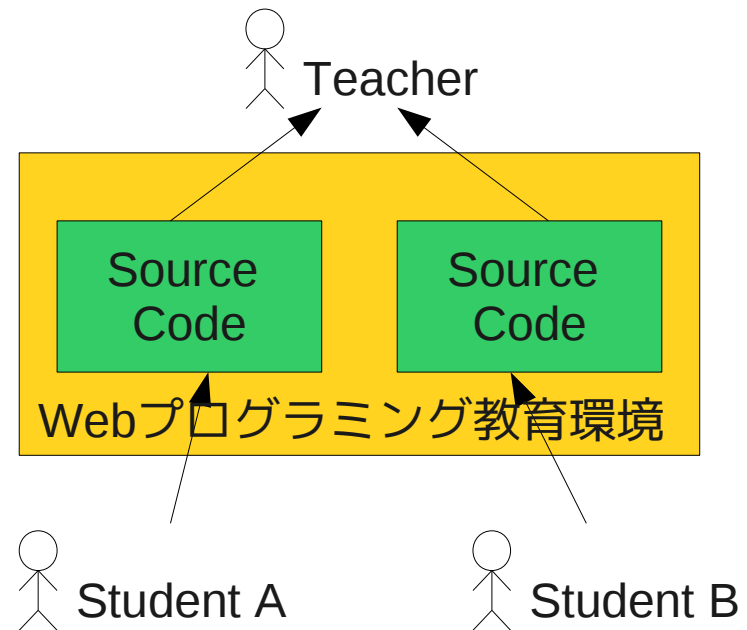
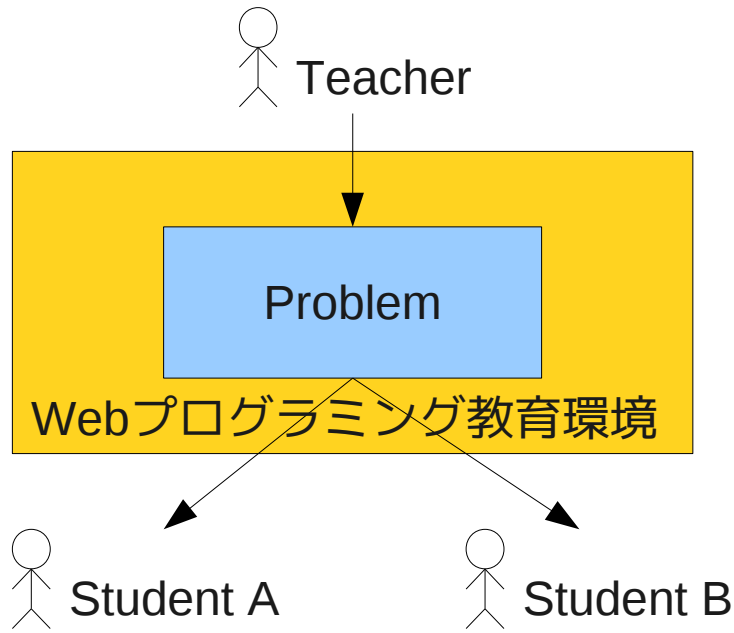


Webプログラミング教育環境 SEKENS

杉本 泰平

背景(1/2)

- ・ 近年、大学等の教育機関を中心に、Webプログラミング教育環境が普及している。
- ・ 特に、教育者が課題を提示し学習者がプログラムを作成し提出する形式の授業で用いられることが多い。



背景(2/2)

- ・ しかし、学習者の評価材料がソースコードのみであるWebプログラミング教育環境では、どの学習者も同等の得点となることが多い。
- ・ さらに、Web プログラミング教育環境のように他者のソースコードが公開されている場合、他の学習者の記述したソースコードを引用すれば、プログラミングの不得意な学習者でも他者と同等の得点を取ることが可能である。

結果、どの学習者も得点が同等となりやすいため、正当な評価が行いにくいという問題があった。

動機

- そこで我々は、
正当な評価を行いやすいWebプログラミング教育環境を構築する必要があると考えた。

目的

- そこで我々は、
 - 引用されたソースコードとそうでないソースコードの区別が可能
 - 引用行為の有無で得点差が表れる
- といった要求を満たすWebプログラミング教育環境を構築する。

