

人と技術で次代を拓く

**MEITEC**

***Engineering Firm at The Core***

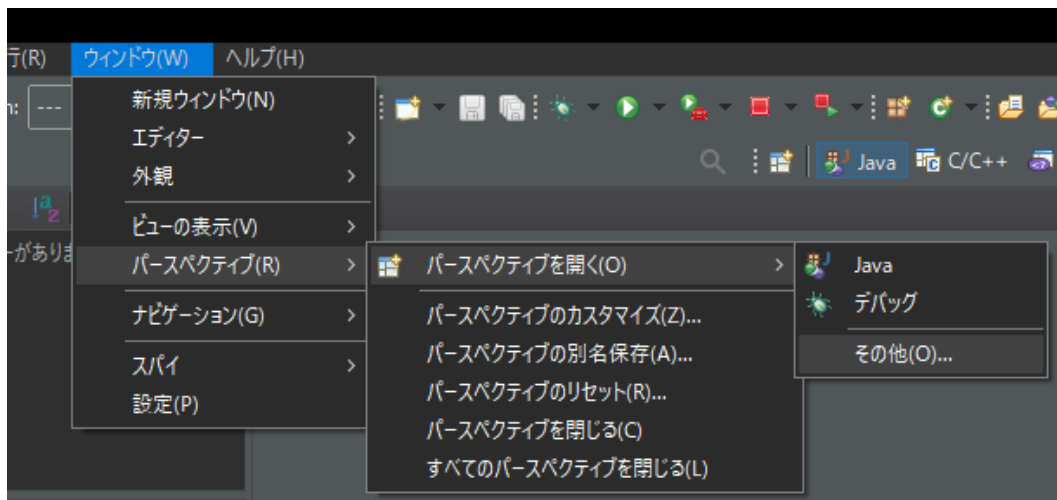
# Eclipse-GitHub連携

# 目次

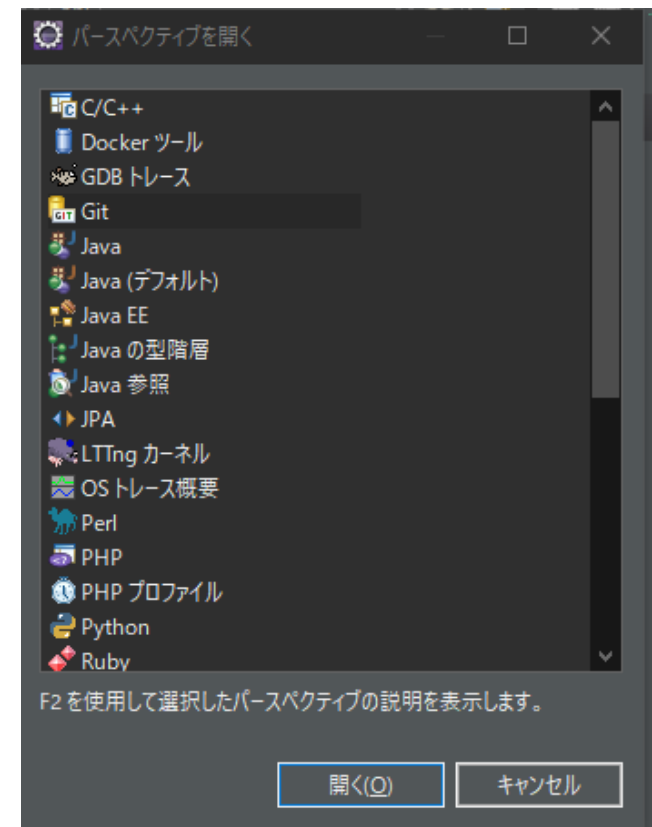
1. <u>「Git」のパースペクティブを開く</u>	5~6
2. <u>ローカルリポジトリの作成</u>	7~8
3. <u>プロジェクトの作成</u>	9~11
4. <u>新規ファイル作成</u>	12~13
5. <u>ローカルリポジトリへのコミット</u>	14~15
6. <u>GitHubのアクセストークンを作成</u>	16~19
7. <u>リモートリポジトリへプッシュ</u>	20~25
8. <u>リモートリポジトリのクローン</u>	26~33

