

SNS の機能を利用したプログラミング学習支援

Programming study support using the review function of SNS

石川 齊*
Hitoshi Ishikawa

丸山 一貴†
Kazutaka Maruyama

寺田 実*
Minoru Terada

1 はじめに

1.1 背景

近年、パーソナルコンピュータの普及や IT の著しい進歩によりプログラムの需要が増加している。そのため、情報系の教育機関におけるプログラミング教育は当たり前のものになっている。また開発環境の設定が容易になったことで、本やウェブサイトなどを見ながら個人で勉強する人も多く見られるようになった。

1.2 問題点

本やウェブサイトを見ながら学習する人は情報系に多く見られる。ここで、授業による学習方法との比較を考えてみる。この学習方法の利点としては、以下のことが上げられる。

- 時間的束縛がない
- 好きなペースで学習可能

しかし、この学習方法では以下のような欠点がある。

- 孤独感および作業感によるモチベーションの低下
- フィードバックを受ける事ができない
- プログラムが自己流の書き方に陥りやすい
- 容易にはなったが、まだ初学者にとって開発環境の設定は難しい

1.3 着目点

プログラミング学習において、授業のように学生同士で相談できたり、プログラムに対するフィードバックを受けられることは非常に重要だと考えられる。

本研究では、授業のように学生同士で教え合ったりできるプログラミング学習支援はできないのかと考えた。そこで、SNS の機能を取り入れた Web サイト型の学習システムを構築することで、ユーザ同士で問題を出し合ったり、プログラムソースに対して閲覧や相互評価ができるシステムを実装することを考えた。

プログラミング初学者にとって、最初は本などに書かれているプログラムなどを真似していくことで、書き方や使い方を覚えていく。そのため他人のプログラムを閲覧することは、書き方を真似するだけでなく新しいアルゴリズムの勉強にもなるので、本やサイトを見て勉強するよりも優位性があると考えられる。さらに初学者はモチベーションが維持しづらい傾向があるので、自分で勉強した分がス

コアやランキングに反映して返ってくるのはモチベーション維持の一つとして有効だと考えられる。

1.4 目的

本研究では、全てのユーザが簡単に問題投稿や評価を行うことができる環境を構築することで、教師が不在でも、ユーザ同士でシステムを正しく使い学習していくかどうかについて検証する。

2 関連研究

2.1 Social Networking Service

Social Networking Service(SNS) とは、人と人とのつながりを促進・サポートする、コミュニティー型のサービスのことである。また、そのようなサービスを提供するウェブサイトも含まれるので別名 Social Networking Site と呼ばれる。SNS は 2002 年の Friendstar などの登場により本格的に普及し、今では Twitter, Facebook などが有名である。SNS の基本的な機能として以下のものがある。

- プロフィール機能
- ユーザ検索機能
- メッセージ送受信機能
- ユーザ相互リンク機能
- コミュニティ機能
- レビュー機能

2.2 WBT を利用した学習支援

2.2.1 概要

廣林らの研究 [1] は Web サイトを使った非同期型の学習システムの構築である。このシステムは学生が Web サイトを通してビデオ教材などの時間的束縛がない教材を使い好きな時に授業を受けることができる。また Web サイト型なので、プログラムの実行環境の構築の必要がなく学生同士がお互いに学習状況を確認し合える利点がある。

2.2.2 本研究との差異

廣林らの研究は教師が生徒との答案を評価をするだけでなく、よりプログラミングを理解させるために生徒同士で教え合うことができる環境を整えている。しかし解く問題は教員が考える必要があり生徒同士で問題を出し合うことができない。

本研究では、ユーザが独自に問題やテストデータを投稿することができるので、多くの問題を生成することができる。その結果、良い問題が生まれる可能性が高くなると考えられる。