提案

- そこで我々は、Web プログラミング教育環境 SEKENS(シーケンス、Social Environment for Konoha Education Networking System)を提案する。
SEKENS は、ソースコード の引用に制限を加え、ソースコード 以外の評価基準を設けることで、教育者がプログラミングの得意な学習者・不得意な学習者の区別を行いやすいプログラミング教育環境である。

要求

- 我々は、SEKENSでは以下の要求を満たすべきであると考える。
 - 引用行為の制限、明示化
 - ソースコード以外の評価材料の導入により、学習者間の得点差を大きくする

設計

- SEKENSでは、既存のWebプログラミング教育環境に新たに以下の2点の要素を追加する。
 - 条件付き引用
→引用行為の明示化、制限付き引用
 - ソースコード以外の評価材料
→引用の有無、引用された数、課題提出まで要した時間

設計1

~条件付き引用:Fork~

- SEKENSでは、ソースコードの引用行為自体は禁止していないが、引用行為に対し、
 - 引用行為を明示的にし、他から引用したソースコードかどうかを判別可能
 - 引用行為に制限を設け、引用を行ってない学習者の得点が高くなるようにする

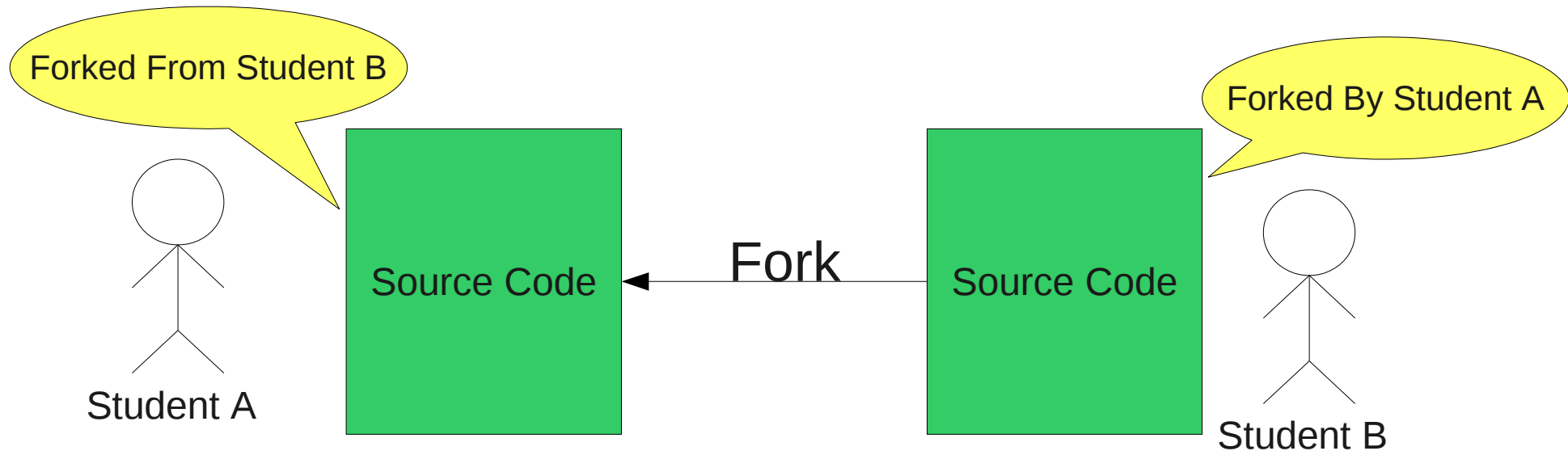
以上の2点の条件を加える。

SEKENSでは、これらの条件下での引用行為をForkと呼ぶ。

設計1-1

~条件付き引用:Fork~

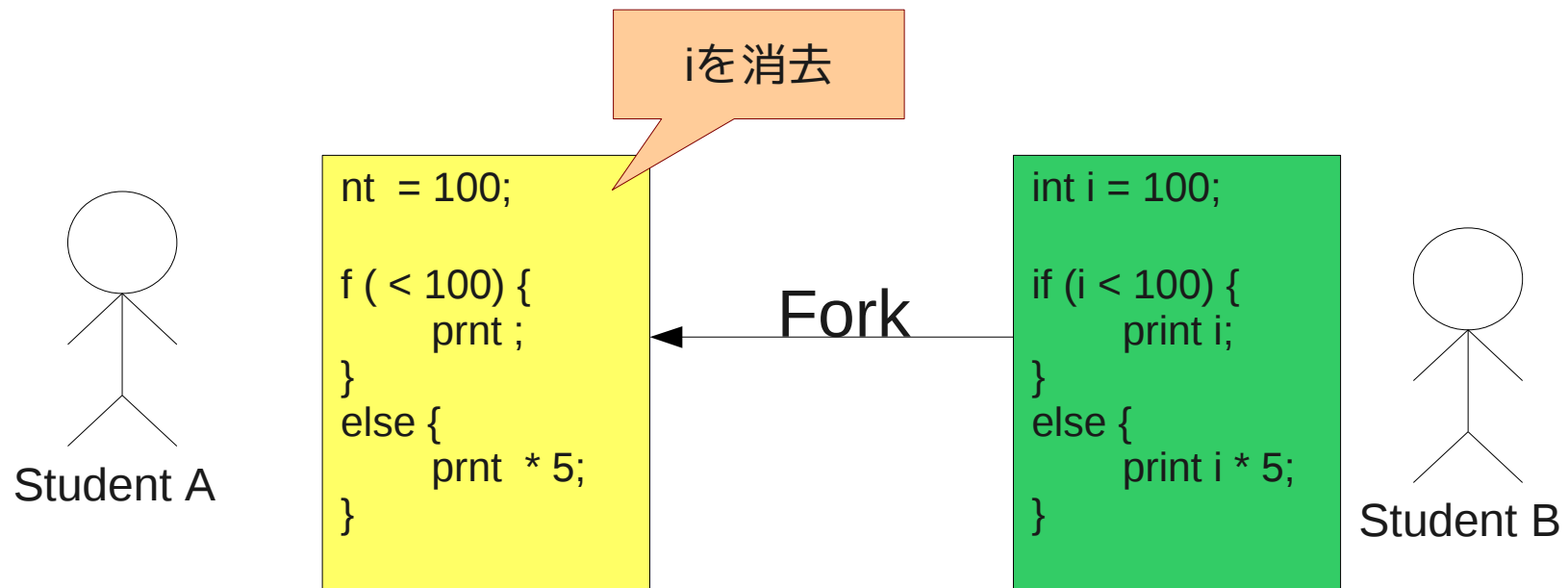
- 引用の明示化：他者のソースコードをコピー＆ペーストできないようにし、引用の際には、引用元・引用先の両者に引用した事が分かるようにし、そのソースコードが引用物か否かを判別可能となる形式で引用できるようにする。



設計1-2

~条件付き引用:Fork~

- 引用の制限：明示的に引用したとしても引用物をそのまま提出することは、無条件でコピー＆ペーストすることと変わらないため、引用したソースコードにバグを混入したり、一部欠損させることで、それを防いでいる



設計2

~学習者の評価方法~

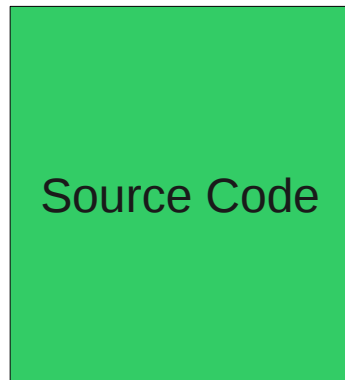
- ソースコードのみの評価材料で学習者の評価を行った場合、特にforkされたソースコードとfork元のソースコードの違いが少ない場合、学習者の得点差が表れにくいため、SEKENSでは以下の評価基準を設けている。