## 「Git」のパーспекティブを開く

- Eclipseの「ウィンドウ」→「パーспекティブ」→「パーспекティブを開く」→「その他」を選択する。



- Gitを選択し、「開く」をクリック。

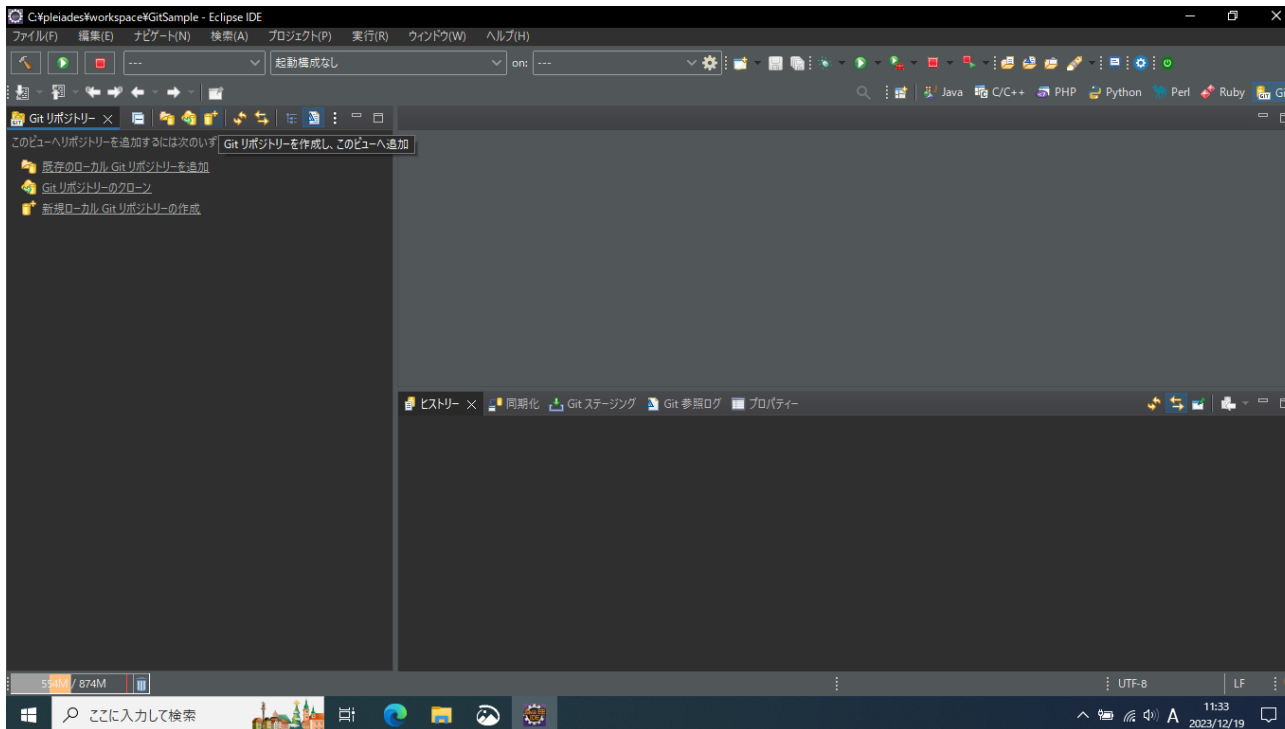


## 「Git」のパーспекティブを開く

- Gitのパーспекティブが作成される。

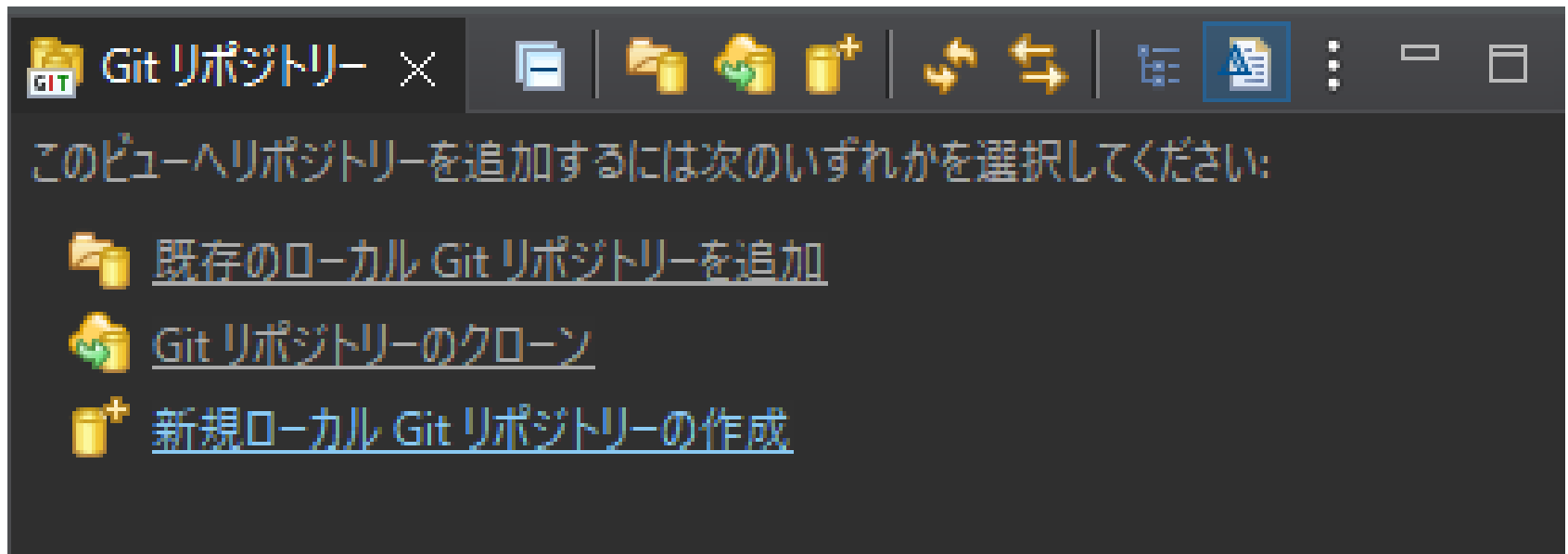


- Gitのパーспекティブをクリックする。



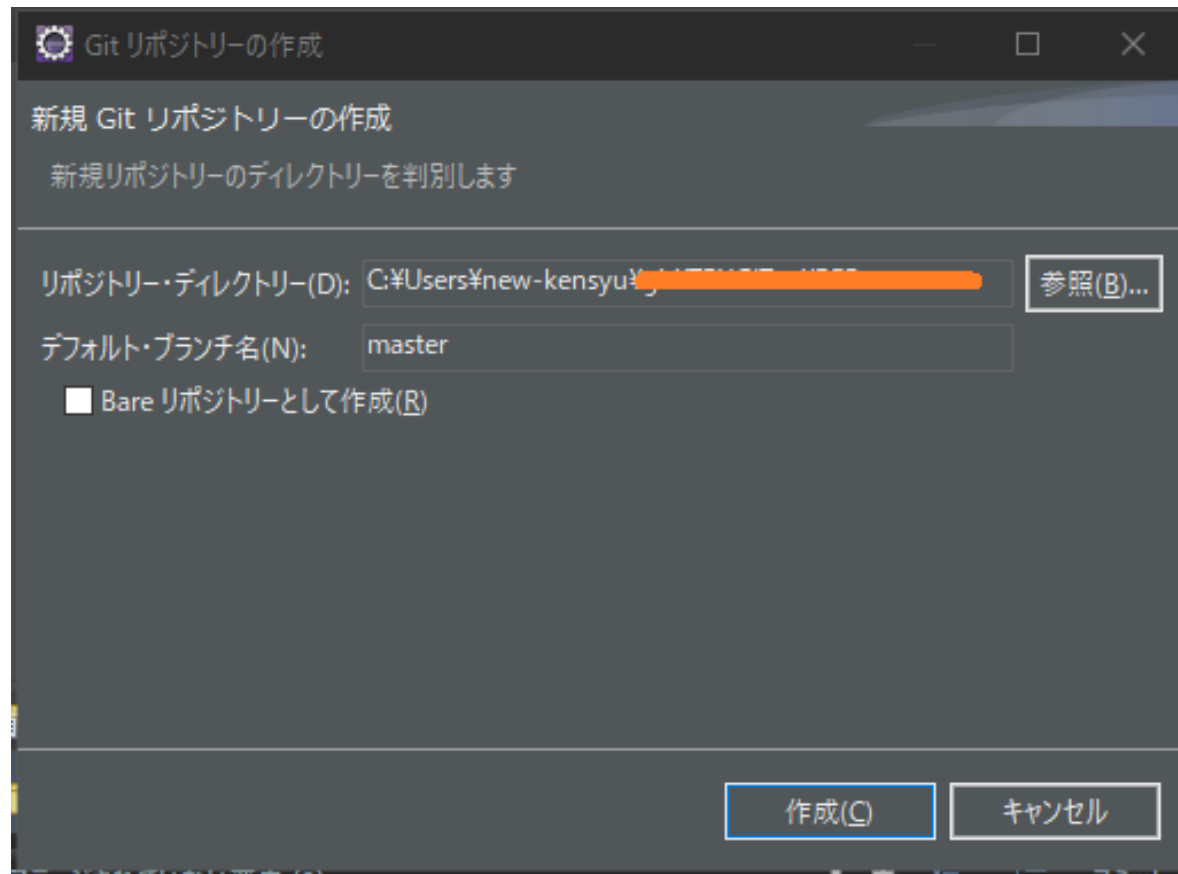
## ローカルリポジトリの作成

- 「新規ローカルGitリポジトリの作成」をクリックする。



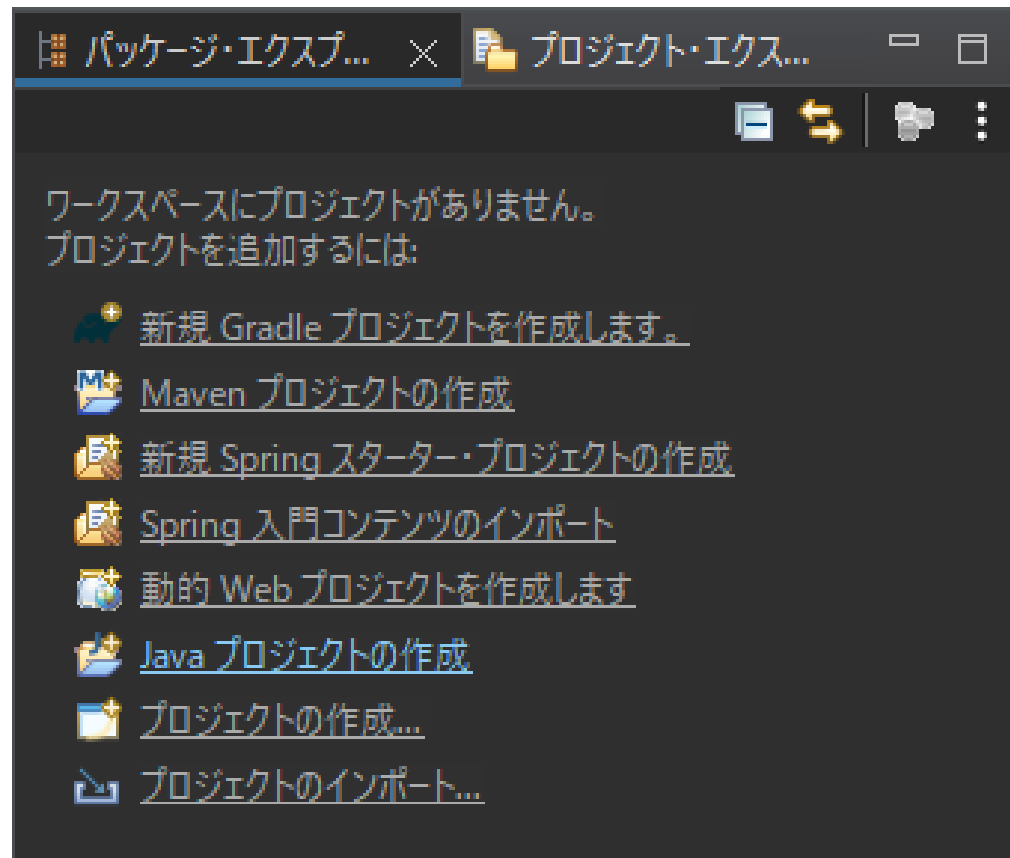
## ローカルリポジトリの作成

- 「参照」をクリックし、ローカルリポジトリのディレクトリを設定し、「作成」をクリックする。



