

# Eclipseユーザーのための Github講座

---



岩瀬優太

# Githubとは？

---

- ・プログラム管理サービス
- ・プログラムコードのクラウドのようなもの

# Githubのメリット

---

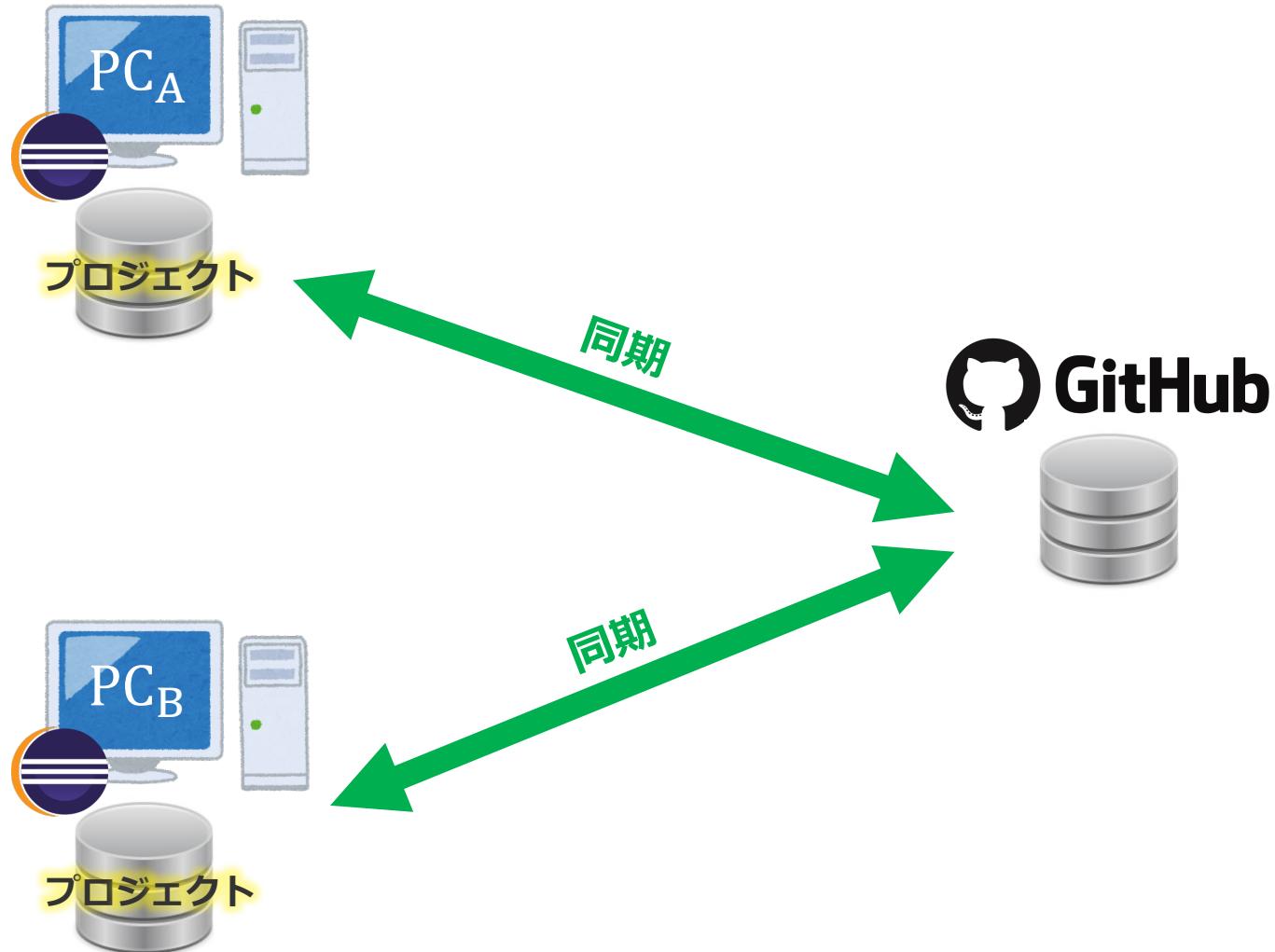
- ・プログラムコードをネット上に**保存・公開**できる。
- ・別のPCからアップしたデータを**同期**できる。
- ・履歴機能でプログラムを前バージョンへ  
**巻き戻せる。**
- ・複数の編集者によって**コードの競合**が起きた場合も  
**解決**できる。
- ・EclipseにはGithubとの連携機能を有し、  
簡単な操作でコードの同期ができる。
- ・Github Pagesという機能で**自作のウェブサイト**を  
**公開**できる。

# Githubのデメリット

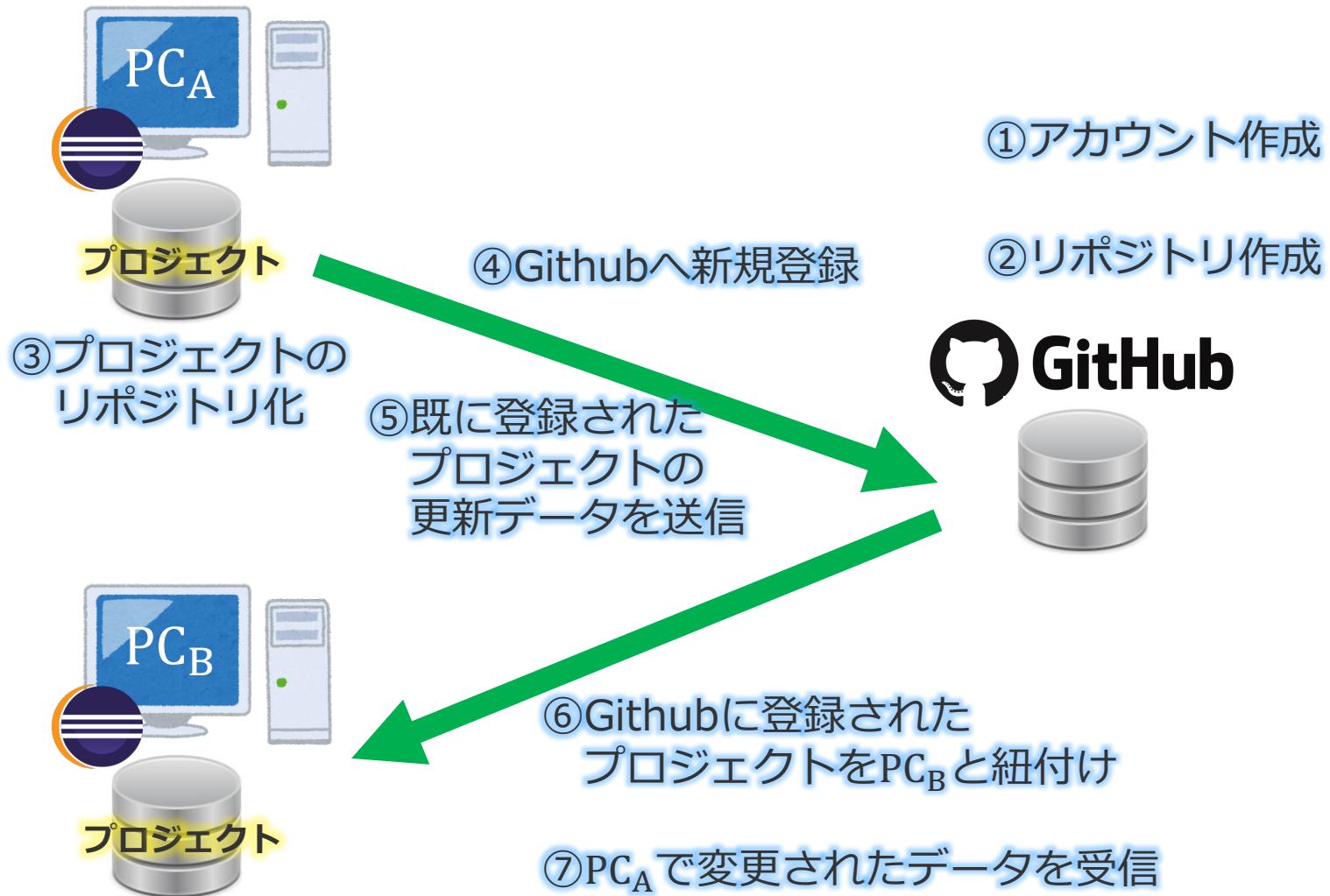
---

- 設定がむだに難しく参入しづらい。
- クラウドと違い自動的には同期されない。  
データの送信・受信は、常に手動で行う必要がある。  
(自動受信する方法は存在するかも？求む情報)
- 無料版ではコードは全世界に公開されてしまう。  
(プロジェクト名をNetworkなどの  
ありきたりな名前にすれば木を森に隠せる。)  
(学生は無償で有料版が使えるという情報も、求む情報)

# データ構造 イメージ図



# 手順



# ①アカウント作成

1. 公式サイト(<https://github.com>)へ
2. ユーザー名、メールアドレス、パスメードを入力  
(ユーザー名は他の人に見られるので  
変な名前にしないように注意)

The screenshot shows the GitHub sign-up interface. It has three input fields: 'Username' containing 'test20181050x', 'Email' containing 'nuzogyonyo@macr2.com', and 'Password' which is redacted with dots. Below the password field is a note: 'Use at least one letter, one numeral, and seven characters.' At the bottom is a large green button labeled 'Sign up for GitHub'. A small note at the bottom right states: 'By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.'

# ①アカウント作成

3. planは「Unlimited public repositories for free.」

Help me set up an organization nextは  
集団用のオプションなので外す

Welcome to GitHub  
Send me updates on GitHub news, offers, and events  
は通知メールを受信するかなのでお好きに

The screenshot shows the GitHub account setup process at Step 2: Choose your plan. The 'Set up a personal account' step is completed. The 'Choose your personal plan' section is highlighted, showing two options:

- Unlimited public repositories for free.
- Unlimited private repositories for \$7/month. ([view in JPY](#))

A note below states: "Don't worry, you can cancel or upgrade at any time."

Below the plan selection, there are three checkboxes, each with a red box around it:

- Help me set up an organization next
- Send me updates on GitHub news, offers, and events

The 'Both plans include:' sidebar lists the following features:

- ✓ Collaborative code review
- ✓ Issue tracking
- ✓ Open source community
- ✓ Unlimited public repositories
- ✓ Join any organization

At the bottom right is a green 'Continue' button.

# ①アカウント作成

4. アンケートはお好きに  
面倒ならskipで

## Welcome to GitHub

You'll find endless opportunities to learn, code, and create, @test20180220y.

Completed Set up a personal account	Step 2: Choose your plan	Step 3: Tailor your experience
--	-----------------------------	-----------------------------------

How would you describe your level of programming experience?