設計2-1

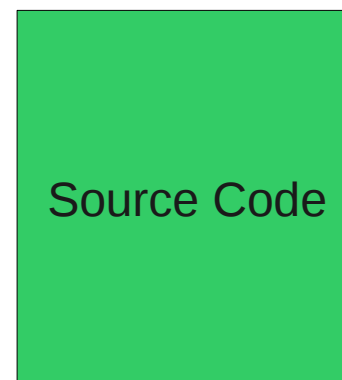
~学習者の評価方法~

- 1) Forkの有無：Forkは、プログラミングが不得意であり、課題が解けなかった学習者が行うことが多い。そのため、Forkの有無を評価基準に含める。

Forked from : **Student A**



Forked from : **none**



=

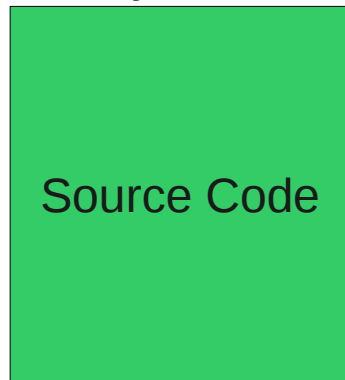
Score < Score

設計2-2

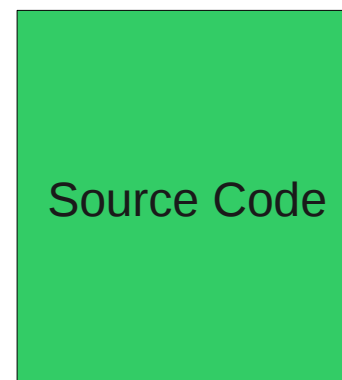
~学習者の評価方法~

- 2)forkされた数：forkされた数が多いソースコードは可読性が高く、質の高いソースコードである可能性が高いことを利用し、forkされた数が多いソースコードを作成した学習者により高い得点を与えることとする。

Forked by 1 Students



Forked by 10 Students



=

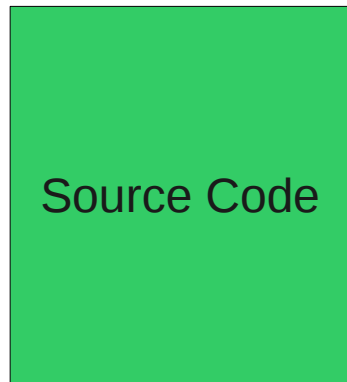
Score < Score

設計2-3

~学習者の評価方法~

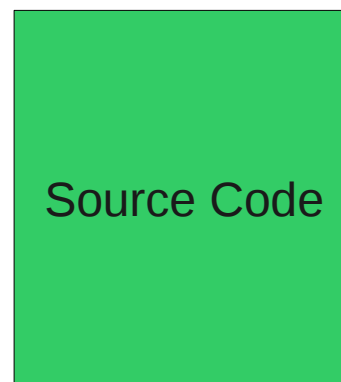
- 3)課題提出までの時間：引用の制限によって、ソースコードをforkしてから提出できる状態にするまでに時間をかけさせることができる。
この評価基準を設けることで、forkを行った学習者とそうでない学習者に得点差が生じるようにしている。

Time : 0 h 15 m 30 s



=

Time : 0 h 10 m 30 s



Score < Score

設計3

～その他～

- ユーザの情報、ソースコードの情報は登録ユーザなら誰でも閲覧できるようにする。
- ユーザは、TeacherとStudentの2種類に分け、StudentはTeacherのグループに参加し、Teacherの出す課題はStudent全員に一斉に送信されるようにする。
- Teacherが課題を提示すればそのグループに所属する学習者全員に課題提出用のディレクトリとファイルが自動で作成される。

設計3

～その他～

- 無条件でコピー＆ペーストができないようにするため、他者のソースコードはテキストでなく、画像データとして表示するようにする。
- forkの引用元・引用先が確認できるよう、forkによる繋がりが分かるようなツリーを作成し、表示できるようにする。
- GUIの部分はシンプルなものとし、ページ繊維を伴わないようにする。

設計 ～外観～

Editor

Peers

Directory

Score

Problem: フィボナッチ数列の和 From: Teacher A

[File: fibo.k](#) Forked From: none Forked By :0 Students

```
int fibo (int n) {  
    if (n < 3) {  
        return 1;  
    }  
    else {  
        return fibo(n - 2) + fibo(n - 1);  
    }  
}  
  
print fibo(10);
```