## プロジェクトの作成

- 「Java」のパースペクティブを開き「Javaプロジェクトの作成」選択します。





## プロジェクトの作成

- 「プロジェクト名」を入力し「デフォルト・ロケーションを使用」のチェックを外し、「次へ」をクリックする。

新規 Java プロジェクト

Java プロジェクトの作成

Java プロジェクトをワークスペースまたは外部ロケーションに作成します。

プロジェクト名(P):

☐ デフォルト・ロケーションを使用(D)

ロケーション(L): C:\Users\new-kensyu\... 参照(R)...

JRE

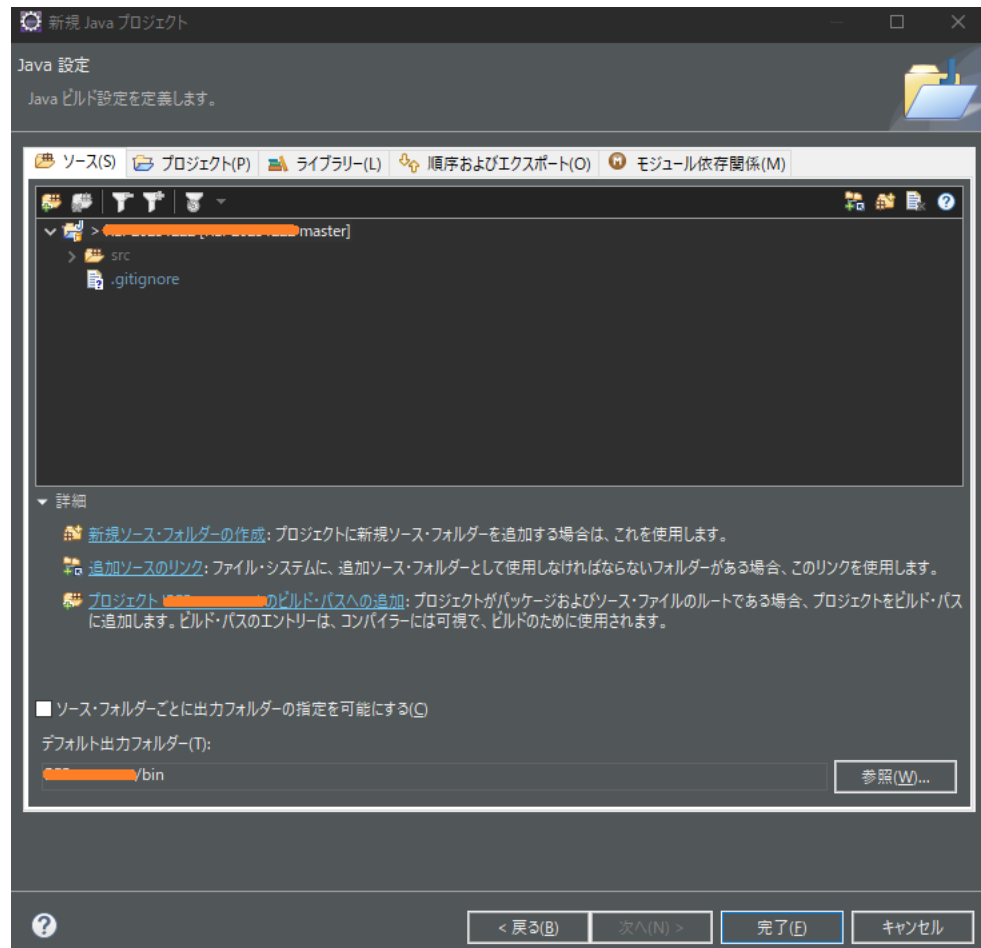
☒ 実行環境 JRE の使用(V): JavaSE-17

☒ プロジェクト固有の JRE を使用(S): Java17

☐ デフォルトの JRE 'Java17' およびワークスペース・コンパイラー設定を使用する(A) JRE を構成...