- Very experienced       Somewhat experienced       Totally new to programming

What do you plan to use GitHub for? (check all that apply)

- Project Management       Design       School projects  
 Development       Research       Other (please specify)

Which is closest to how you would describe yourself?

- I'm a professional       I'm a hobbyist       I'm a student  
 Other (please specify)

What are you interested in?

e.g. tutorials, android, ruby, web-development, machine-learning, open-source

Submit      [skip this step](#)

# ①アカウント作成

## 5. ガイドもお好きに

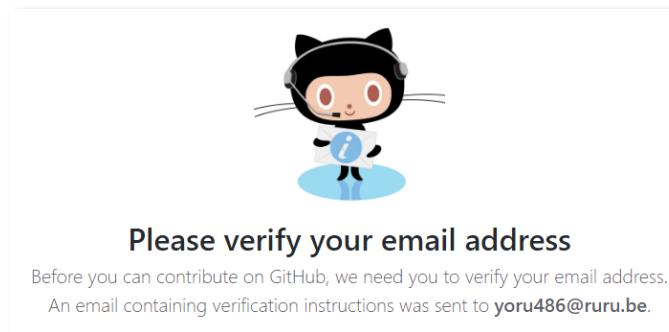
Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

## 6. 確認メールを認証する

*Done!*

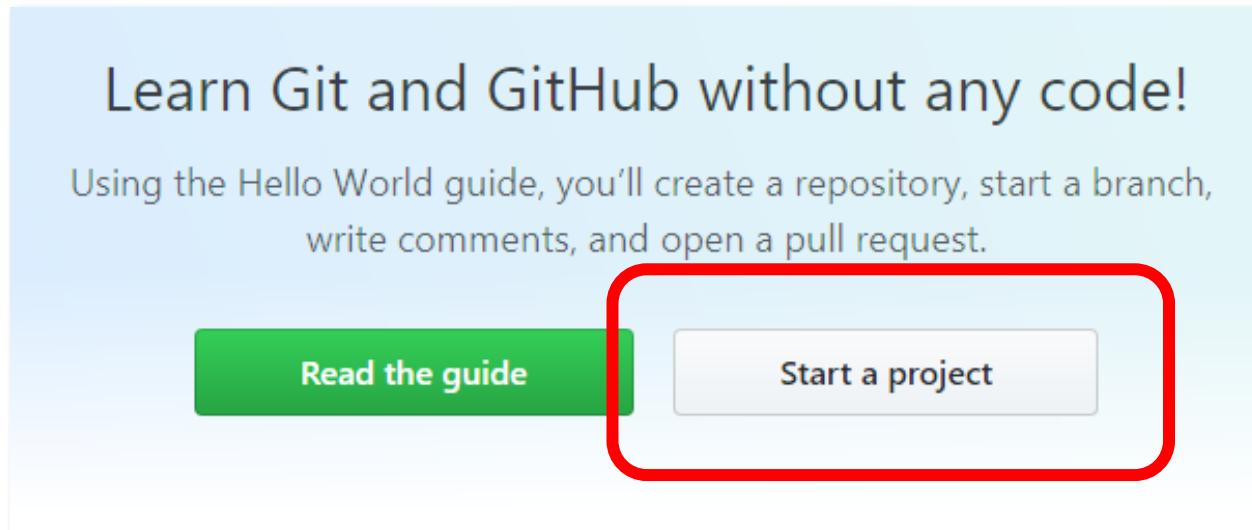


# 手順



## ②リポジトリ作成

1. Githubに飛べばStart a Projectがあるのでこれで作成

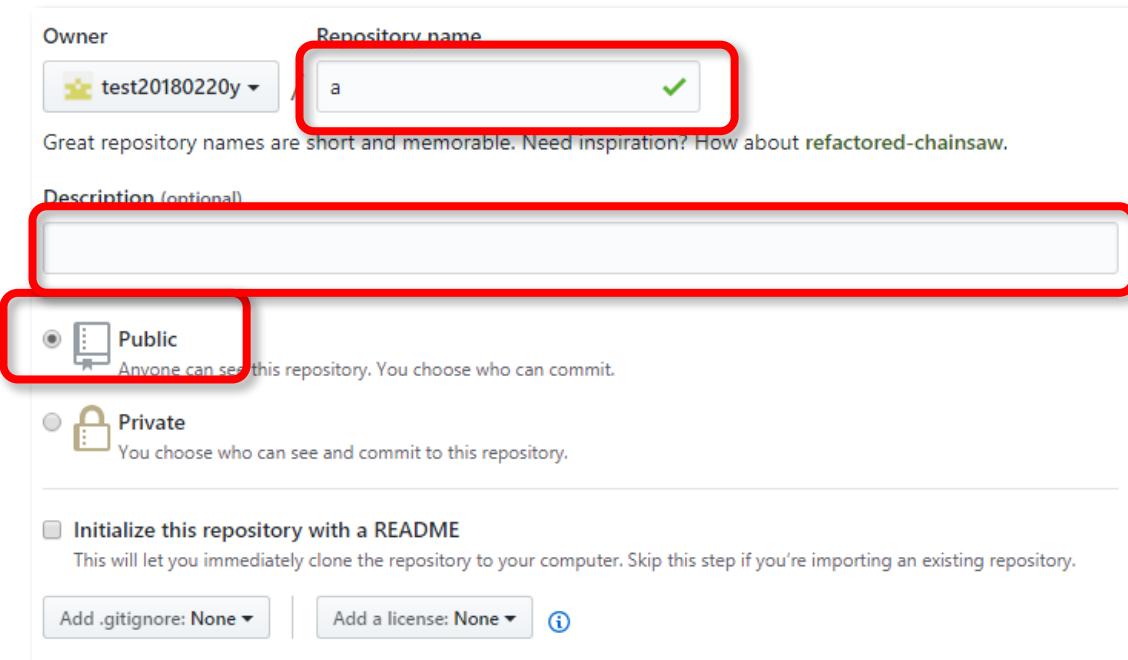


## ②リポジトリ作成

2a. 名前を入力(Eclipseにあるプロジェクト名と同じ方が後で混乱しないと思う)

Description/説明は**お好きに**

公開設定は無料版では**Public**しかできないのでこれで

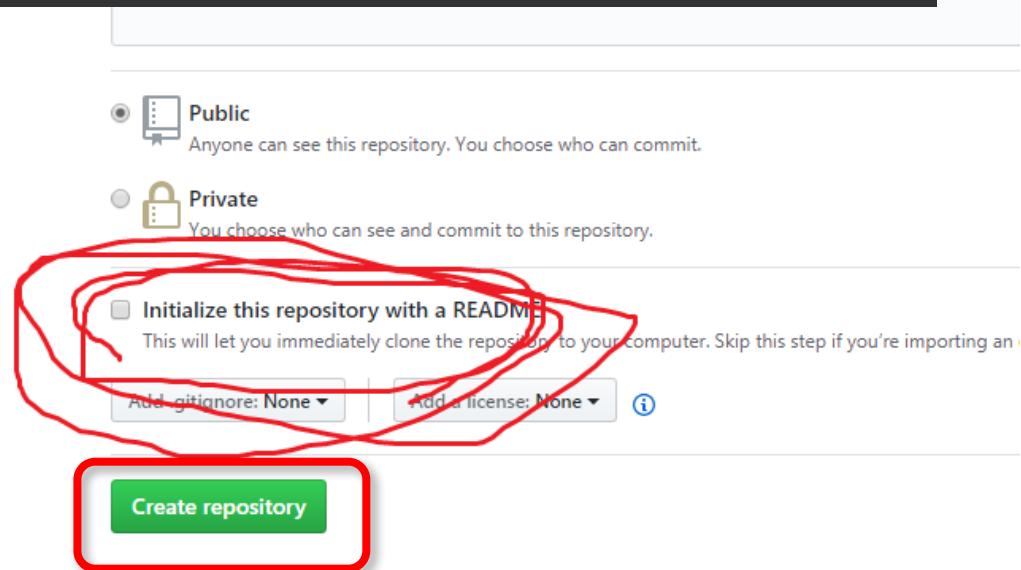


## ②リポジトリ作成

2b. Eclipseのプロジェクトをアップする場合は、

**絶対にInitialize this repository with a READMEにチェックを付けないこと**

3. Create repositoryで  
リポジトリ作成



## ②リポジトリ作成

4. このページが表示されたら完了  
ページを開いたまま次はEclipseの操作

Done!

The screenshot shows a section of the GitHub 'Quick setup' guide for a new repository. It includes options for 'Set up in Desktop' (selected), HTTPS, and SSH, with the URL <https://github.com/test20180220y/a.git>. It also recommends including a README, LICENSE, and .gitignore. Below this, there are three command-line examples: one for creating a new repository, one for pushing an existing one, and one for importing code from another repository.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH <https://github.com/test20180220y/a.git>

