥を探すことなく無造作にコードをいじってしまう、コード・アンド・フィックス型のプログラミング[4]につながる。Python Tutor ではユーザが理解できない箇所を調べるためにはソースコードを1行目から順に実行する必要があるため、行数の多いプログラムではその箇所にたどり着くまで時間がかかる。ペアプログラミング中に隣にいるナビゲーターに質問するくらい即座に答えが返ってくるようなインタラクションが望ましい。

### 2.5 簡単な操作で調べられる

プログラミング初学者が使用するツールとして、直感的かつ少ない操作で調べられるように設計する必要がある。Visual Studio などの IDE で実装されているブレークポイントと呼ばれる

A Variable-Centered Program Understanding Support  
Approach for Early Programming Learners

<sup>†</sup>Syuhei Shimoda, Hiroki Yokohama, Kumiyo Nakakoji,  
Yasuhiro Yamamoto, Department of Media Architecture,  
Future University Hakodate

機能を使用することで、プログラムの処理のトレースやバグの調査が可能となる。ユーザはプログラムを一時停止させたいソースコードの行数に入力カーソルを移動させて、ブレークポイントを設定する。プログラムを実行し、ブレークポイントにより処理が一時停止した際に、宣言されている変数の値を確認することができる。Visual Studio ではマウスカーソルを変数に合わせるだけで値の確認ができるようになっている。これ以上操作が複雑になってしまうと、確認すること自体に時間がかかってしまい、学習は円滑に進まないと考えられる。

### 3. プロトタイプツール

上記で説明した性質を考慮してデザインした、プログラミング初学者用のプログラムの内容理解支援ツールについて説明する。

#### 3.1 ユーザインタフェース

ツールの GUI は Python の GUI ライブラリである Tkinter で作成している。プログラミング初学者であるユーザは、調べたいサンプルプログラムのソースコードを自身のフォルダから参照する。テキストボックスに表示されたソースコードを編集することは出来ない。ユーザはそのソースコードの中で理解につまずいた箇所を直接クリックして選択する。確認ボタンをクリックすると、選択した箇所での各変数の値をシステムに尋ねることができる。確認が終わった後は再度ソースコードを読み進めていき、理解につまずく箇所が出てきたらその箇所をクリックで選択し、変数の確認を行う。このプロセスを繰り返すことによって、ユーザはプログラム全体の内容の理解を目指す。

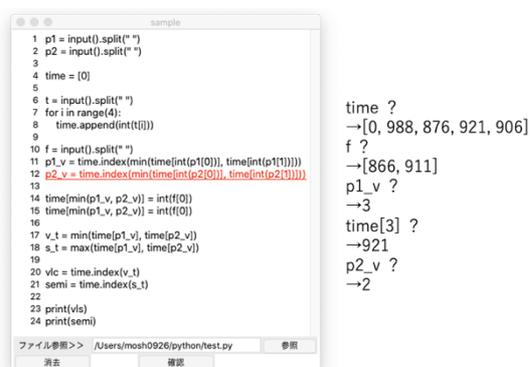


図1 確認ボタンクリック後、選択した行までの変数の値をシステムに尋ねる

#### 3.2 機構の実装

システムの処理の流れを説明する。GUI は、ユ

ーザが確認ボタンをクリックした時に参照されている py ファイルと、ユーザが選択した箇所の行番号を、IPython カーネルを利用するプログラム実行部に渡す。プログラム実行部は、受け取った py ファイルを読み込み、ユーザが指定した行の直後に「sys.exit()」メソッドを挿入する。「sys.exit()」メソッドは SystemExit 例外を発生させ、これが読み込まれた時点でプログラムの実行を停止する。「sys.exit()」メソッドを挿入する際、for 文や if 文などのブロック構造の中に挿入される可能性も考慮し、プログラムが実行可能になるようなインデント数を計算し挿入する。

「sys.exit()」メソッドが挿入された py ファイルは、変数の値を保持できるノートブック形式で実行するために ipynb ファイルに変換する。変換した ipynb ファイルを外部で起動した IPython カーネルで実行すると、「sys.exit()」メソッドが挿入された箇所で実行が停止される。IPython カーネルにはプログラムが停止された箇所までで計算された変数の値が保持されており、WebSocket を用いて IPython カーネルと通信することで、ユーザが確認したい変数の値を IPython カーネルから取得することができる。

### 4. 終わりに

現在、ユーザの入力部分の GUI のプロトタイプやシステムの機能部分は完成している。今後は、変数の値をユーザにどのように提示するかを検討し、ユーザに対しての変数の提示の仕方や UI の改良をする予定である。

### 参考文献

- [1] Andrew J. Ko, Brad A. Myers. 2004. Designing the Whyline: A Debugging Interface for Asking Questions about Program Behavior. CHI, Volume6, Number1, pp.151-158.
- [2] Philip J. Guo. 2013. Online python tutor: embeddable web-based program visualization for cs education. ACM, SIGCSE'13, pp. 579-584.
- [3] 桑田喜隆, 石坂徹, 政谷好伸, 長久勝, 横山重俊, 浜元信州. 2018. クラウドを利用した対話的なプログラミング教育環境とその評価手法の提案. 人工知能学会, 第23回知識流通ネットワーク研究会, p. 6.
- [4] Stuart Wray, Royal School of Signals. 2010. How Pair Programming Really Works. IEEE, Computer Society, pp. 50-55.