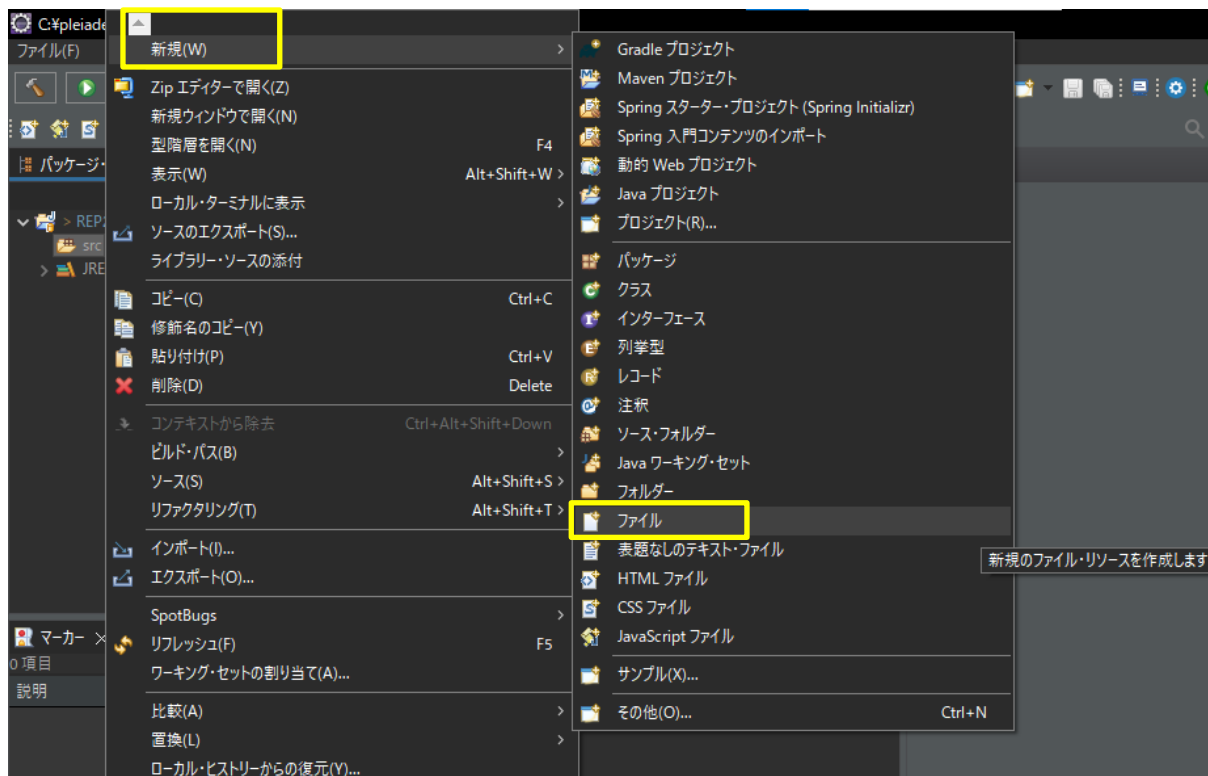
# プロジェクトの作成

- 「完了」をクリックする。



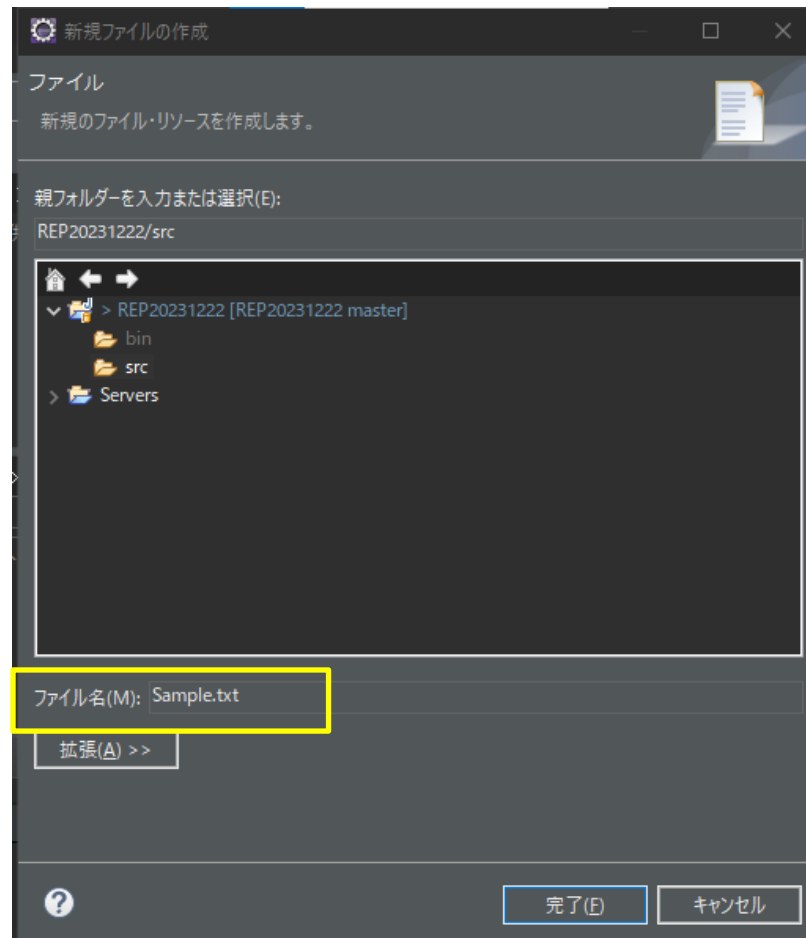
## 新規ファイル作成

- 「Java」のパーспекティブで「新規作成」→「ファイル」を選択する。




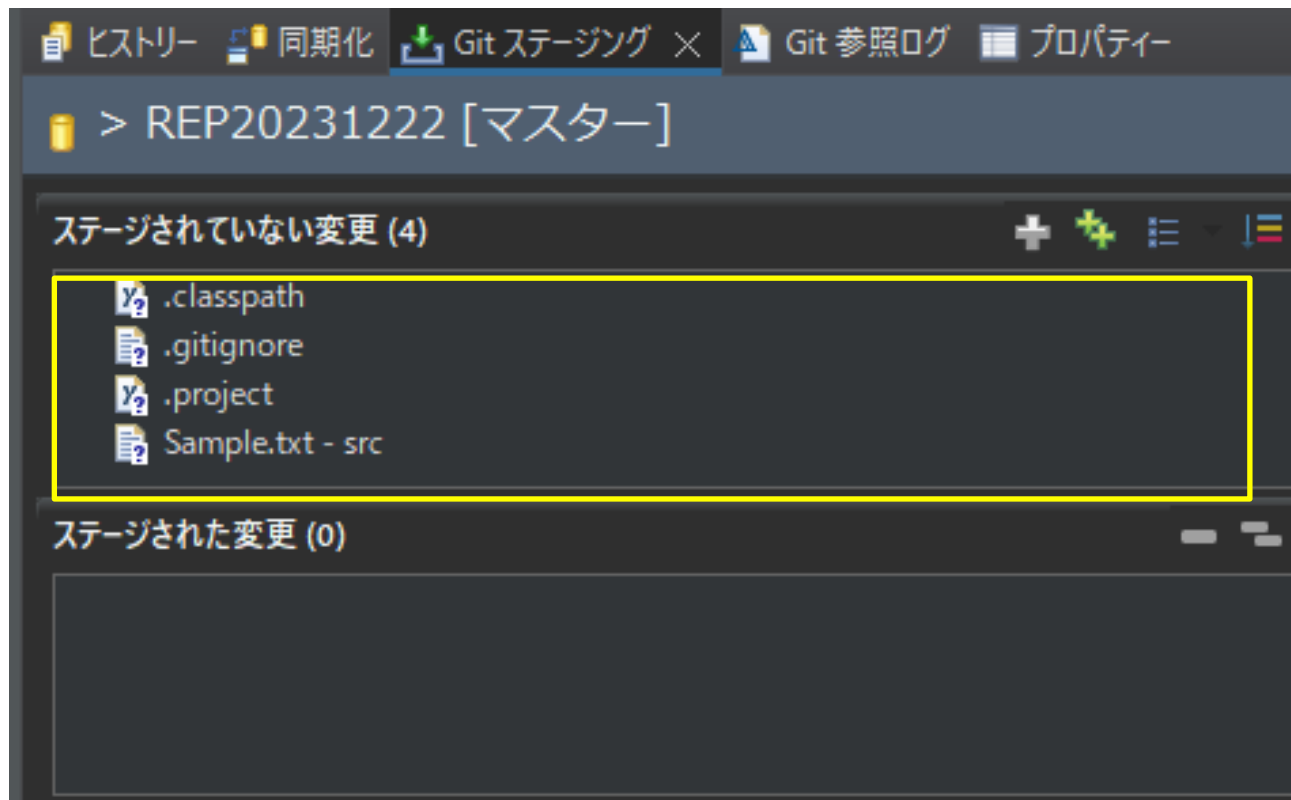
## 新規ファイル作成

- 「ファイル名」を入力し、新規ファイルを作成する。



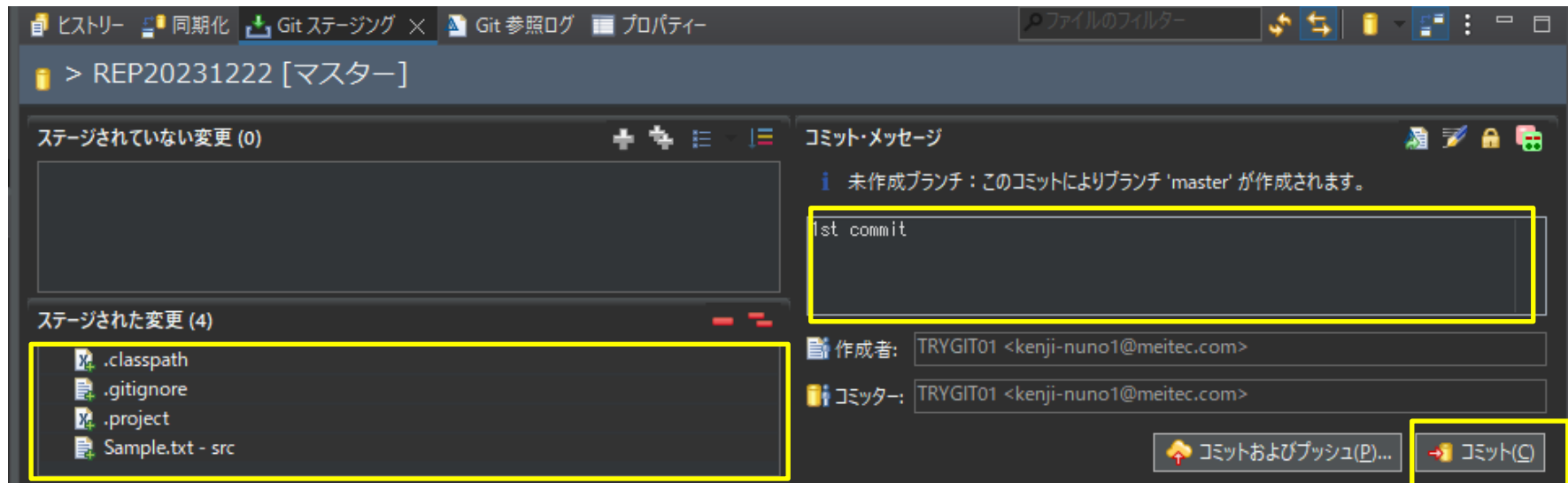
## ローカルリポジトリへのコミット

- 作成したファイルを編集・保存し、「Git」パースペクティブを開き、addしたいファイルを選択し、 をクリックする。



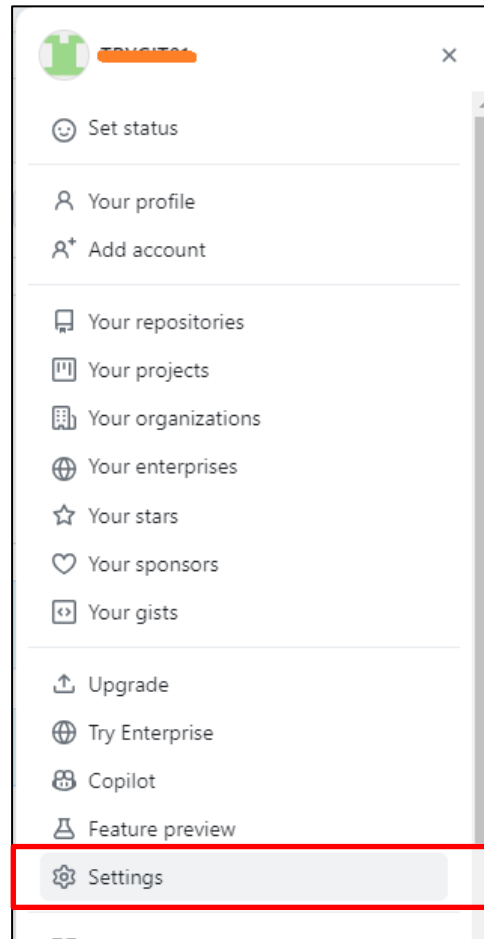
## ローカルリポジトリへのコミット

- 「コミットメッセージ」を入力し、「コミット」をクリックして、「ステージされた変更」コミットする。



## GitHubのアクセストークンを作成

- GitHubの設定から「Settings」を選択する。



## GitHubのアクセストークンを作成

- 「<> Developer settings」を選択する。

The screenshot shows the GitHub profile settings page. On the left sidebar, the 'Developer settings' option, represented by a code icon (<>), is highlighted with a red rectangular box. The main content area on the right includes sections for 'Link to social profile' (three links), 'Company' (a text input field), 'Location' (a text input field), and a checkbox for 'Display current local time'. At the bottom right is a green 'Update profile' button. A disclaimer at the bottom states: 'All of the fields on this page are optional. By clicking Update, you consent to share this data wherever it is needed to use GitHub, and about how we use this information.'