Run

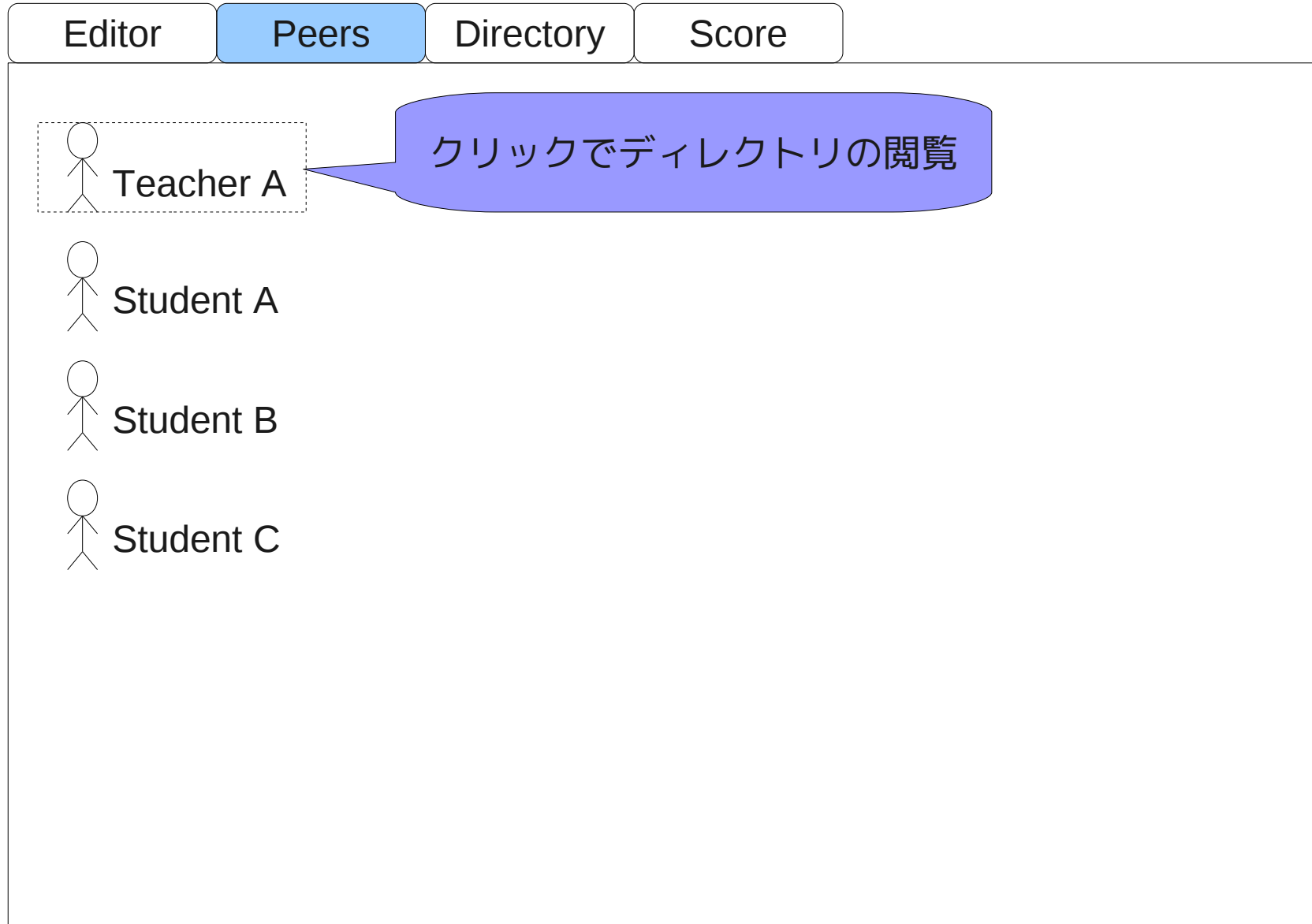
Save

Load

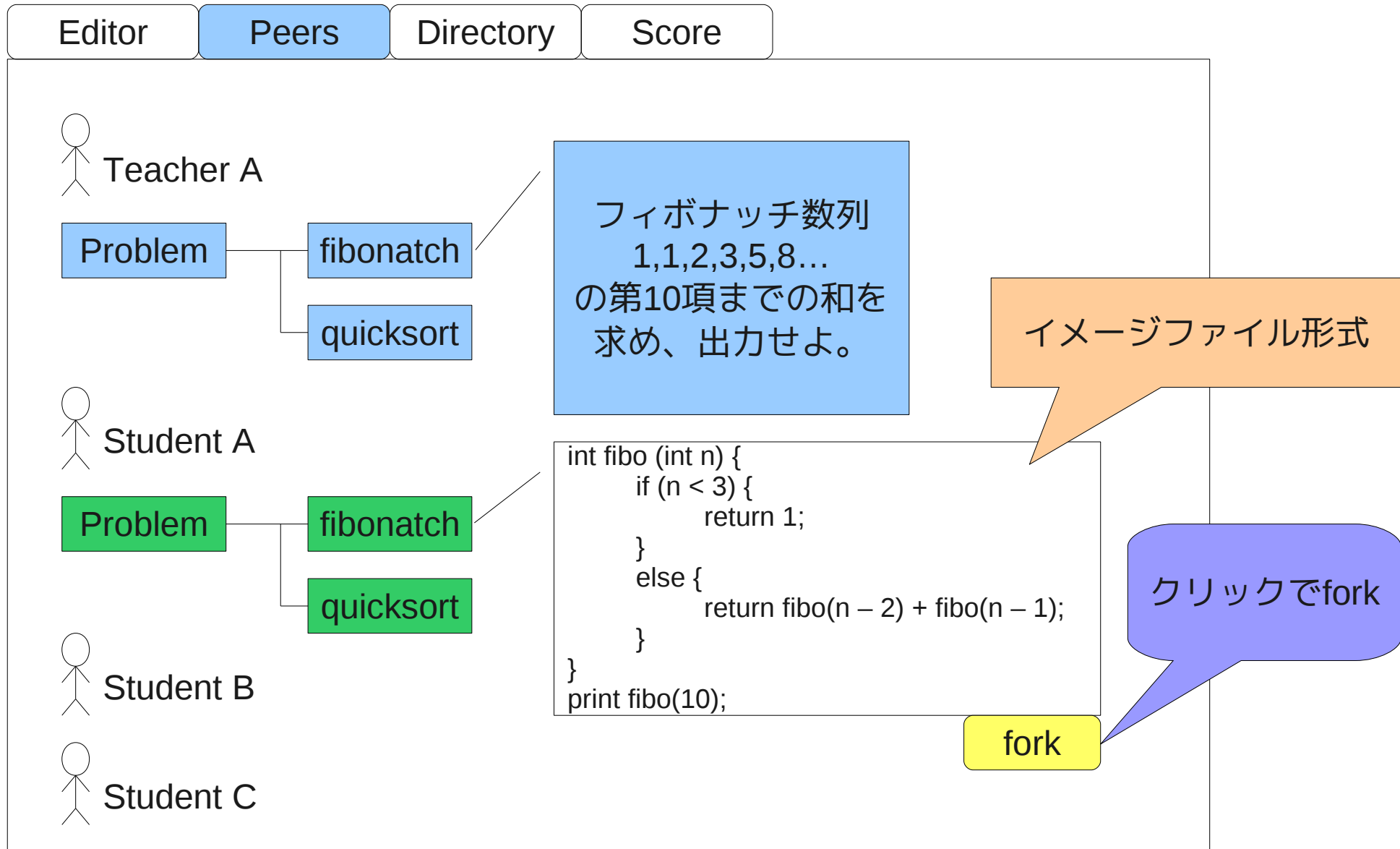
console

```
((eval):1) 55
```

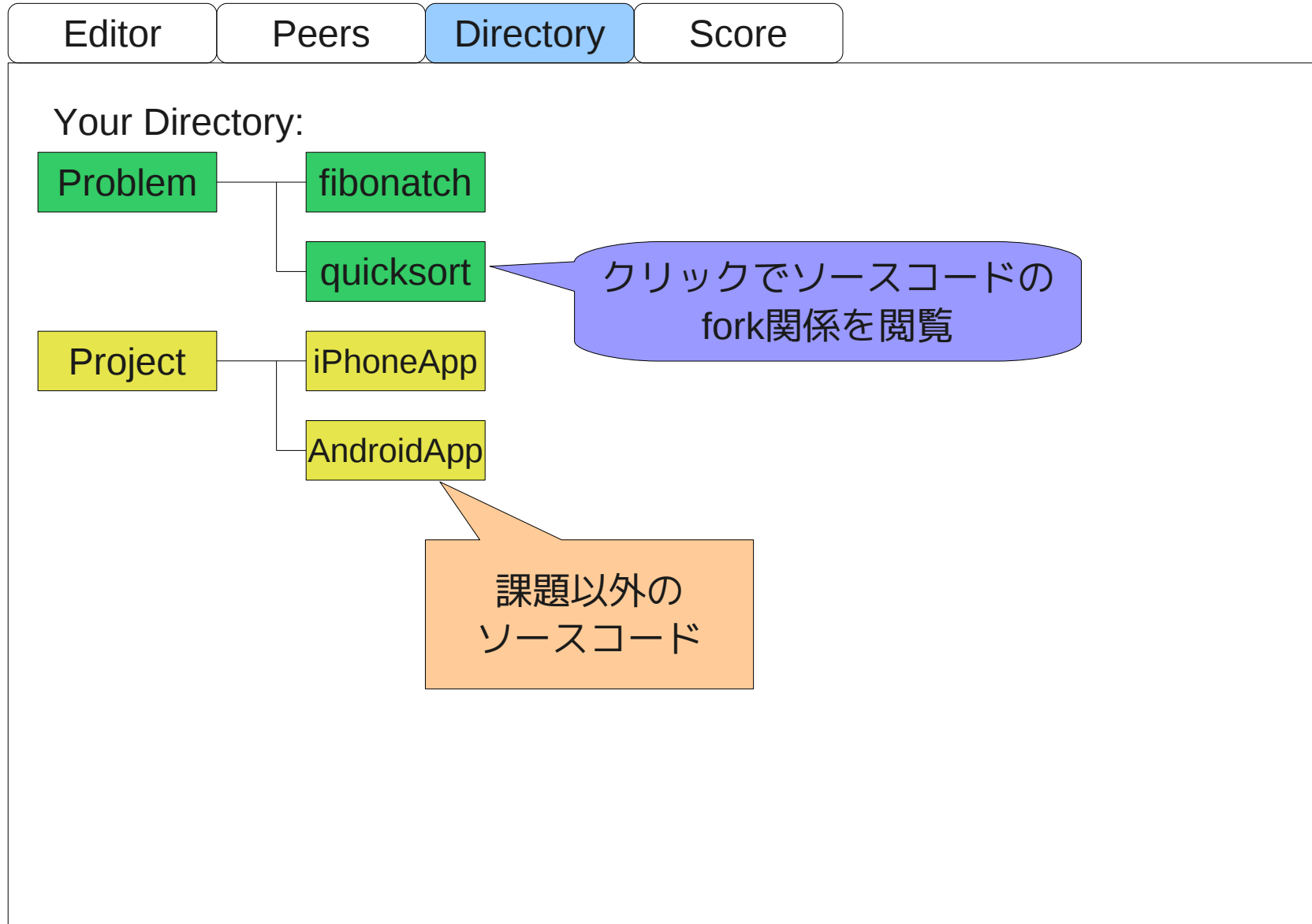
設計 ～外觀～



設計 ～外觀～



設計 ～外觀～



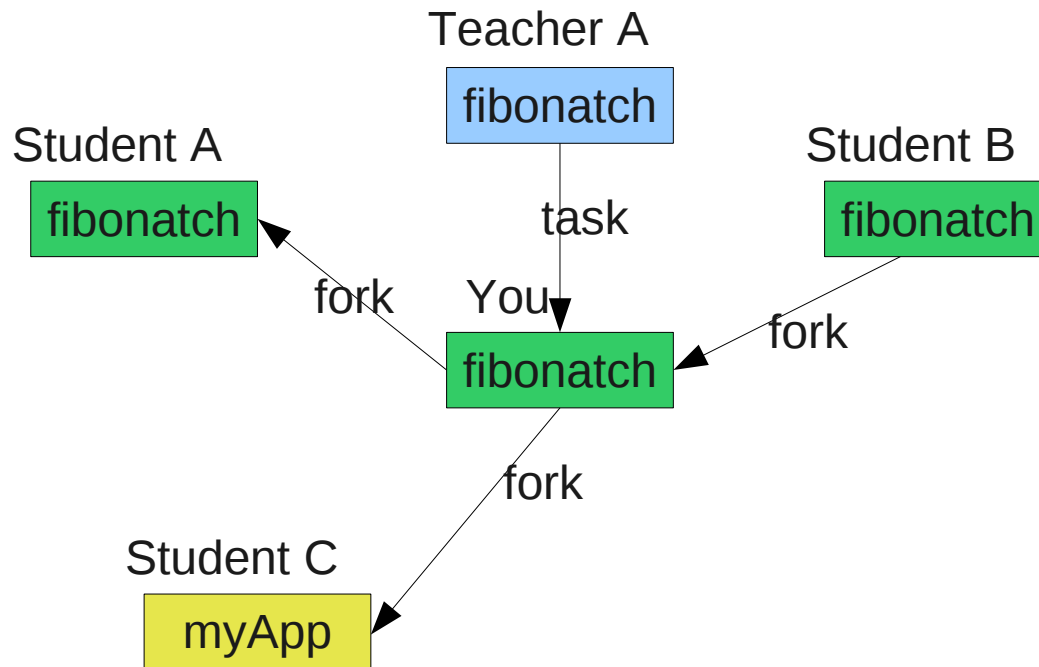
設計 ～外觀～

Editor

Peers

Directory

Score







実装 ～外觀～

Editor	Peers	Directory	Score
--------	-------	-----------	-------

Problem:

BasicScore = $100 \times (\text{Fork:None} = 1.0, \text{Done} = 0.5) \times (1 + \text{Fork Pt} \times 0.1) \times (\text{AvgTime} / \text{Time})$
AvgTime: 0h20m0s