* 電気通信大学 The University of Electro-Communications

† 東京大学 情報基盤センター Information Technology Center, The University of Tokyo

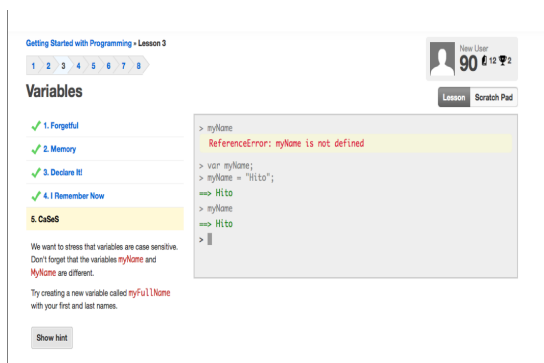


図1 codecademy (出所)http://www.codecademy.com

2.3 codecademy

2.3.1 概要

codecademy^{*3}は Javascript の勉強を対話的に行っていく学習サイトである。実際の学習の様子を図1に示す。説明に従って真ん中のスクリプト画面にコマンドを入力していくことで学習していく。またレッスンをクリアしていく毎に右上にあるスコアなどが変動していき、ある一定のスコアに達するとご褒美としてオリジナルの画像を手に入れる事ができる。

2.3.2 本研究との差異

codecademy では対話的にプログラミングを勉強していくため、ユーザ同士が問題を出し合ったりなどの協力関係を持つ事が出来ない。

本研究では、ユーザ同士が問題を出し合う事で常に新しい問題を解く事が出来る。また自分のプログラムに対してのフィードバックを受ける事が出来るので、協力関係を作りやすくしている。

2.4 自動テスト機能をそなえたプログラム提出システム

2.4.1 概要

望月らの研究[2]は生徒が提出したプログラムの動作確認を自動化するシステムを開発した。提出されたプログラムはそのシステムを通すことで、コンパイルエラーのチェックおよびテストデータによる実行結果を確認することができる。コンパイルエラーが起きた場合は日本語でエラー内容と解決のヒントを提示するようになっている。これにより、教員がいなくてもプログラムの正当性が判定することができ、またテストデータにより見落としやすい例外などチェックを行える利点がある。テストデータ実行時の例を図2に示す。

2.4.2 本研究との差異

望月らの研究では問題に対するテストデータは教師が予め用意しているため、問題の数が増えたり内容が難しくなるにつれてテストデータの用意に時間がかかってしまう欠点がある。

本研究ではテストデータの作成についてもユーザに行わせる。これにより多くのユーザがテストデータを投稿することでテストケースを増やすことができ、より良いテストデータが生まれる可能性が出てくる。

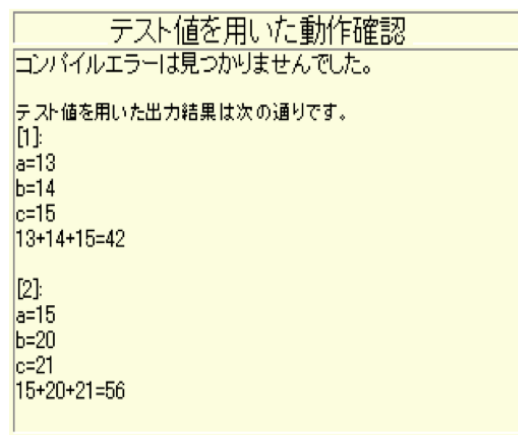


図2 テストデータ実行時の画面 (出所) 参考文献 [2]

3 提案システム

3.1 概要

本研究では、手続き型言語として広く一般的に使われているプログラム言語の Java を学習の対象にして、ユーザ同士が問題を出し合ったりプログラムソースを評価し合う事が出来るシステムを提案する。対象者としては Java の経験が無い人から、Java で「条件分岐」や「反復処理」を使ってプログラムを書ける人を対象とする。また、太田によるライバル認知による学習態度の研究結果^{*4}から、心理的に近い相手に負けたくない傾向を利用して、得点制度とランキング機能を加えることで、動機づけの効果があると考えた。図3はシステムの概要図である。

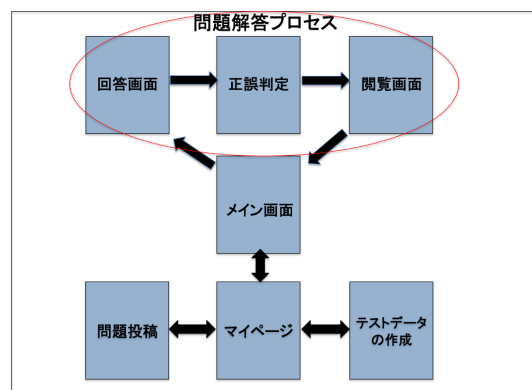


図3 システム全体図

Web サイト型の学習システムなので、環境構築の手間がなく、ソースコードを書くことでフィードバックを受けることができる利点がある。さらに問題を解いたり、投稿することでスコアが手に入り、それによりランキングも変動するので、ゲーム感覚かつ手軽に学習できるように設計している。