## GitHubのアクセストークンを作成

- 「Personal access tokens」→「Tokens (classic)」を選択し、「Note」を入力、「repo」にチェックを入れる。

The screenshot shows the GitHub interface for creating a new personal access token (classic). The left sidebar contains the following items:

- GitHub Apps
- OAuth Apps
- Personal access tokens** (highlighted with a red box)
- Fine-grained tokens (Beta)
- Tokens (classic)** (highlighted with a red box)

The main content area is titled "New personal access token (classic)". It includes the following fields and options:

- Note:** A text input field containing "CustomerRepo" (highlighted with a red box).
- Expiration:** A dropdown menu set to "30 days". A note indicates "The token will expire on Thu, Jan 18 2024".
- Select scopes:** A section where scopes are defined. The "repo" scope is checked (highlighted with a red box), and "repo:status" is also checked. The descriptions for these scopes are "Full control of private repositories" and "Access commit status" respectively.

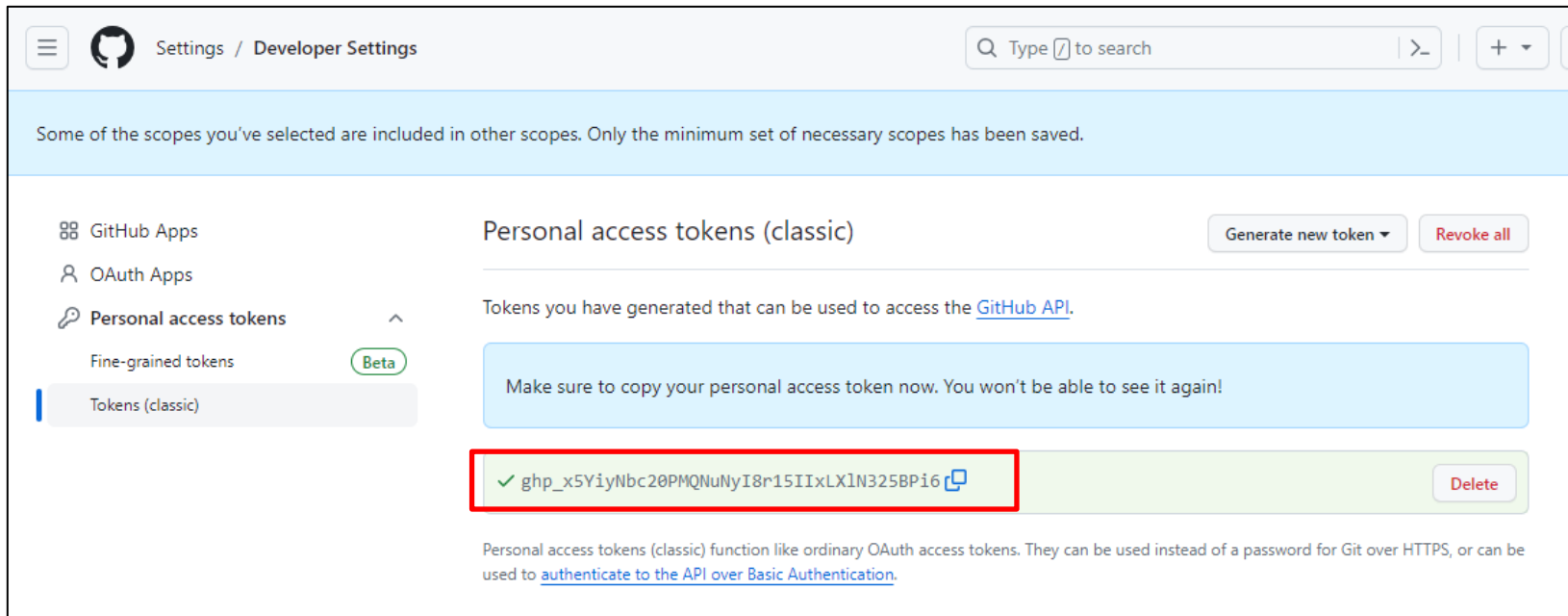
## GitHubのアクセストークンを取得

- 「Generate token」をクリックする。

Generate token

Cancel

- トークンが作成され、でコピーして使用する。



The screenshot shows the GitHub Developer Settings page. The left sidebar contains a menu with 'GitHub Apps', 'OAuth Apps', 'Personal access tokens' (selected), 'Fine-grained tokens' (marked Beta), and 'Tokens (classic)'. The main content area is titled 'Personal access tokens (classic)' and includes a 'Generate new token' button and a 'Revoke all' button. A message states: 'Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.' Below this, a warning box says: 'Make sure to copy your personal access token now. You won't be able to see it again!'. A table lists one token: 'ghp\_x5YiyNbc20PMQNuNyI8r15IIxLXlN325BPi6', which is highlighted with a red box and a copy icon. A 'Delete' button is next to it. At the bottom, a note explains that these tokens function like ordinary OAuth access tokens and can be used for authentication over HTTPS.

Settings / Developer Settings

Some of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens Beta

Tokens (classic)

Personal access tokens (classic) Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

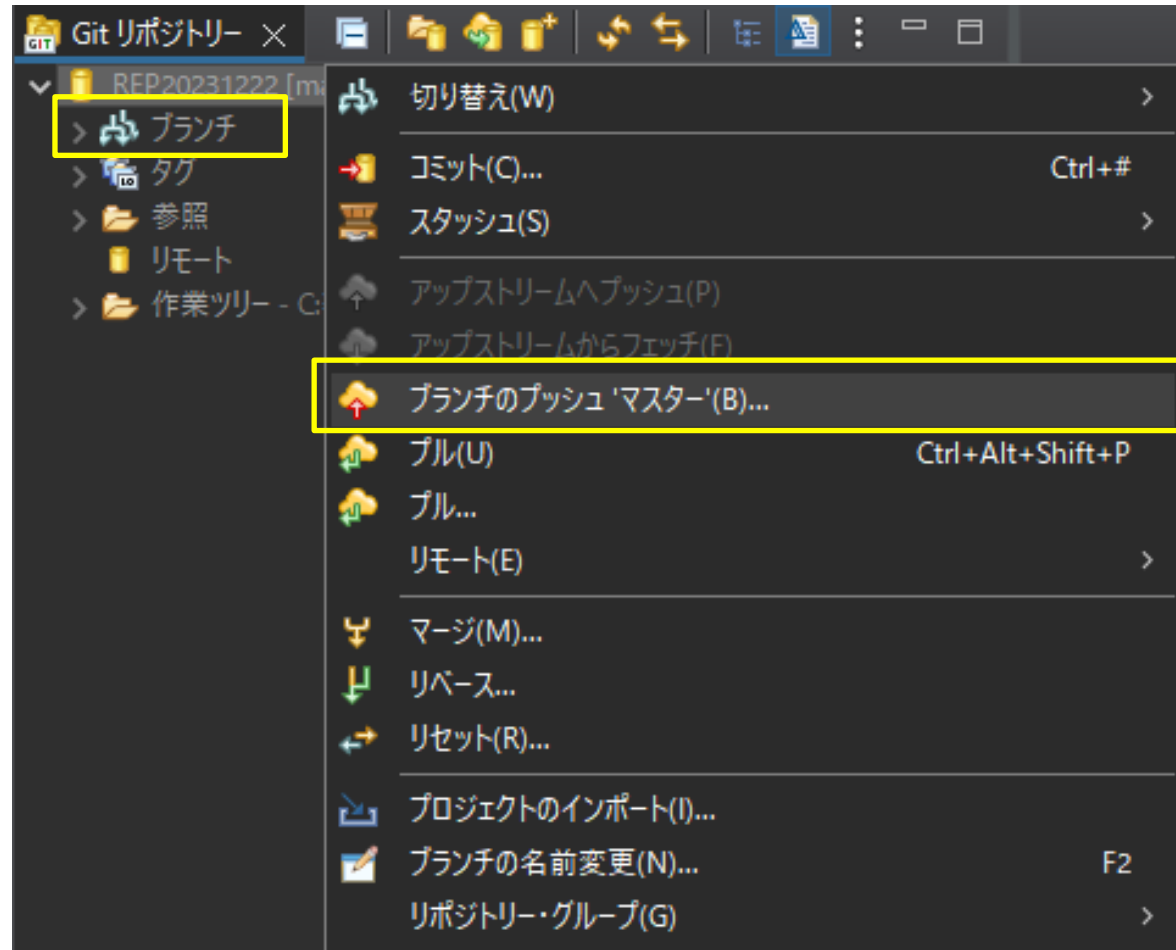
Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_x5YiyNbc20PMQNuNyI8r15IIxLXlN325BPi6 Copy Delete

