

プログラム設計

<http://bit.ly/design4d>

Javaの基礎演習

前期 第6週

2019/5/22

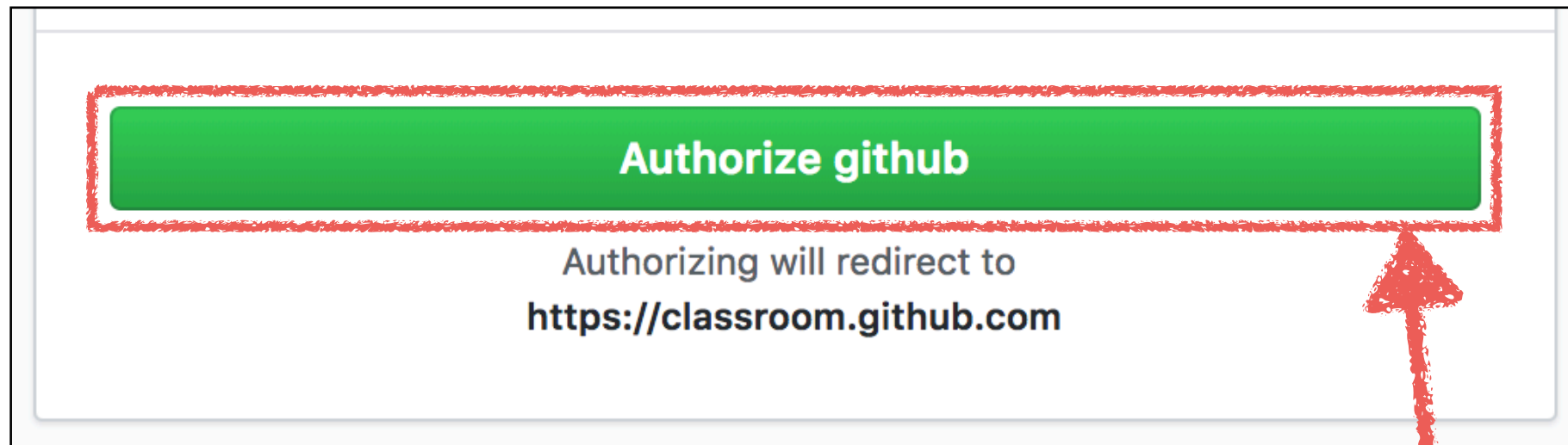
今回は

- ▶ **GitHubを授業で使うための準備**
- ▶ **クラス（第3週）の復習**
- ▶ **GitHubでの提出の練習**

Classroomへの参加

以下のWebサイトにアクセスしてClassroomに参加する
講義資料の配付ページにも掲載

<https://classroom.github.com/a/cEaFm2T5>



クリック

【注】 演習室のFirefoxでは参加できないので、
自分のPCまたはスマートフォンから参加して下さい。

Classroomへの参加

課題名が表示される

Accept the assignment

Accepting this assignment will give you access to the

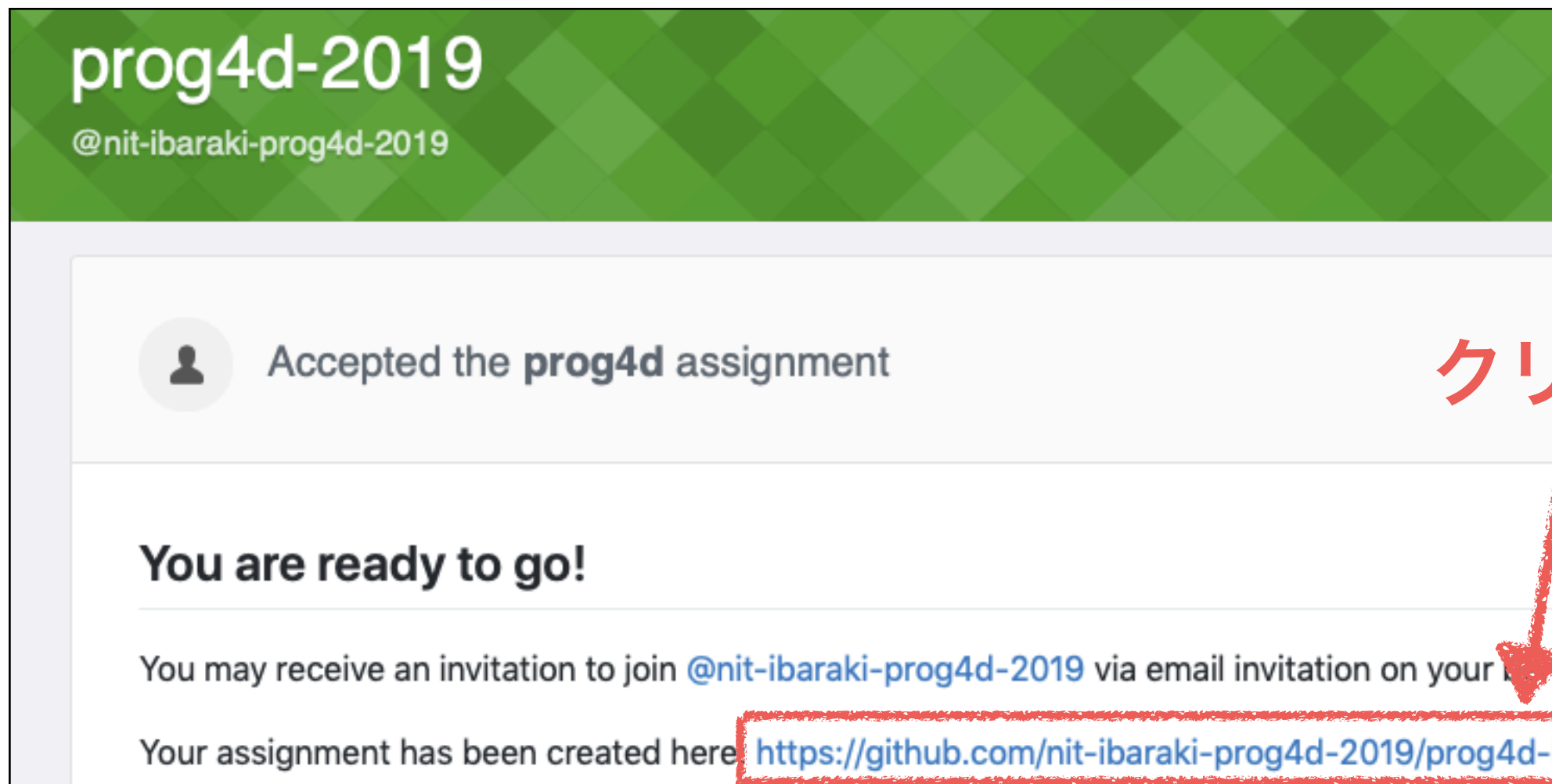
クリック

Accept this assignment

「prog4d」という課題の受け取り完了

課題提出のための設定

「prog4d」という課題の画面を表示する



演習室でのGitHubの準備

今後演習室での手間を少なくできるようにするための準備

端末内で以下のスクリプトを実行して、
自分のGitHubリポジトリをPCにダウンロードする

```
$ mygitclone4d 「自分のGitHubアカウント名」
```

ちなみに、mygitcloneは以下の3つをまとめて実行しています

```
git config --global http.proxy http://po.cc.ibaraki-ct.ac.jp:3128/  