3.2 メイン画面

図3に示す通り本システムはメイン画面を軸として様々な画面に移行することができ、また以下のことを行うこと

^{*3} http://www.codecademy.com

^{*4} 太田 伸幸 「学習場面におけるライバル認知に関する研究」 愛知工業大学研究報告, 2004 年

ができる。

- 解く問題を選択して、解答画面に移行
- スコアやランキングの確認
- マイページやヘルプ画面に移行
- ログアウト機能

3.3 問題解答プロセス

図 3 に示す問題解答プロセスには予め、

- 問題文
- テストデータ

この 2 つを用意しておく必要がある。ユーザはプログラムを記述することで、自動正誤判定システムがテストデータを用いてプログラムが正しく書かれているかを判定し、その結果をユーザに表示する。その後、他のユーザのプログラムを閲覧できる閲覧画面に移行する。閲覧画面では他人のプログラム閲覧だけでなくプログラムに対してフィードバックを残すことができ、ユーザ同士がお互いにアドバイスできるようになっている。この流れをここでは問題解答プロセスとして定義する。

3.3.1 解答画面

解答画面を図 4 に示す。ユーザが問題文に沿ってプログラムを実際に記述する場所である。ここで書かれたプログラムは次の正誤判定で正当性を判定される。

図 4 解答画面

3.3.2 正誤判定

ユーザのプログラムをテストデータを用いて、比較を行い正当性を判定している。登録されている全てのテストデータと比較を行い、その結果をユーザに表示する。テストデータとの比較例を図 5 に示す。

またエラーの内容も以下の 3 つのパターンに分けて表示することで、プログラムの修正に役立てられるようにしている。

- タイムアウトもしくは制限したデータ量を超える (無限ループ)
- コンパイル時のエラー
- 実行時のエラー

図 5 正誤判定の出力例

3.3.3 閲覧画面

閲覧画面を図 6 に示す。他のユーザのプログラムソースを閲覧することができる。また他人のプログラムソースに対してフィードバックを残す事ができ、ユーザ同士で評価し合うことができる。

3.4 マイページ

マイページでは自分が書いたプログラムソースとそれに対するフィードバックを見る事が出来る。また「問題投稿」、「テストデータ作成」画面に移行することで、自分でオリジナルの問題や新しいテストデータを作り投稿することができる。

3.4.1 問題投稿

問題制作者は問題文と模範解答のプログラムを記述することで問題を登録することができる。登録された問題は全てのユーザが解くことができ、またテストデータの投稿ができるようになる。

3.4.2 テストデータの作成

テストデータを投稿する問題を選択し、テストデータを作成することで登録される。登録されたテストデータは正誤判定の時に使われる。

4 実装

4.1 開発環境

本システムはプログラムソースの保存や、ユーザ情報の保持のためファイルやディレクトリの操作が多くなると考

図 6 閲覧画面

えた。そこで CGI での構成を考え、プログラミング言語 Perl で実装している。

4.2 テストデータ

テストデータとは、プログラムの正当性を判断するデータのことである。あるプログラムが与えられた時にその動作結果を予め用意することで、予想結果と実行結果を比較することができる。本システムでは、自動正誤判定システムで用いている。

4.3 自動正誤判定システム

自動正誤判定システムはユーザの書いたプログラムが問題文に沿って正しく書かれたものかをテストデータを用いて判定するシステムである。正誤判定の流れを図7に示す。

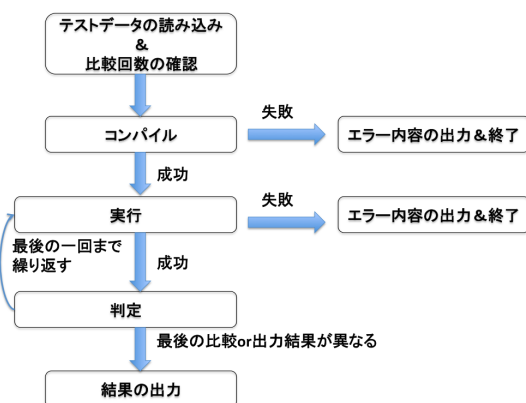


図7 自動正誤の概要

4.3.1 コンパイル

自動正誤判定システムでは、ユーザが記述した Java ファイルのコンパイルを行う。この時に、エラーが起きた場合はユーザにコンパイル時にエラーが起きたことを示し、その内容を表示する。