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

# リモートリポジトリへプッシュ

- 「ブランチのプッシュ」を選択する。



パスワードは、GitHubのアクセストークン

# リモートリポジトリへプッシュ

- 「プレビュー」をクリックする。

ブランチのプッシュ master

リモートのブランチへプッシュ

リモートおよびリモートのブランチ名を選択してください。

ソース:

マスタ 128d929 1st commit

宛先:

リモート: origin: https://github.com/TRYGIT01/REP20231222.git

ブランチ: master

☒ プッシュおよびプルの上ストリームを構成する(U)

プル時: マージ

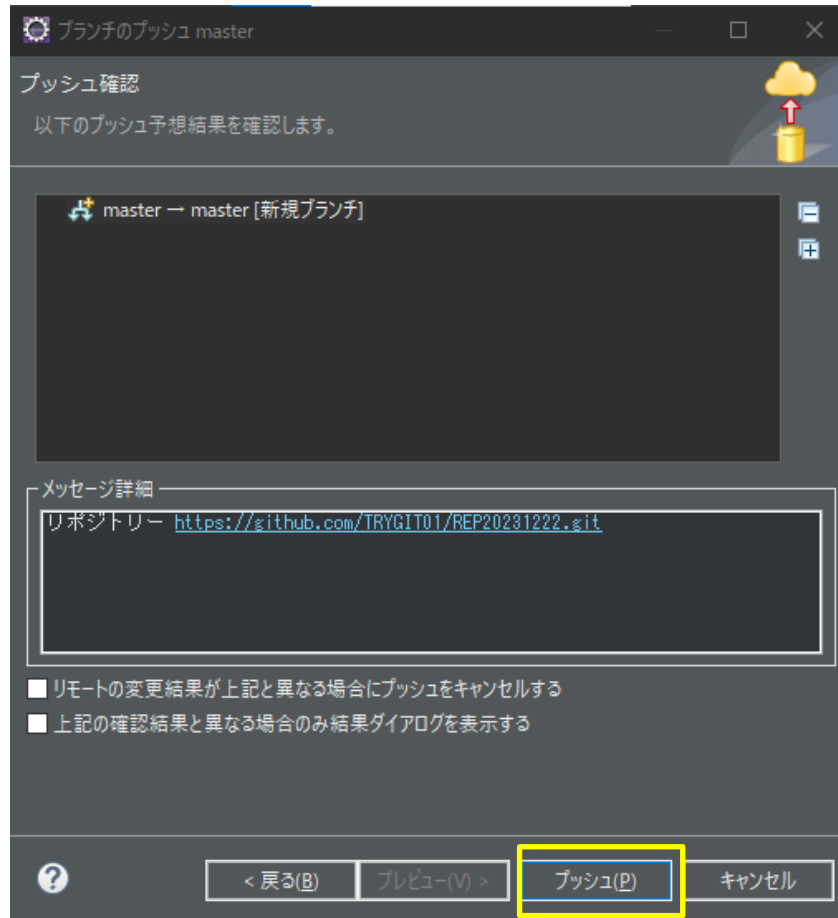
☐ リモート・ブランチが存在し分岐されている場合は強制的に上書きする(F)