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# a" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/test20180220y/a.git
git push -u origin master
```

...or push an existing repository from the command line

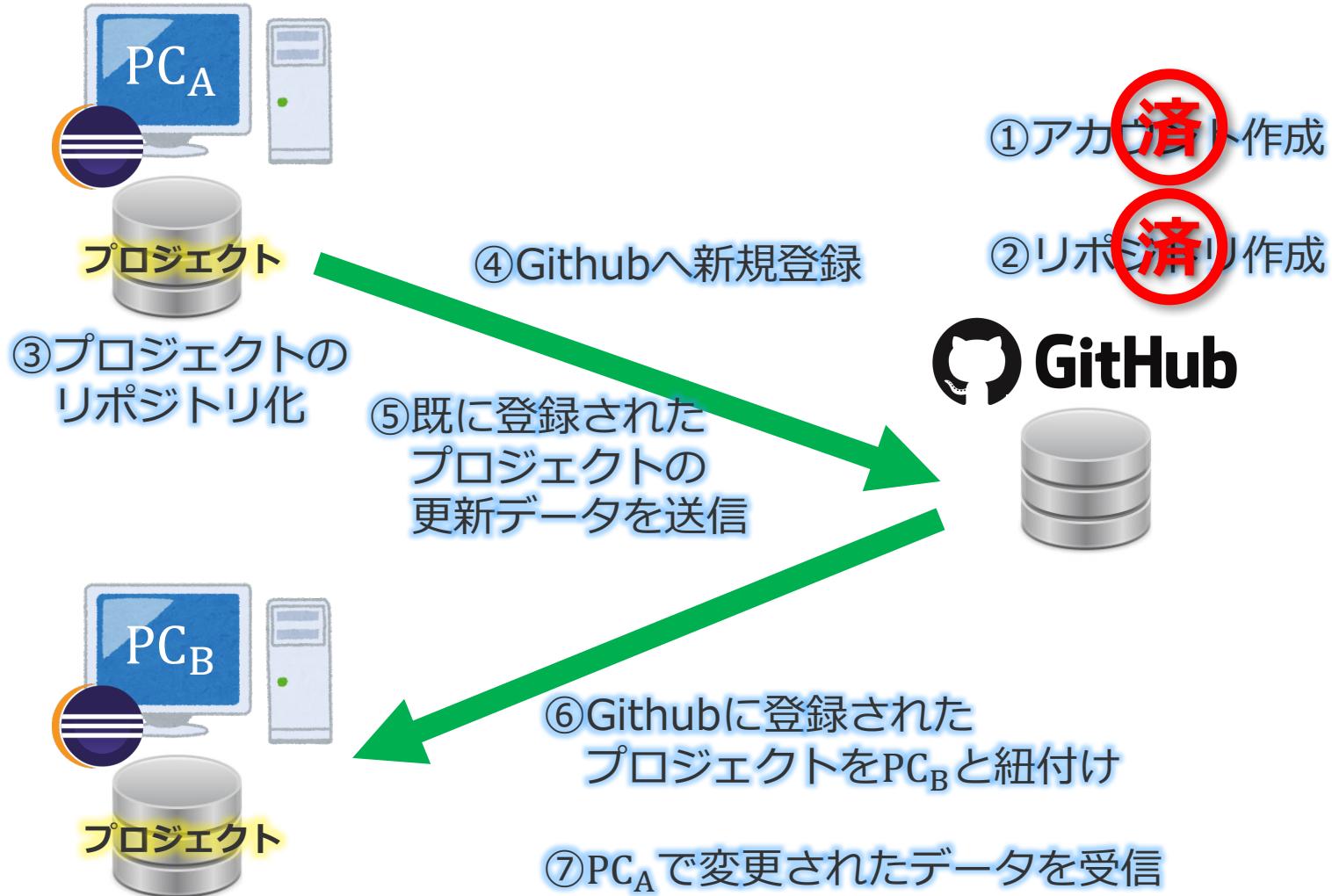
```
git remote add origin https://github.com/test20180220y/a.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Import code

# 手順

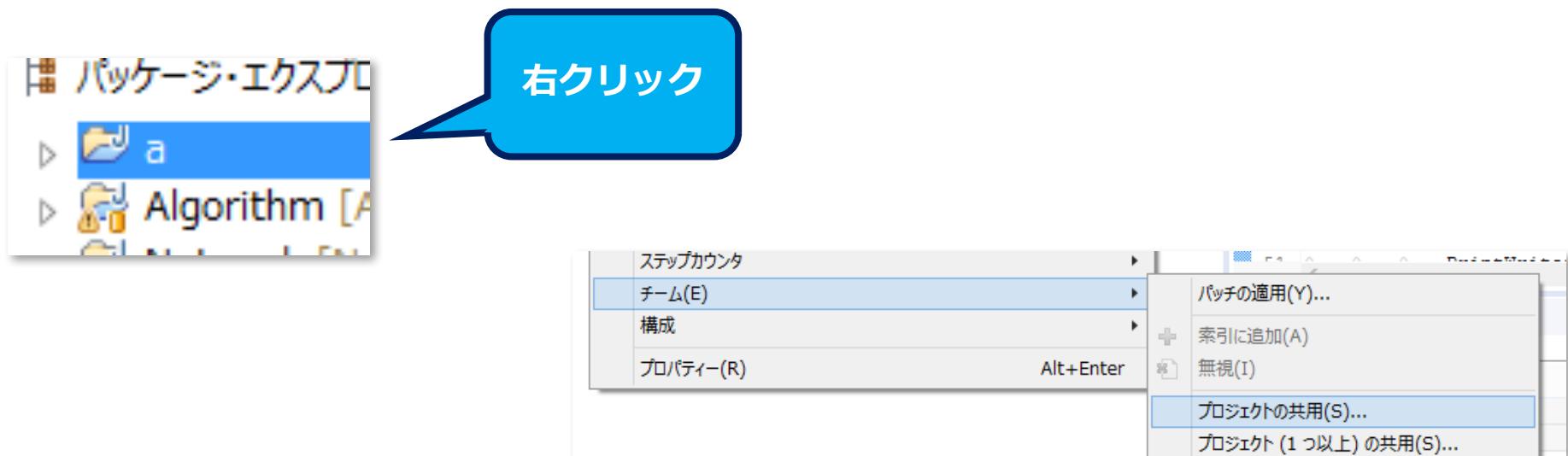


# ③プロジェクトのリポジトリ化

ここからEclipseの操作

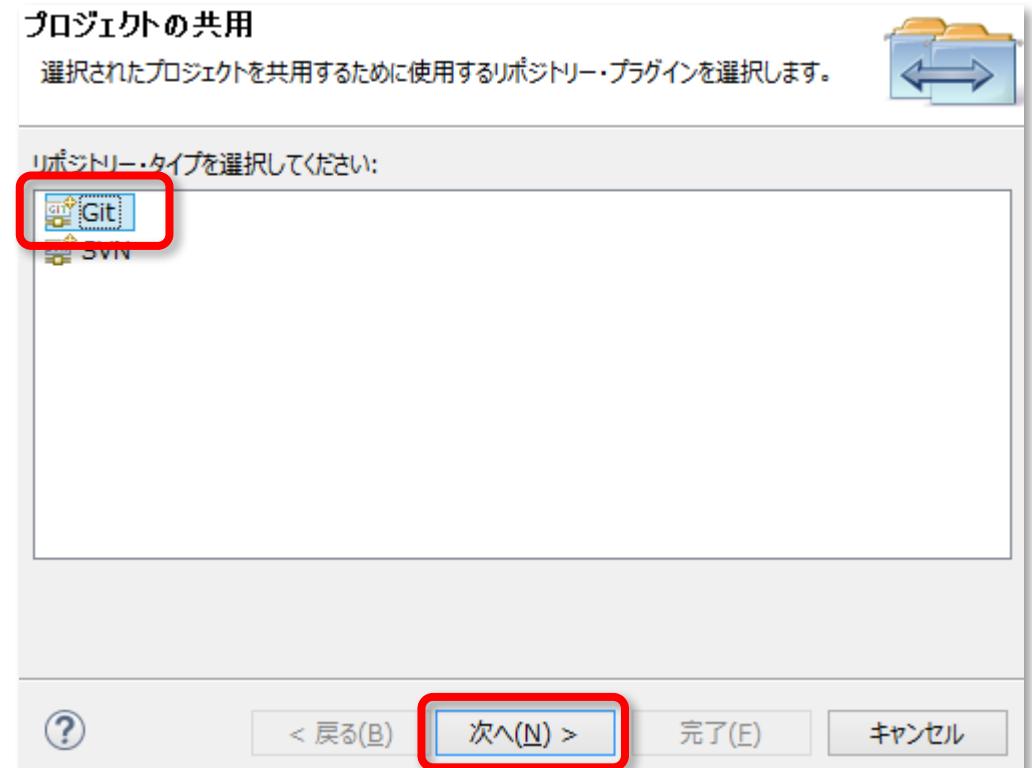
参考: <https://git-manual.net/push-to-remote-eclipse-project/>

- 1a. アップしたいプロジェクトを右クリック
- 1b. チーム->プロジェクトの共用 を選択



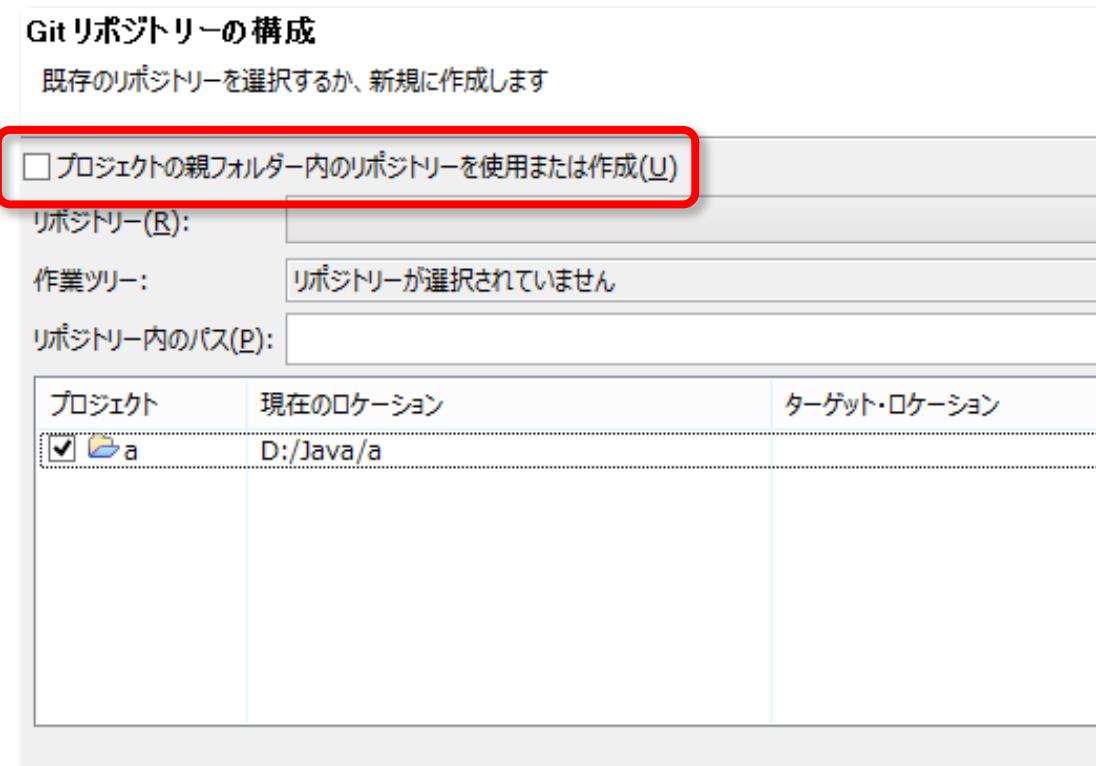
# ③プロジェクトのリポジトリ化

## 2. Gitを選択、次へ



## ③プロジェクトのリポジトリ化

- 3a. 「プロジェクトの親フォルダ内のリポジトリーを使用または作成」をチェック



# ③プロジェクトのリポジトリ化

3b. プロジェクトをクリック  
リポジトリの作成を選択、完了  
(場所はデフォルトの場所でいい)