4.3.2 実行

コンパイルが通った場合は次にテストデータ引数に取り実行を行う。ここで実行時に2つの制限を付けている。一つは実行時間、もう一つはファイル容量である。この二つを制限として実行することで、もしユーザが無制限ループなどのプログラムを書いたとしてもサーバーに大きな負担をかける危険性がなくなる。

4.3.3 テストデータによる比較

テストデータの比較回数は、その問題に登録されているテストデータの数だけ比較を行う。一つでも予想結果と異なる結果を出力した場合は、図5のような結果をユーザに表示する。全てのテストデータを通った場合は最後に比較した内容のみを表示するようにしている。

4.4 テーブルによるデータ管理

本研究ではユーザ情報、問題内容、テストデータなどの多くの情報を扱う必要がある。また問題が投稿された時や、ユーザが回答を行う事でファイルの数はさらに増加していってしまう。そこで、ユーザ情報や問題などの項目毎にテーブルによる管理を行う事で、データの管理を行っている。

4.5 フィードバックの保存

閲覧画面では他人のプログラムにフィードバックを入力することができる。保存方法として Java のコメント機能を利用している。あるプログラムに対して誰かがそのプログラムのフィードバックを入力した場合、その内容をプログラムソース内にコメントという形で書き込む。それにより、ソースの閲覧時にコメントされている場所を図6のフィードバック表示欄に表示している。

4.6 スコア&ランキングの計算

学習面において一つの動機付けとしてスコアやランキング機能を実装している。加算されるスコアに関しては現在の所、問題を解く毎に7点、コメントを入力する毎に2点、問題を制作する毎に25点、テストデータを投稿する毎に10点を獲得できるようにしている。スコアの配分は本システムのゲームバランスととしても重要なので、今後の課題としてスコアバランスの改善が必要である。

またランキングの変動に関しては、ログイン時もしくは得点が入る毎にランキングの計算をする関数を呼び出して実現している。

5 評価実験

5.1 評価方法

実験内容は期日内に被験者に本システムが使えるサイトにアクセスしてもらい、サイト内に書かれている説明などを読みながら使ってもらった。また事前に簡単な問題とそれに対するテストデータを作成しておいた。最後にアンケートに答えてもらった。

5.2 被験者

本大学の学生5名と外部の方1名の計6名に実施した。被験者は全員プログラミング経験があり、被験者の内、2名がJava経験者であった。

5.3 実験結果

5.3.1 被験者の解答状況

事前に4つの問題を用意しておき、被験者がどの問題を解くのかを確認した。用意した問題は以下のものである。

問題1 『Hello』を出力せよ

問題2 1～10までの総和を出力せよ

問題3 二つの整数を引数として与えられた時、最大の方を出力せよ

問題4 0～100までの整数を入力した時に、50未満なら「Low」、50なら「Match」、51以上なら「Hight」を出力せよ

被験者に本システムを使ってもらった結果、解答状況は以下の表1になった。

表1 被験者の解答状況

問題番号	挑戦した人数	正解した人数
問題1	6	6
問題2	6	6
問題3	2	2
問題4	1	1

5.3.2 投稿

1名の被験者が新しい問題の投稿を行っていた。投稿された問題内容は「3つの入力された整数値を小さい順にソートする」という内容であった。またテストデータの投稿に関しては2名の被験者がテストデータの投稿を行っていた。投稿されたテストデータは、新しく投稿された問題に対して3つ、表1の問題3に対して1つ、問題4に対しては2つのテストデータが投稿されていた。投稿されたテストデータの内容については問題の内容に対して適切なテストデータが投稿されていたことが確認できた。フィードバック機能については、2名が使っていたが、他人のプログラムに対するフィードバックではなく、自分のプログラム内容の説明として使われていることが分かった。

5.3.3 アンケート結果

被験者に本システムを使用後、4段階評価と自由記述によるアンケートに答えてもらった。アンケート結果として以下の表2が得られた。

表2 アンケート結果

評価 (悪1⇔良4)	1	2	3	4	平均
スコアやランキングの効果	1	1	1	3	3.0
自動正誤判定	1	1	4	0	2.5
閲覧機能の利用	0	0	2	4	3.7
フィードバックの必要性	0	1	3	2	3.2
問題作成	0	0	3	3	3.5
開発環境の手間	0	1	2	3	3.3
使いやすさ	3	2	1	0	1.7
学習支援の効果	0	0	4	2	3.3