 Student A	· Fork : None · Fork Pt : 10 · Time : 0h10m0s BasicScore = $100 \times 1.0 \times (1 + 1.0) \times (20/10) = 400$
 Student B	· Fork : done · Fork Pt : 0 · Time : 0h20m0s BasicScore = $100 \times 0.5 \times (1 + 0) \times (20/20) = 50$
 Student C	· Fork : done · Fork Pt : 15 · Time : 0h40m0s BasicScore = $100 \times 0.5 \times (1 + 1.5) \times (20/40) = 62.5$
 Student D	· Fork : None · Fork Pt : 20 · Time : 0h10m0s BasicScore = $100 \times 1.0 \times (1 + 2.0) \times (20/10) = 600$
BasicScore ; · AVG : 278.125 · MAX : 600 · min : 50	

クリックで学習者の
基本点を計算

実装

- SEKENSの実装は以下の方法で行う。
 - GUI:Ext Js 4.0ライブラリを使用してJavaScript
 - サーバ側CGIの記述:KonohaScript
 - ユーザ情報の管理:MySQLをkonohaScriptにバインドしパッケージ化したもの、
 - ユーザのソースコード管理:git
- 現在、GUIの部分は概ね実装出来ているが、
- CGI部分(ユーザ情報とソースコード の管理)の実装はまだ出来ていない.

実装

The screenshot displays the SEKENS application interface, which is divided into two main panels: Menu and User.

Menu Panel:

- Account Registration:** Contains three input fields for "User Name:", "Password:", and "Repeat Password:". Below these fields are two buttons: "Register!" and "Login".
- Find:** Contains a "Search Object:" label and an input field for "Object Name". Below the input field are three checkboxes: "Student" (checked), "Teacher", and "Group". A "Search" button is located at the bottom of this section.

User Panel:

- Editor:** The active tab, showing a large "TextArea:" for "Source Code".
- Group:** A tab for group management.
- Peers:** A tab for peer management.
- Data:** A tab for data management.
- Peers:** A second tab for peer management.
- Run:** A button to execute the code.
- Save:** A button to save the code.
- Load:** A button to load the code.
- Console:** A section for output, containing a "Console" label and a text area for displaying results.

SEKENSの外観