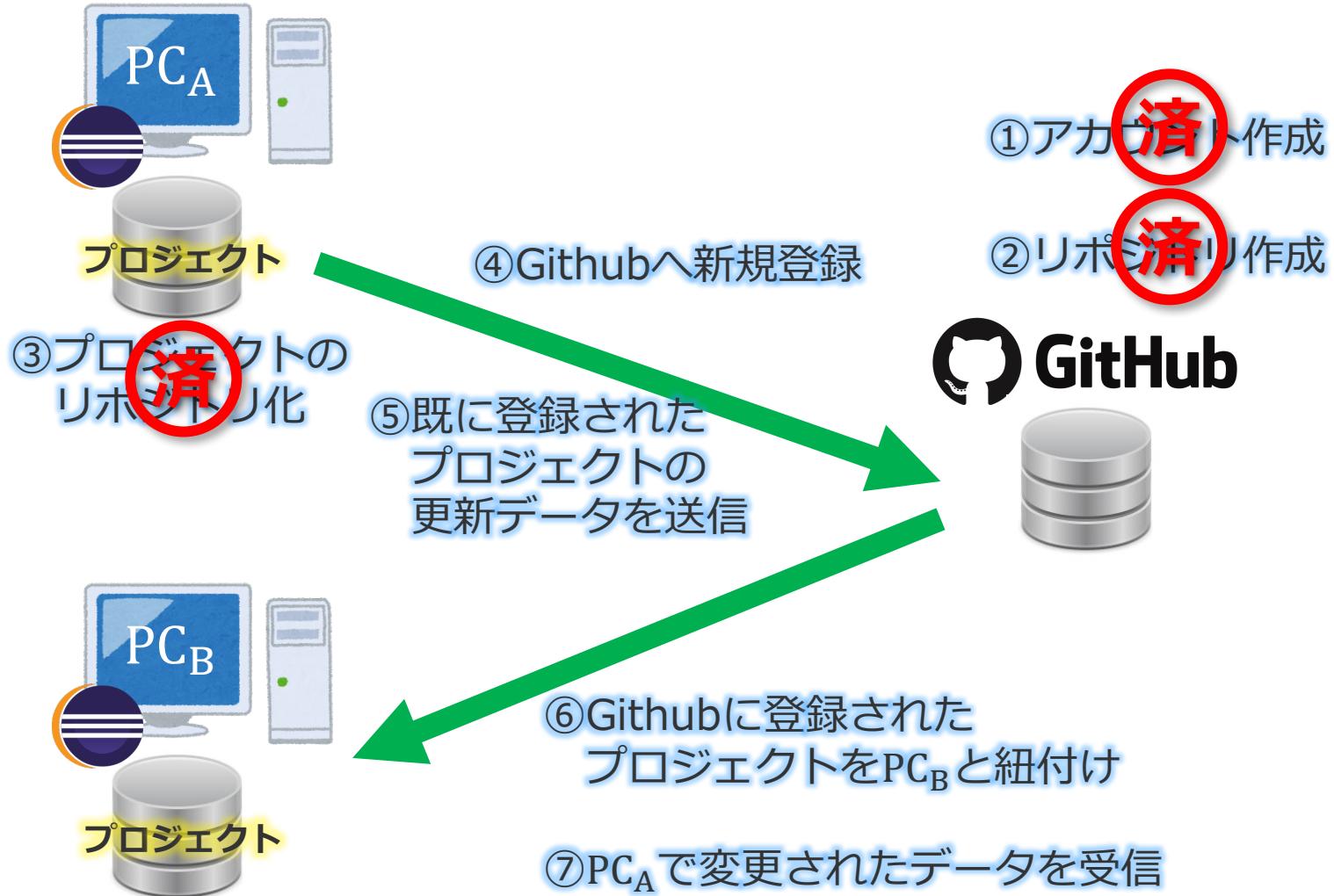
4. プロジェクトのアイコン  
がちょっと変わったら成功



Done!



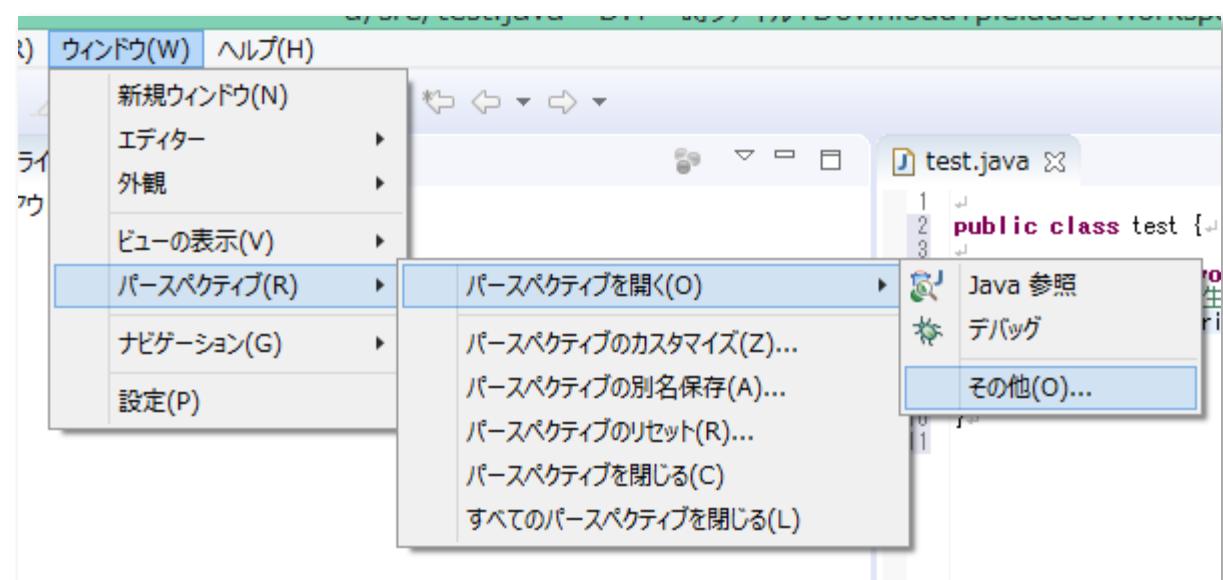
# 手順



# ④Githubへ新規登録

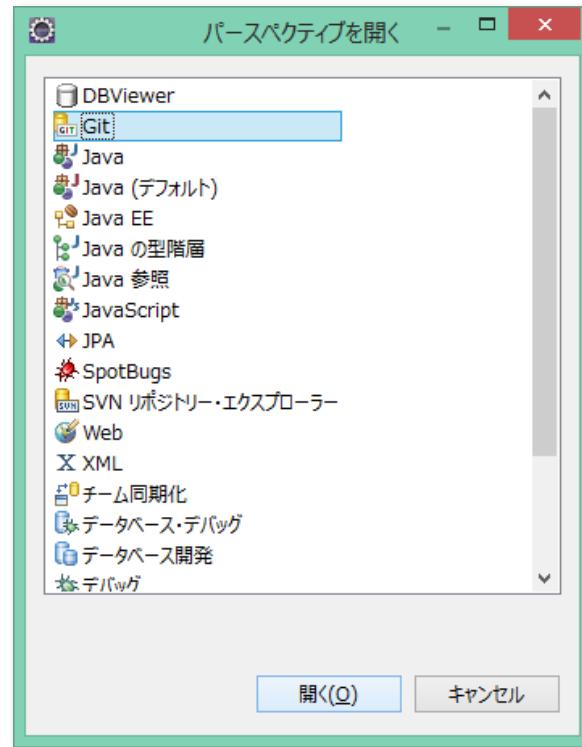
1a. メニューより

ウィンドウ->パースペクティブを開く->その他

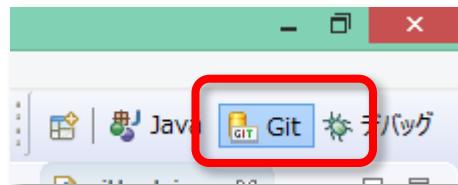


# ④Githubへ新規登録

## 1b. Gitを選び、開く



右上にGitが追加される



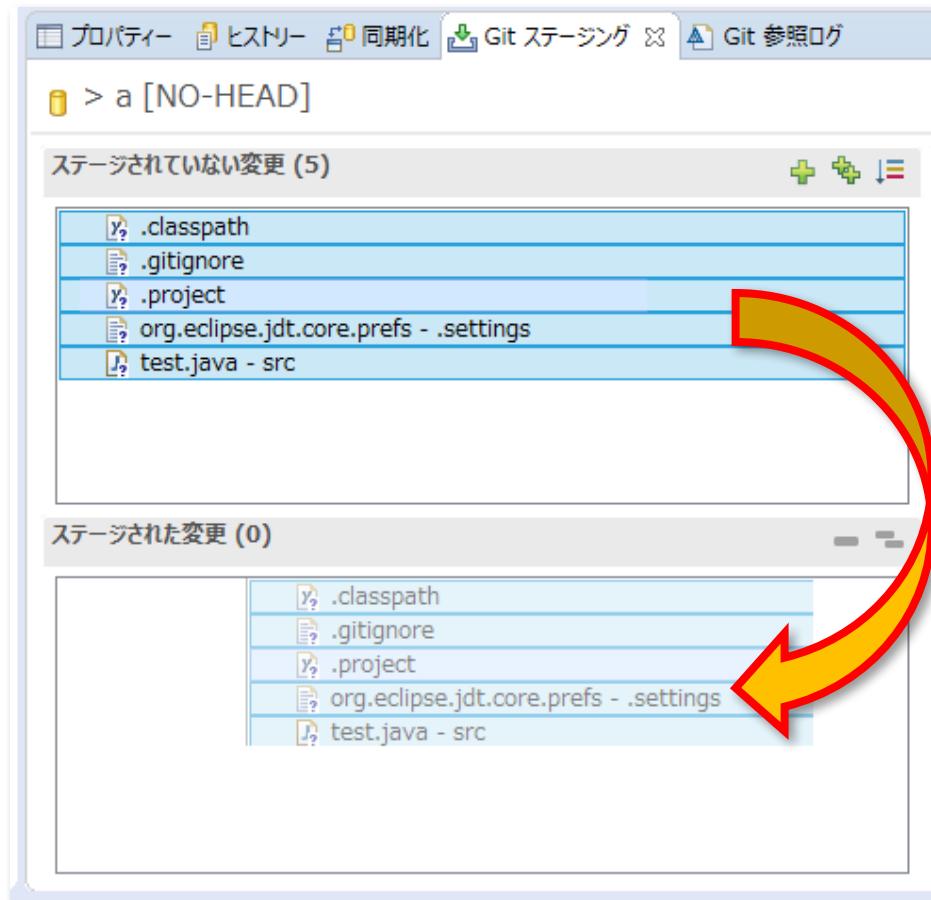
## ④Githubへ新規登録

2a. Gitステージングのタブを開く



## ④Githubへ新規登録

2b. 上のウィンドウのファイルをすべて選択して  
下のウィンドウにドラッグアンドドロップ



## ④Githubへ新規登録

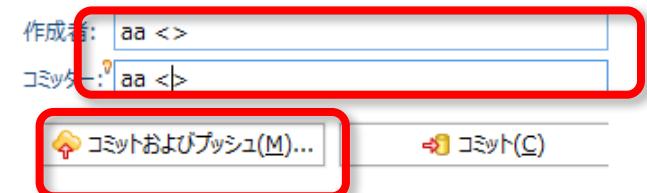
2c. コミット・メッセージに適当に入力する  
(メッセージは必ず何か入力しなければならない)

(初期設定を忘れたので、ここ適当)  
作成者・コミッター情報を打ち込む

名前 <メールアドレス?(空欄可)>  
の形式?