5.3.4 良い点

- 得点システムにより点数を稼ぐ仕組みから学習意欲が湧く
- 他人のプログラムを閲覧できたり、フィードバックを受けられる
- 開発環境の手間が省けて手軽

5.3.5 悪い点

- 他人のプログラムを見るために、同じ問題を解かないといけなの面倒
- 間違った問題の投稿を削除する機能がないので、初学者が勘違いしてしまう可能性がある
- ヒント機能がない
- 操作方法が分かりづらい

5.3.6 提案意見

被験者から以下のような意見を得る事が出来た。

- スコアバランスの調整
- ソースやフィードバックに得点が付けられる機能の追加
- 一度解いた問題なら閲覧画面に飛べる機能の追加

6 考察

6.1 学習機能の考察

アンケート結果の表2を見ると、スコアや閲覧機能またコメント機能や問題投稿はプログラミングの学習の手助けになるという意見が多く、平均で3.0以上の評価であった。プログラムを書いたらそれで終わりではなく、他人のプログラムを見る事で新しい知識を得られたり、フィードバックをもらうことができるのは学習面で大事な要素だからだと考えられる。また学習した分、スコアという形でユーザに表示するので、自分の努力が形になって返ってきて実感が湧きやすいといった意見もあった。

だが、自動正誤判定システムに関しての評価は2.5だった。これは、テストデータがどのようにして使われているのかユーザ側が分からないという意見があった。また事前に用意しておいた問題が全て難しい問題ではないので、テストデータと比較することによる有効性を実感させることができなかつたのも一つの要因だと考えられる。

6.2 使い勝手の考察

アンケート結果の表2の質問6,7を見ると、全体的な使い勝手としてはあまり良くないと考えられる。Webサイト上にあるので、開発環境の設定がいらないのはプログラミング初学者にとって使いやすいという意見があった。初学者によくある、開発環境の設定などが分からなくてあきらめてしまう人にとっては、敷居が低いので手軽感を与えられるからだと考えられる。

しかし、システムの使いやすさは1.7と非常に悪かった。これは画面のレイアウトが整理されていないので見づらく、問題などの削除機能がなかったため、間違った投稿や操作ミスを引き起こす作りになってしまったからだと考えられる。また初学者にとってはプログラムの書き方が分からない状態で始めた場合、本やサイトを見ながら問題を解いていかなければならないので、画面を行ったり来たりしなければならないので使いづらいという意見があった。

7 結論

本研究では、ユーザ同士が問題を出し合ったりプログラムを評価し合ったりできるシステムを提案し、解答インタフェースや問題投稿インタフェースを実装した。その結果、5.3.2よりユーザ同士で問題の投稿やテストデータの投稿が行われたので、教師がいなくても他のユーザが教師役を努めてくれるなどのソーシャル的な使われ方がされることが分かった。また、スコアやランキングの機能により、ユーザはゲーム感覚で学習することができるのでモチベーションの維持に役立っていることも分かった。

しかし、フィードバック機能に関してはこちらが意図した使われ方がされていなかった。これはプログラム上にユーザが説明として残したいコメントをシステムの都合上、使えなくしてしまっているため、ユーザがプログラムの説明をする場合に、閲覧画面のフィードバック機能を使ってしまうからだと考えられる。またフィードバック機能を利用した人数が少なかった。原因としては文章によるプログラムの評価はユーザにとって気軽にできるものではないためと考えられる。

8 今後の課題

8.1 投稿機能の改良

現状では問題やテストデータを投稿すると、投稿された問題に対しての模範解答やテストデータの修正が出来ない。そこで、問題投稿者には模範解答の修正する権限を与えたり、またテストデータに関しては、ユーザが投稿されているテストデータの中から任意で使用するテストデータを選べるようにする。これにより間違った解答を修正したり、間違ったテストデータが投稿されてもある程度許容することができるようになる。

8.2 自動正誤判定システムの改良

予め用意したテストデータと比較することでプログラムの正当性を判断しているが、間違ったテストデータを入力されると上手く動作しなくなってしまう欠点がある。改良点としては用意されたテストデータを使うだけでなくユーザがその場テストデータを作成できる環境を整備することで、ユーザが自由にテストデータを選択できるようになる。