git config --global https.proxy http://po.cc.ibaraki-ct.ac.jp:3128/  
git clone https://github.com/nit-ibaraki-prog4d-2019/prog4d-          .git
```

ここにGitHubアカウント名が入る

演習室でのGitHubの準備

今後演習室での手間を少なくできるようにするための準備

以下のコマンドを実行して、

自分の名前とメールアドレスの設定をGitHubに入れておく

```
$ cd prog4d-(ユーザ名)
$ gedit myconf &
```

ファイルを編集して、自分の名前とメールアドレスを書く

```
#!/bin/sh
git config --global user.name "自分の名前 (ローマ字) "
git config --global user.email "学校のGmailアドレス@gm.ibaraki-ct.ac.jp"
```

演習室でのGitHubの準備

今後演習室での手間を少なくできるようにするための準備

以下のコマンドを実行して、

今回の変更をGitHubにアップロードしておく

1. 端末内で、

```
$ ./myconf  
$ git add -A  
$ git commit -m "myconf追加"  
$ git push origin master
```

2. 以下の自分の課題用リポジトリを開いて、myconfがアップロードされていることを確認する

<https://github.com/nit-ibaraki-prog4d-2019/prog4d->(ユーザ名)

次回からは…

演習室で作業する前に、以下のコマンドを
入れるだけで準備が完了する

```
$ mygitclone4d 「自分のGitHubアカウント名」  
$ cd prog4d- (ユーザ名)  
$ ./myconf
```