2d. コミットおよびプッシュを選択



# ④ Githubへ新規登録

3a. Githubのページに戻り右上のアドレスをコピーし  
URIの欄へ貼り付け

# 認証欄にgithubのユーザー情報を打つ

セキュア・ストアに保管をチェック  
次へ

Quick setup — if you've done this kind of thing before

 Set up in Desktop or   <https://github.com/test20180220y/a.git>

We recommend every repository include a `README`, `LICENSE`, and `.gitignore`.

...or create a new repository on the command line

```
echo "# a" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/test20180220y/a.git
git push -u origin master
```

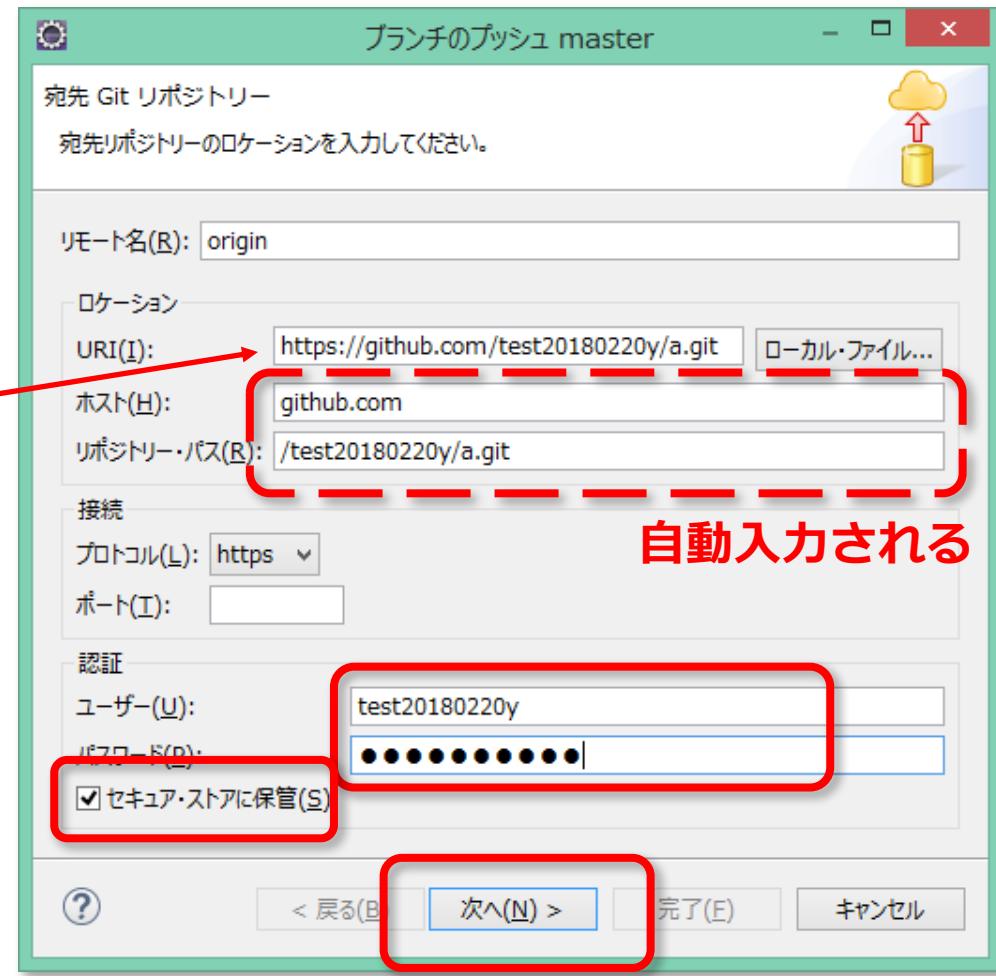
...or push an existing repository from the command line

```
git remote add origin https://github.com/test20180220y/a.git
git push -u origin master
```

...or import code from another repository

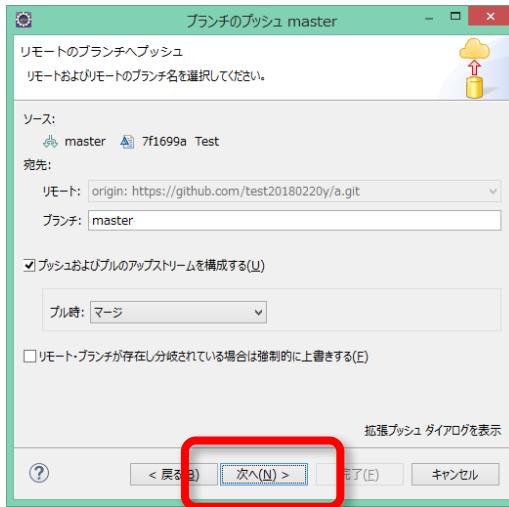
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.





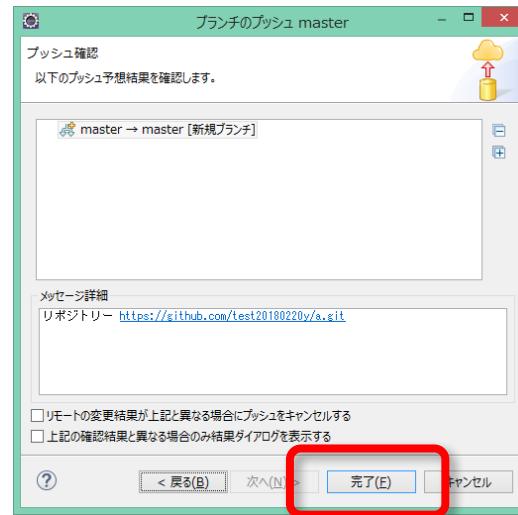
# ④Githubへ新規登録

3b. そのまま次へ



3c. 完了

*Done!*

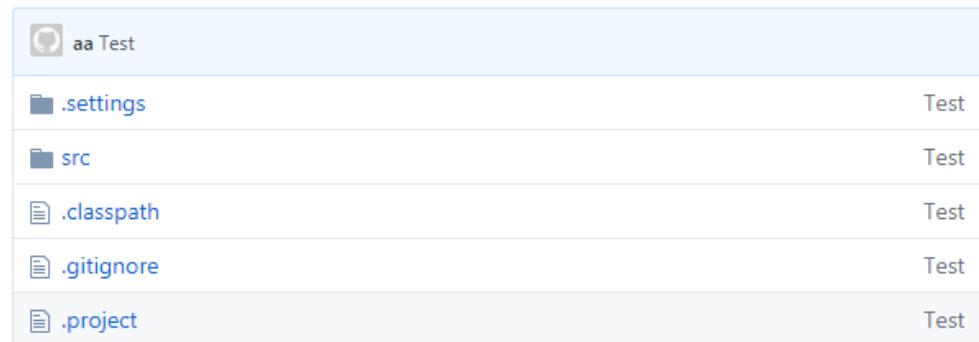


# ④Githubへ新規登録

## 同期確認

1. さっきのページのここから  
マイページに飛べる
2. リポジトリをクリックして  
反映されていれば成功 (コードはsrcフォルダにあるはず)

A screenshot of a GitHub user profile page. On the left is a large yellow 'E' icon. At the top, there's a message: "GitHub users get to know you." Below it, there are tabs for "Overview", "Repositories 1", "Stars 0", and "Followers 0". Under "Popular repositories", there is a card for a repository named "a", which is described as "Java". A red box highlights this repository card, and a large yellow arrow points from it towards the right side of the image.



# 手順



## ⑤既に登録されたプロジェクトの更新データを送信

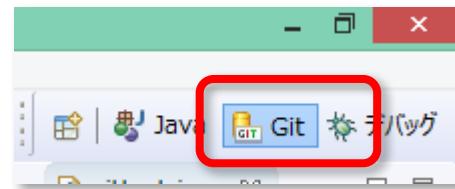
④を完了することで、以降、プログラム更新時にプロジェクトにこのような >マーク が付くようになる。