また、テストデータはユーザによる投稿なのでユーザ自身がプログラムの間違いなのか、投稿されたテストデータが間違っているのかを判断しなければならない。そこで、その問題に対して投稿されているテストデータの内、何個のテストデータに合格できたかを確認できる機能を付けることで、自分のプログラムの間違いなのか投稿されているテストデータの間違いなのかを判断させる判断材料となるのではないのかと考えている。

8.3 フィードバック保存方法の変更

現状のフィードバックの保存方法は、プログラムソースにコメントの形で書き込むことで保存しているため、ソースの制作時にプログラムの説明として使うコメントを残せない問題がある。そこで、テーブルを用いてソース毎にフィードバックを保存するファイルを管理させることで、ソース制作時にユーザが説明などに用いるコメントを残せるようになり、ユーザ同士での相互評価としての使われ方されるようになるのではないのかと考える。

8.4 スコアバランスの改善

問題を解いたりすることでスコアが増えるが、間違った投稿や正しく実行できないプログラムでも得点が入ってしまう。そこで、間違った投稿したユーザにはスコアを引いたり、フィードバックを多く投稿する人には追加のスコアを与えたりする機能を追加することでユーザが積極的に問題やテストデータ、またフィードバックを投稿してくれるようになる。また毎日ログインしてくれるユーザにはボーナスポイントを与えたりすることで、学習の習慣化や動機付けに繋がられる。

8.5 サーバーの問題

本システムはサーバー上で動かしているので、ユーザ数の増加や機能の充実に伴いサーバーの負担が大きくなる事で機能なくなる危険性がある。現状では対策として効果的な案が浮かばないが、必ず取り組まなければならない重要な問題であると認識している。

またセキュリティの観点からもサーバーの安全を確保しなければならないため、データベース認証やサンドボックスによるセキュリティの整備も今後のシステムの拡張

に伴い重要な課題となってくる。

8.6 評価システムの改善

他人のプログラムを評価する場合にフィードバックを書くのだが、それでは評価する側も気軽に評価することができない。そこで、プログラムやフィードバックに対して5段階評価ができる機能を追加する。それにより誰でも気軽にプログラムを評価することができるようになる。またスコアと連動させることで、良い評価をもらっているプログラム制作者にはスコアを与えたりすることができるようになり、スコアバランスの改善にも繋がる。

8.7 データベースの導入

ユーザが増加するにつれて膨大な数のファイル管理をしなければならないことは自明である。そのためデータベースを構築し導入することで、ユーザ情報の更新や必要な情報の呼び出しが容易になり管理しやすくなる。

8.8 ユーザサポート機能の充実

Javaの初学者にとってはプログラムの記述方法が分からないので、他のサイトを見ながらになってしまい使いづらいという意見があった。プログラム記述時にJavaのメインクラスは自動で記述されるようになったり、ユーザによるヒントの投稿機能を追加することでよりユーザ同士が助け合える環境を整備することで、初学者でも勉強しやすい学習サイトができるのではないかと考える。

8.9 チュートリアル導入

本研究はSNSの機能を含んでいるので、この先機能が増えていきそれによりシステムが複雑になっていくので、ユーザが使いづらさを感じてしまう可能性が出てくる。そこで、新規ユーザでも使えるようにシステムの使い方を簡単に説明するチュートリアル機能を追加や、ページ毎のヘルプ機能を充実させることで、使い勝手が向上すると考える。またJavaを使った事がない初心者のために、Javaの記述方法を載せたページを作成することで、ユーザサポートの充実にも繋がる。

8.10 レイアウトの整理

本研究のシステムではWebサイト型の学習システムなので新規ユーザがすぐに学習できるように単純明快なレイアウトが必要不可欠であると考えられる。

しかし、現状では画面に必要な部品を設置しただけなので、見落したり、どこで何ができるのか分かりづらいという意見があった。そのため、レイアウトの改善が必要である。ボタンなどの配置を整理したり、減らす事である程度直感的に操作できるようなレイアウトにすることで学習の妨げを極力排除していくことが重要である。

参考文献

- [1] 廣林雅史, 赤松徹, 吉田博哉. “WBTによるプログラミング学習支援システムの開発”. 社団法人 電子情報通信学会, ET2008-131, 2009-3.
- [2] 望月将行, 森田直樹, 北英彦, 高瀬治彦, 林照峯. “自動テスト機能を備えたプログラム提出システム”. 2003 PCカンファレンス, pp343-344, 2003.