※本体をシャットダウンするまでは、
上記「mygitclone」と「myconf」の設定は有効です

【補足】 参照型の変数について

ポインタと同じように参照の概念を持つ変数は、**参照型の変数**と呼ばれます。（p.223）

❖ 配列型の変数（第1週）

❖ クラス型の変数（第4週, 第5週）

❖ インターフェイス型の変数（後の週で扱う）

今週の課題は…

**第3週までに扱った内容で作成することができる
プログラムです。**

【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して**前期第6週のフォルダ**を作る
\$ cd prog4d-(**ユーザ名**) (←既に移動しているなら不要)
\$ mkdir **week106**
\$ cd **week106**

※課題で作るファイル名は各自で決めて構いません。

【課題6-1】

次のような「投票」を表すクラスを作成してください。

- ☐ クラス名はPoll
- ☐ フィールドとして、以下の変数を持つ
 - ❖ String型のtitle（投票タイトル）、choice1（選択肢1）、choice2（選択肢2）
 - ❖ int型のcount1（選択肢1の投票数）、count2（選択肢2の投票数）、count3（その他の投票数）
 - ❖ String型のhistory（投票の履歴）
- ☐ 情報を出力するメソッドshowInfoを持つ

```
void showInfo()  
    //投票の情報を表示する。表示の書式は、課題6-3の実行例を参照。
```

【課題6-2】

クラスPollに、次のような引数付きのコンストラクタを追加してください。

```
Poll(String t, String c1, String c2)
```

```
//● 引数tをフィールドtitle、c1をフィールドchoice1、
```

```
// c2をフィールドchoice2に代入する
```

```
//● フィールドcount1, count2, count3に0を代入する
```

```
//● フィールドhistoryに空の文字列を代入する
```

【課題6-3】

クラスPollのコンストラクタとメソッドshowInfo()の動作を確認するメソッドmainを作ってください。

初期状態の例（動作が確認できれば、下記以外でも良い）

- ❖ 投票タイトル：普段使うWebブラウザは？
- ❖ 選択肢1：Firefox
- ❖ 選択肢2：Chrome

[実行例]

タイトル： 普段使うWebブラウザは？

選択肢1： Firefox

選択肢2： Chrome

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題6-3提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

【課題6-4】

クラスPollに、次のような「投票する」メソッド
voteを追加し、mainで動作を確認してください。

```
void vote(String str)

//strがchoice1と等しい場合、count1に1を加える
//strがchoice2と等しい場合、count2に1を加える
//strがそれ以外の場合、count3に1を加える
//strの内容に関わらず、historyの末尾にstrとカンマ（,）を結合する
//（結果として、投票履歴がカンマで区切られて残される。）
```

【課題6-4】

課題6-3で作成したmainに、以下のように続けて処理する。
voteによる処理の結果は、課題6-5を作ると確認できる。

```
class Pd6poll {  
    public static void main(String[] args) {  
        //  
        //課題6-3で作った処理がここに入る  
        //  
        //voteの動作確認  
        p1.vote("Chrome");  
        p1.vote("IE");  
        p1.vote("Firefox");  
        p1.vote("Safari");  
        p1.vote("Chrome");  
        p1.vote("Chrome");  
        p1.vote("Firefox");  
        p1.vote("Chrome");  
    }  
}
```

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題6-4提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

【課題6-5】

クラスPollに、「集計結果を表示する」メソッド
showResultを追加し、mainで動作を確認してください。

```
void showResult()  
  
    //書く選択肢の投票数と、投票の履歴を表示する。  
    //（表示の書式は以下の実行例を参照）
```

【課題6-5】

課題6-4で追加したvoteの実行後にshowResultを実行すれば、動作が正しいかどうか自分で確認することができる。

```
class Pd6poll {  
    public static void main(String[] args) {  
        //  
        //課題6-3で作った処理がここに入る  
        //  
  
        //voteの動作確認  
        //  
        //課題6-4で作った処理がここに入る  
        //  
        //showResultの動作確認  
        p1.showResult();  
    }  
}
```

【課題6-5】

[実行例]

Firefox: 2

Chrome: 4

その他: 2

履歴: Chrome, IE, Firefox, Safari, Chrome, Chrome, Firefox, Chrome,

【課題の提出】

以下の流れで、作ったCプログラムをGitHubにプッシュして、Webサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題6-5提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

次回の小テスト

次回は、第5週の内容の小テストを実施します。