⑤では、この更新したデータをGithubへ送る方法を紹介する。

## ⑤既に登録されたプロジェクトの更新データを送信

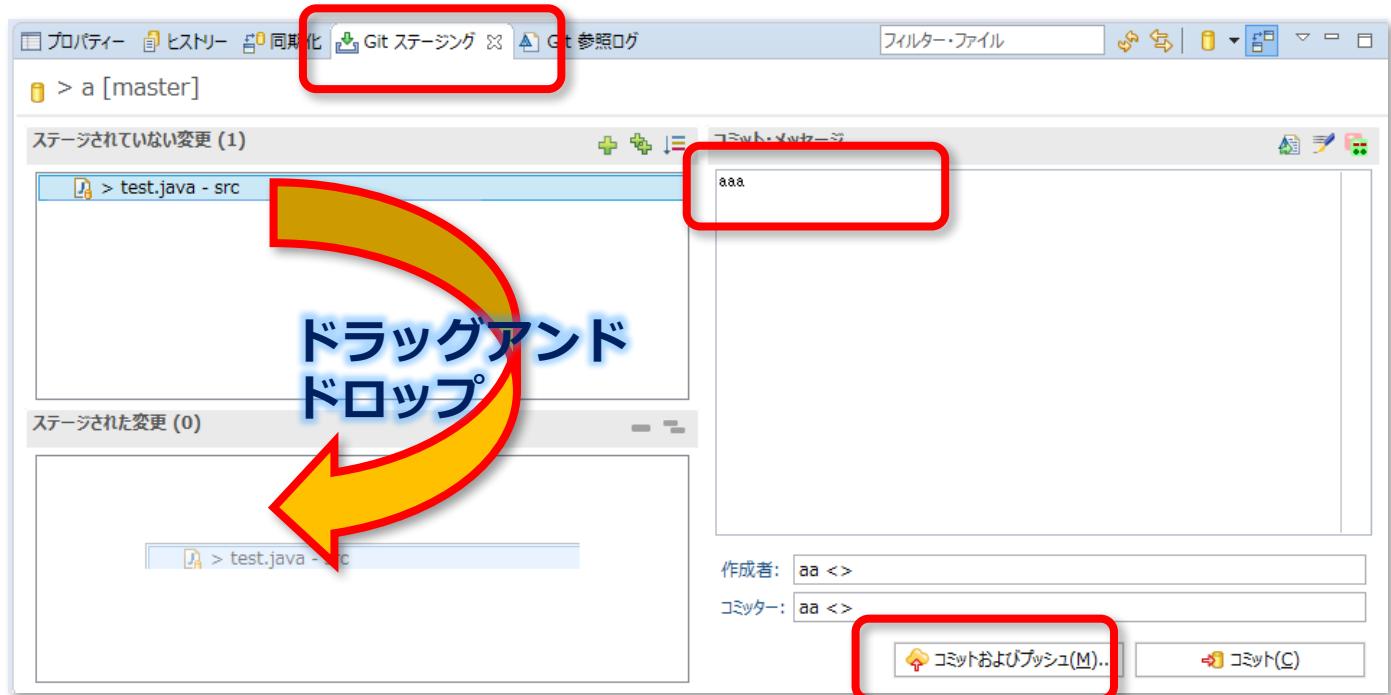
1. Gitパースペクティブを開く



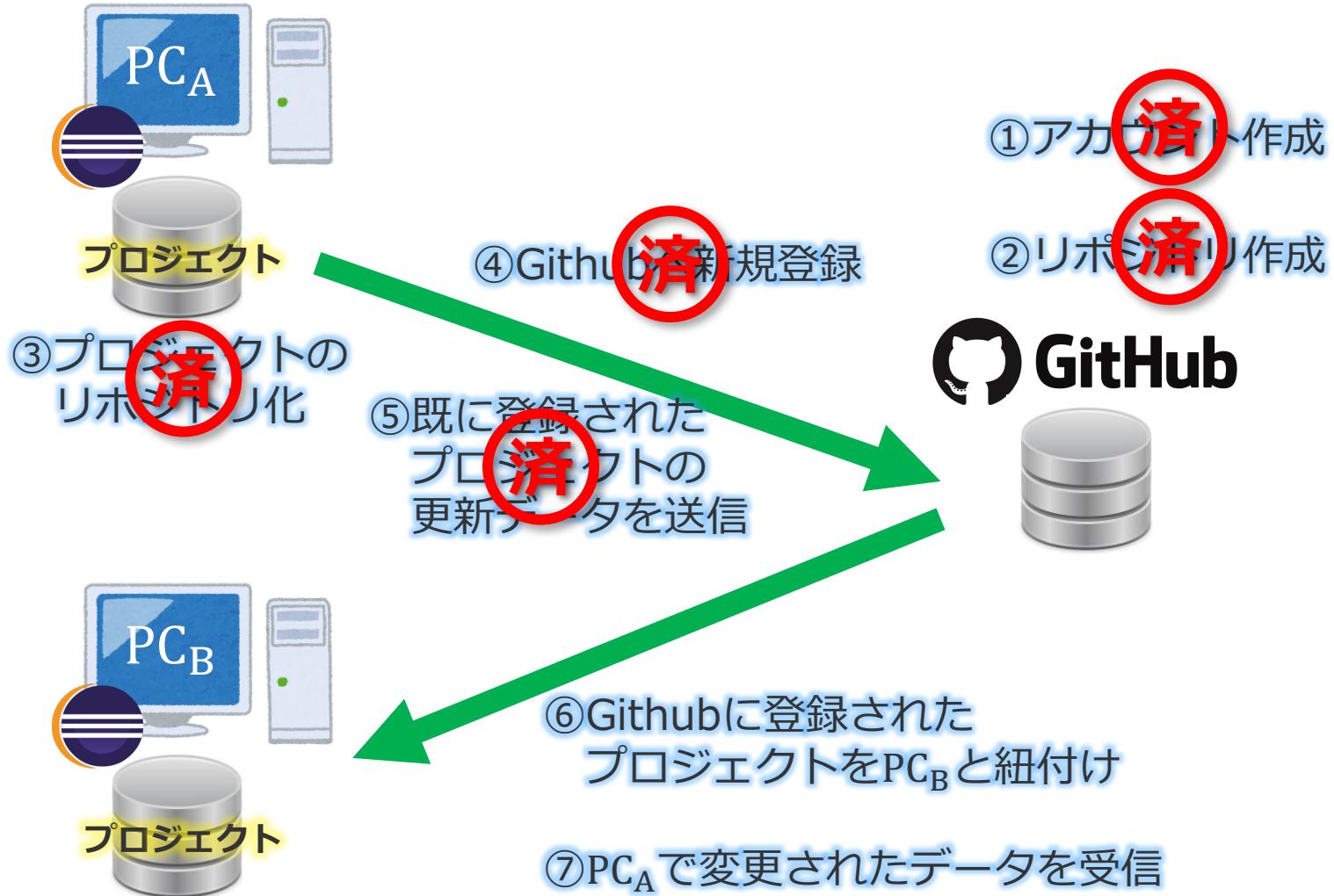
2. ④のステップ2a～2dと同様に行う

3. Githubに更新が反映される

Done!



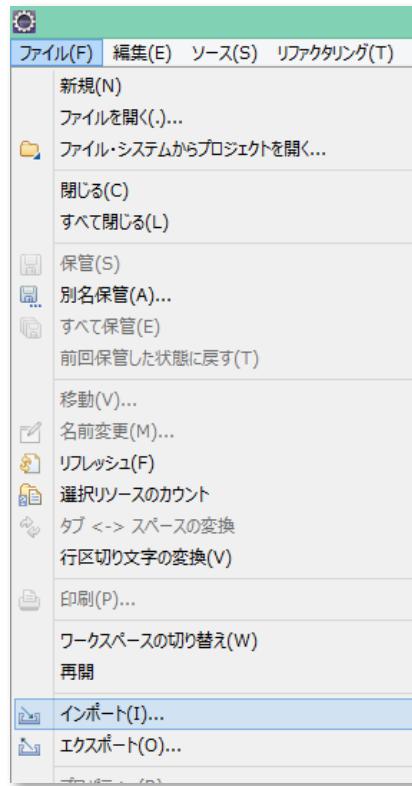
# 手順



## ⑥Githubに登録されたプロジェクトをPC<sub>B</sub>と紐付け

⑥ではGithubにあるプロジェクトを別のPCにインポートする手段を説明する。

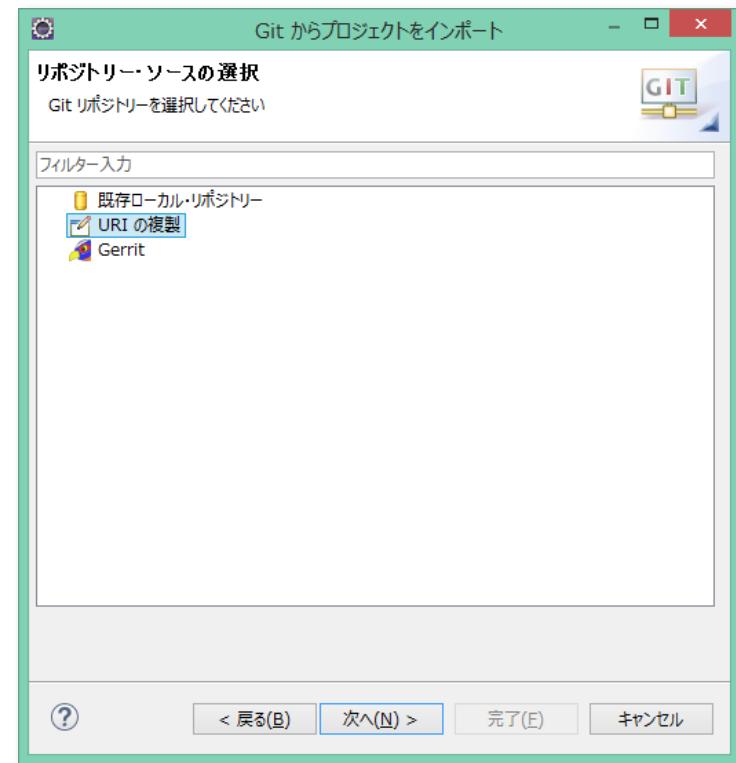
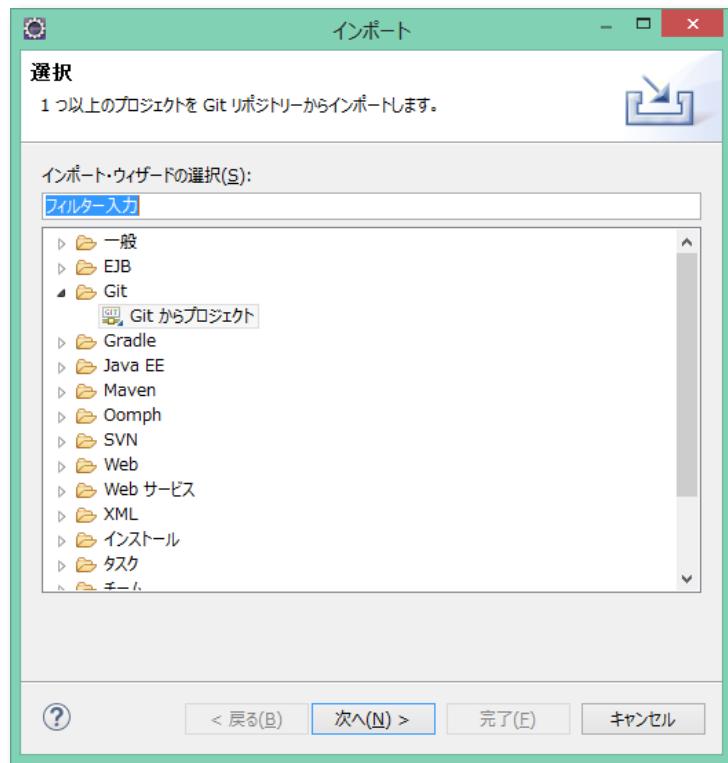
### 1. ファイル->インポートと選択



## ⑥Githubに登録されたプロジェクトをPC<sub>B</sub>と紐付け

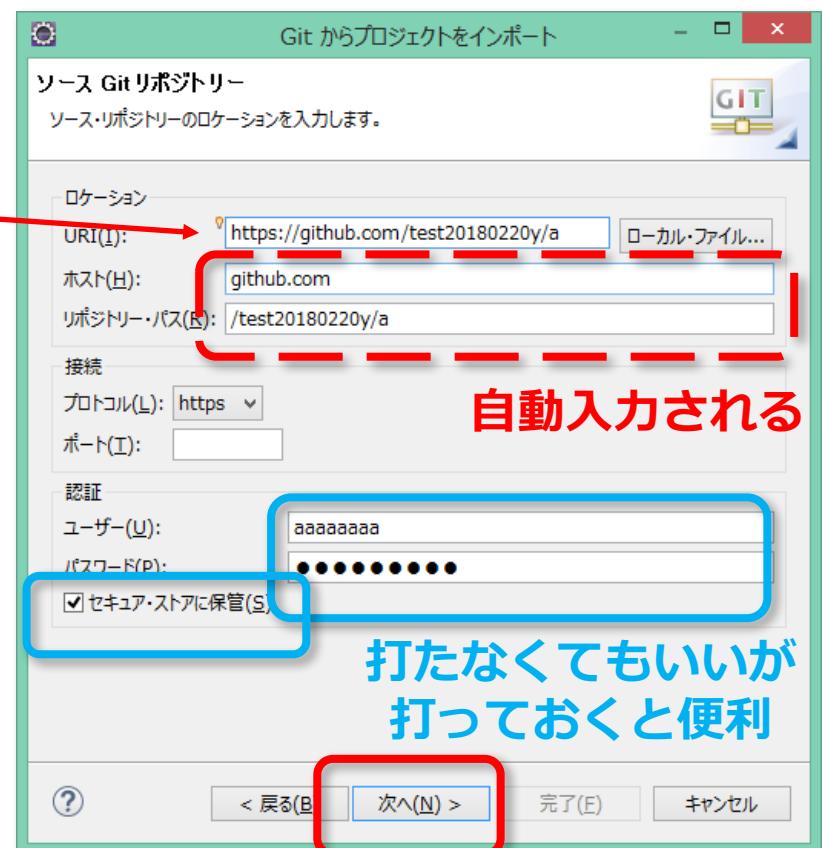
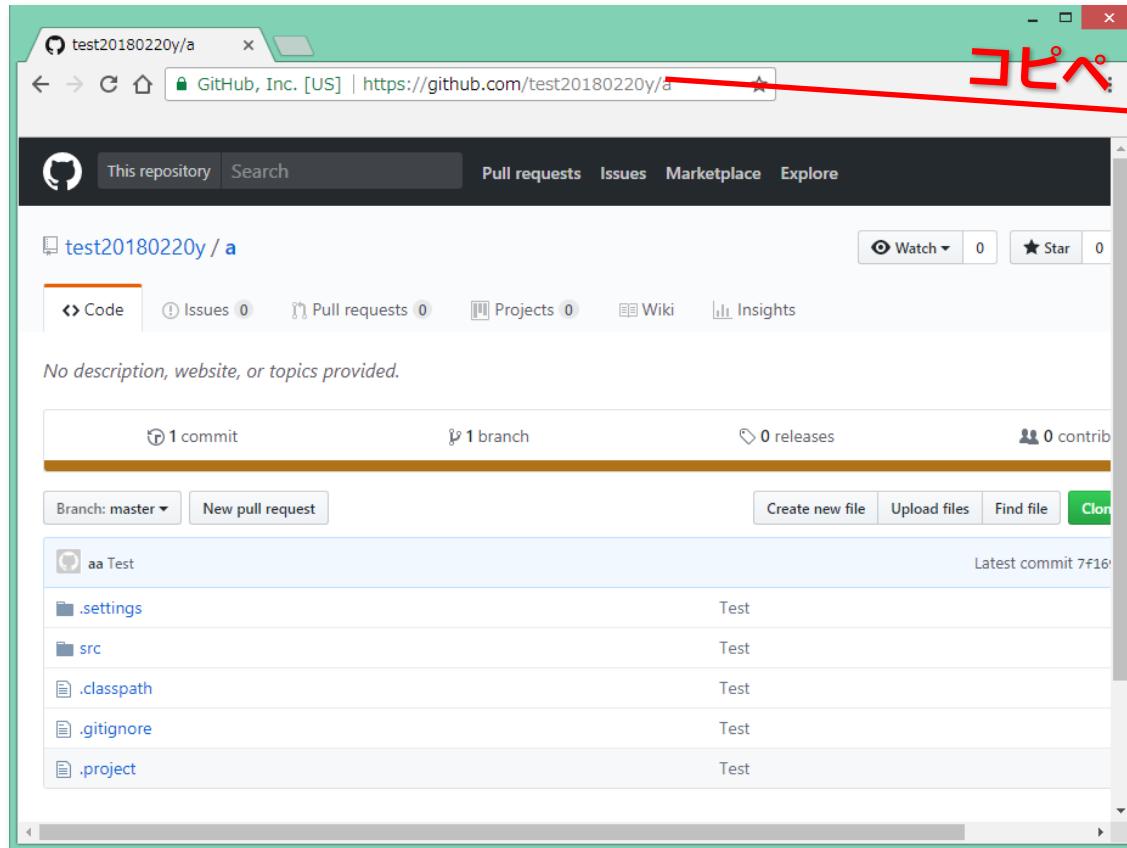
2a. Git->Gitからプロジェクトを選んで次へ

2b. URIの複製を選んで次へ



## ⑥Githubに登録されたプロジェクトをPC<sub>B</sub>と紐付け

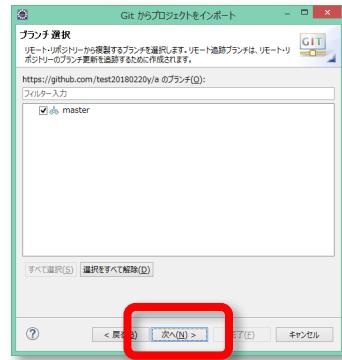
### 2c. URIにマイページのプロジェクトのアドレスを打ち込む



打たなくてもいいが  
打っておくと便利

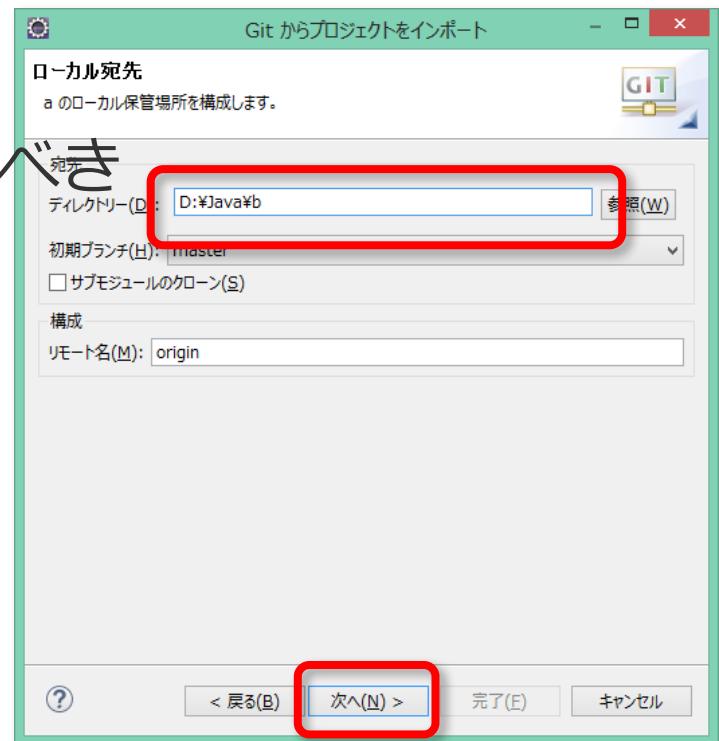
## ⑥Githubに登録されたプロジェクトをPC<sub>B</sub>と紐付け

2d. そのまま次へ



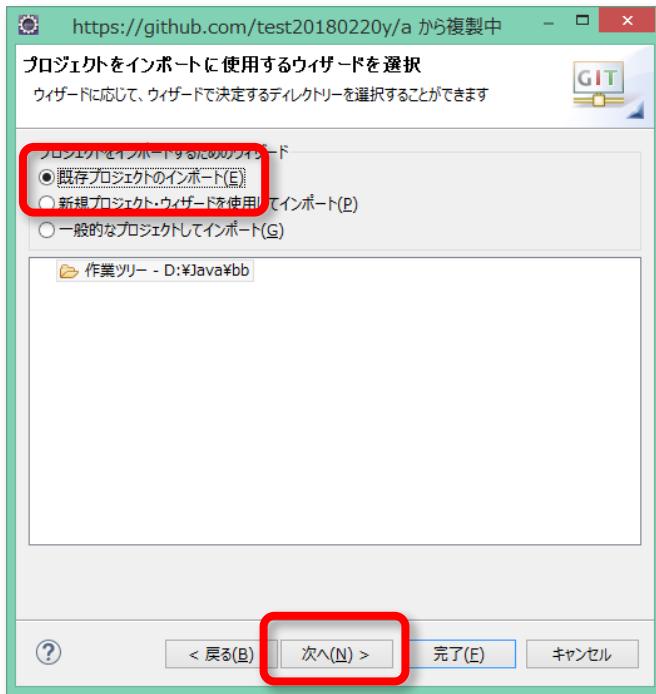
2e. ディレクトリーはEclipseの  
プロジェクトの通常の保存場所にすべき  
(ワークスペース名)¥(プロジェクト名)  
とすれば良い

例えば、  
ワークスペースが、C:¥Java で  
読み込みたいプロジェクト名が、 MyProjectなら  
C:¥Java¥MyProject となる。

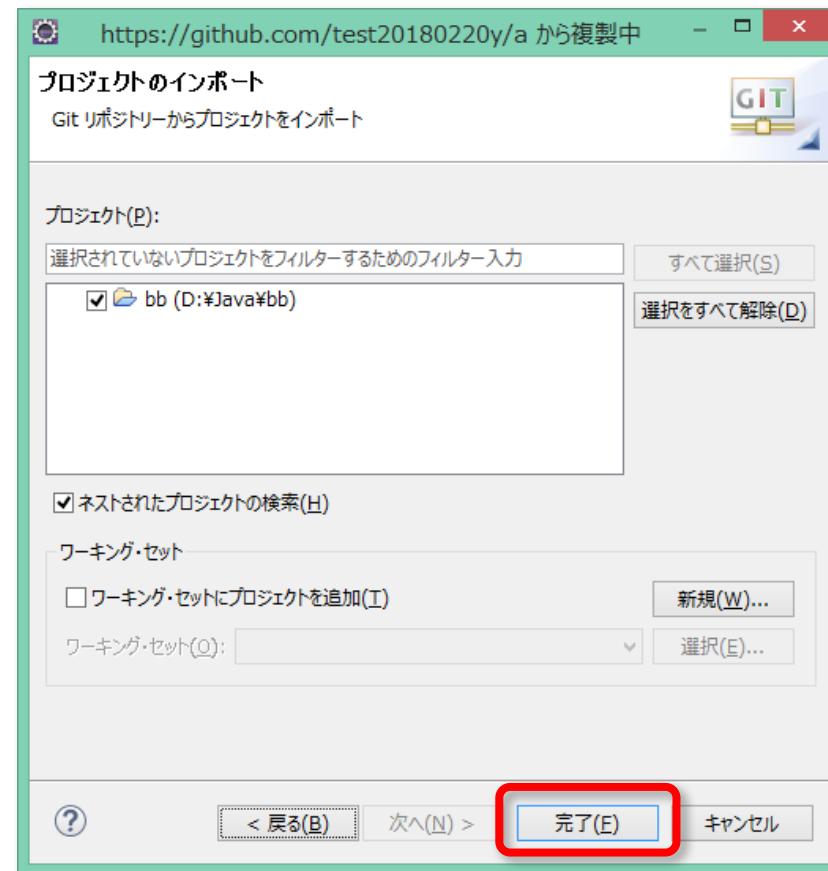


## ⑥Githubに登録されたプロジェクトをPC<sub>B</sub>と紐付け

### 2f. 既存プロジェクトのインポート、次へ

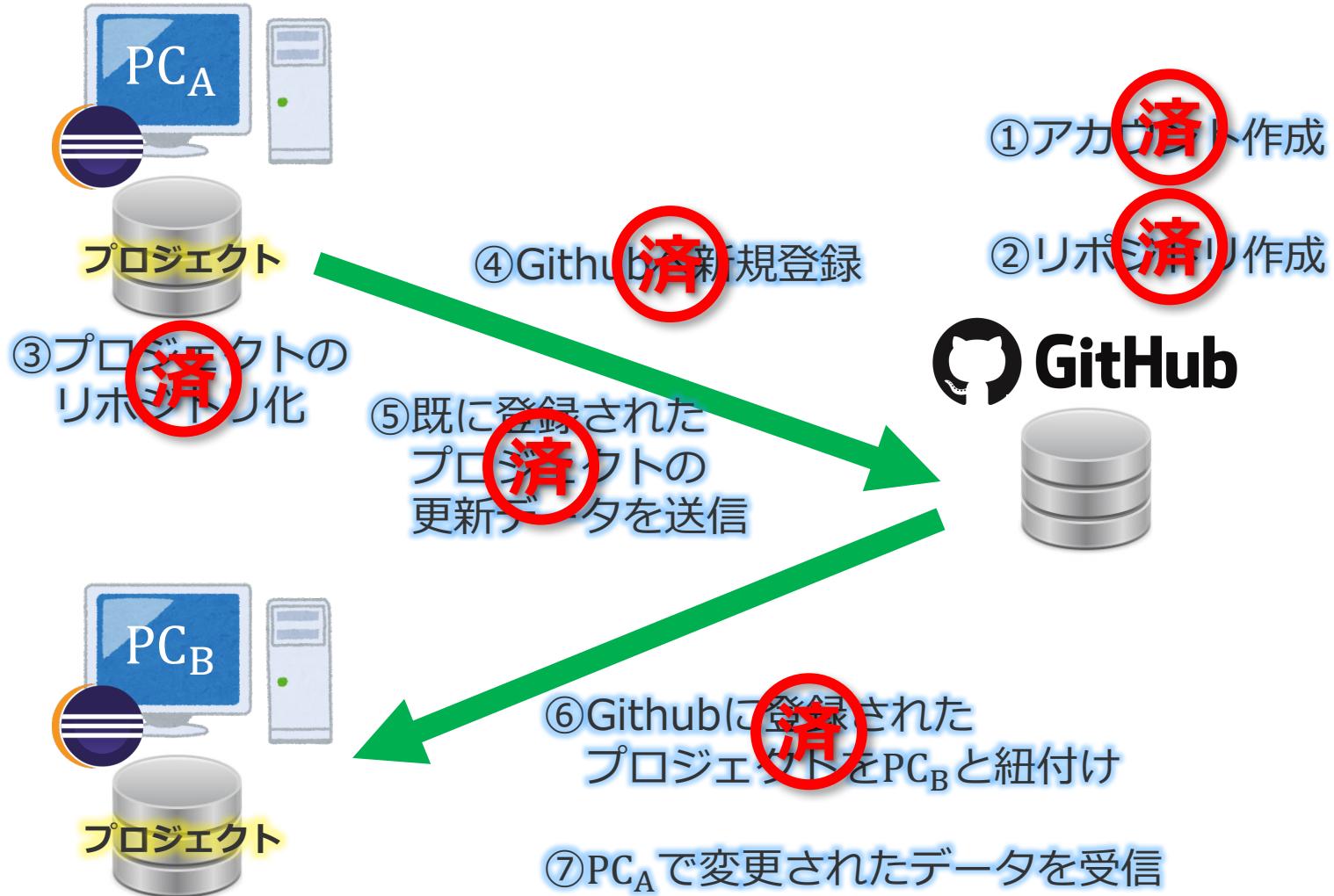


2g. 完了



Done!

# 手順



## ⑥Githubに登録されたプロジェクトをPC<sub>B</sub>と紐付け

⑦では他のPCで更新しGithubに保管されているデータをまた別のPCへ更新を反映させる方法を説明する。

1. プロジェクトを右クリック

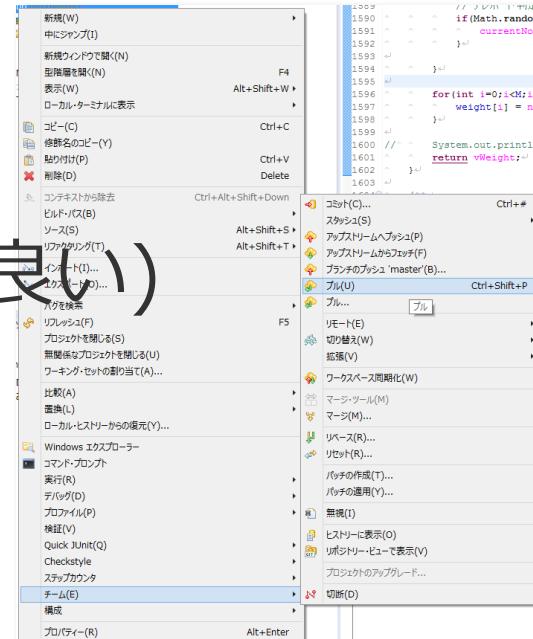
右クリック



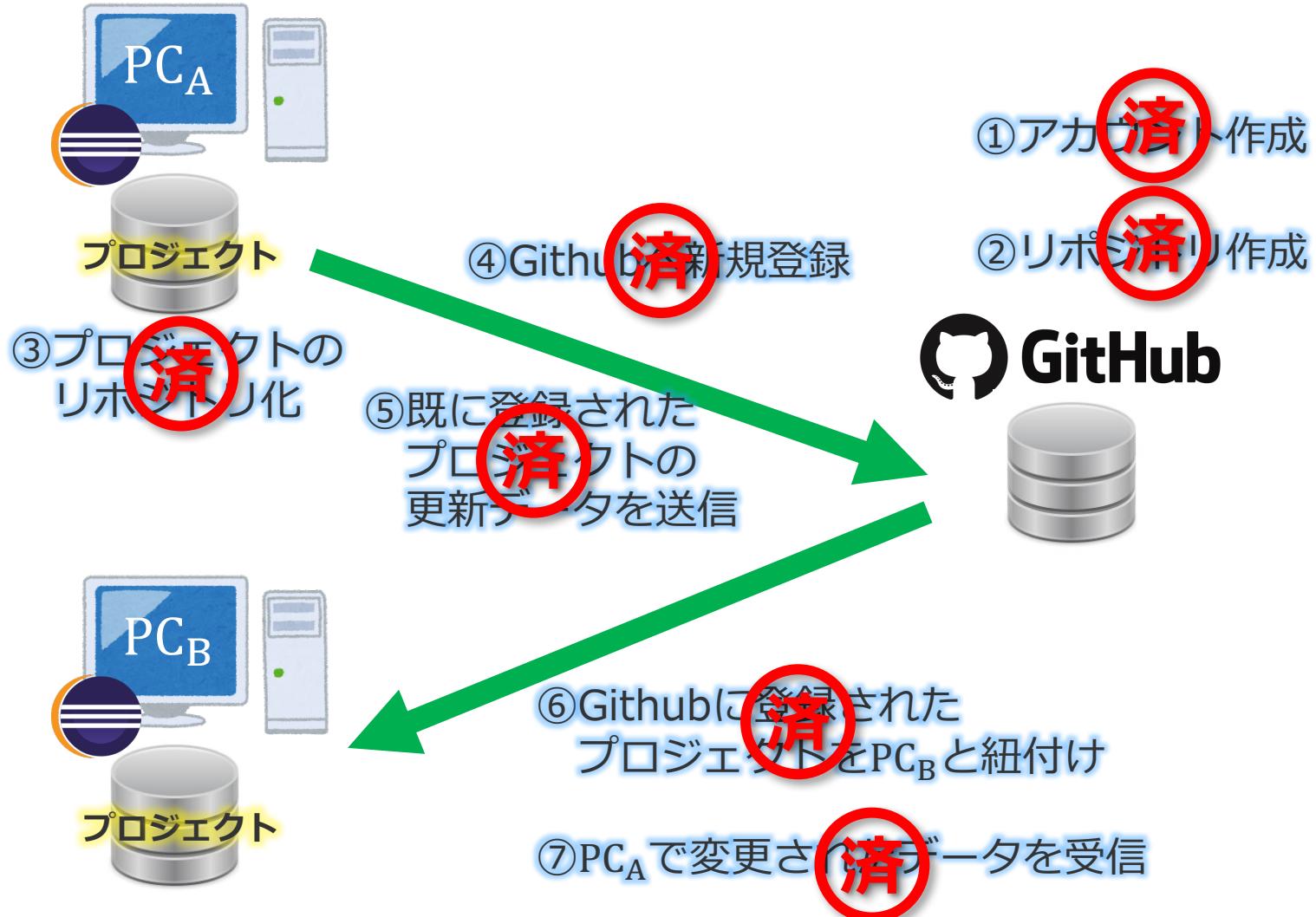
2. チーム->プル を選択

(2つあるが上のやつの方が良い)

Done!



# 手順



# 使い方

---

- ここまで終えることでデータの同期を行なうことができる。
- 更新を送信したい場合は⑤、  
更新を受信したい場合は⑦ を行えば良い。

# 未記載部分と今後の課題

## 未記載部分

- ・前バージョンへの巻き戻し方法  
関連ワード:HEAD改訂
- ・競合が起きたときの解決策  
関連ワード:マージ/merge
- ・他のプログラム言語のコードの連携方法  
関連ワード:Github Desktop, gitコマンド
- ・Githubを利用してwebサイトを公開する方法  
関連ワード:Github Pages

今後の課題@わかる人いたら教えてください

- ・AtomとGithubの連携方法
- ・pull, push, commit, mergeなどの用語の正確な意味が正確にはわかっていない。

← 暇な時に書きます