拡張プッシュ</A> ダイアログを表示

< 戻る(B) **プレビュー(V) >** プッシュ(P) キャンセル

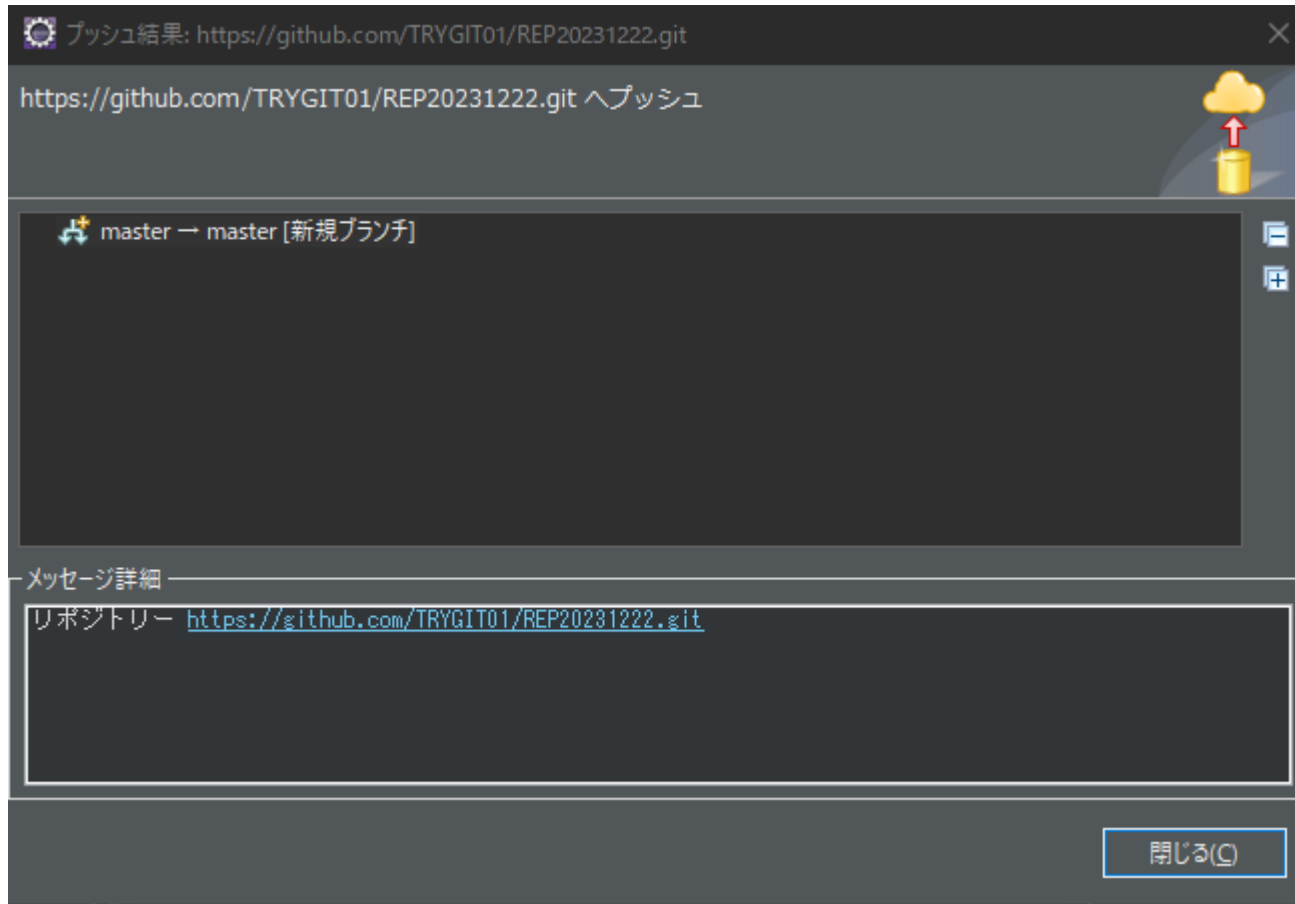
## リモートリポジトリへプッシュ

- 「プッシュ確認」画面で「プッシュ」をクリックする。



## リモートリポジトリへプッシュ

- リモートリポジトリへのプッシュ完了。



## リモートリポジトリの内容確認

- GitHubのリモートリポジトリにファイルがプッシュされていることを確認する。

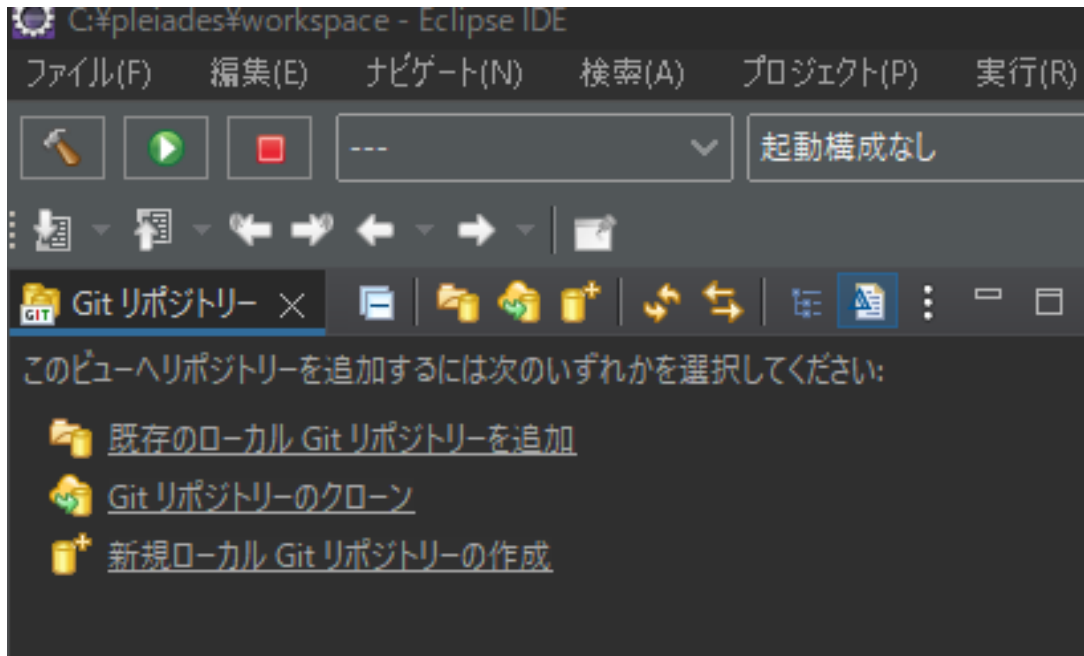
The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. Below this, the repository name is displayed as 'Private'. The main content area shows the 'master' branch with 1 Branch and 0 Tags. A search bar 'Go to file' is present. Below the search bar, a table lists the files in the '1st commit'.

File	Commit	Time
src	1st commit	8 minutes ago
.classpath	1st commit	8 minutes ago
.gitignore	1st commit	8 minutes ago
.project	1st commit	8 minutes ago



## リモートリポジトリのクローン

- 他のユーザが作成したprivateリポジトリにアクセスできるようにしておく。（リポジトリの共有.pptx参照）
- 「Git」 パースペクティブで「Gitリポジトリのクローン」を選択する。



## リモートリポジトリのクローン

- 他のユーザが作成したリモートリポジトリの情報を入力し、「次へ」クリック。

The screenshot shows a 'Clone' dialog box with the following sections and fields:

- ロケーション (Location):**
  - URI(I): [redacted].git
  - Buttons: ローカル・フォルダー..., ローカル・バンドル・ファイル...
  - ホスト(H): github.com
  - リポジトリ・パス(R): /mnt/ssh2/[redacted].git
- 接続 (Connection):**
  - プロトコル(L): https (dropdown)
  - ポート(T): [empty]
- 認証 (Authentication):**
  - ユーザー(U): [redacted]
  - パスワード(P): [masked with dots]
  - ☒ セキュア・ストアに保管(S)
- Navigation:**
  - Buttons: ? (help), < 戻る(B), 次へ(N) > (highlighted), 完了(F), キャンセル

## リモートリポジトリのクローン

- ブランチを選択し、「次へ」とクリック。

https://github.com/TRYGIT02/REP20231225.git のブランチ(O):

フィルター入力

☒ master

すべて選択(S)    選択をすべて解除(D)

タグ・フェッチ戦略

☐ コミットをフェッチするときはタグもフェッチする

☒ すべてのタグとそのコミットをフェッチする

☐ タグをフェッチしない

?    < 戻る(B)    次へ(N) >    完了(F)    キャンセル

## リモートリポジトリのクローン

- ローカルリポジトリを設定し、「完了」をクリック。

宛先

ディレクトリ(D): C:\Users\new-kensyu\git\... 参照(W)

初期ブランチ(H): master

☐ サブモジュールのクローン(S)

構成

リモート名(M): origin

プロジェクト

☒ クローン終了後、すべての既存 Eclipse プロジェクトをインポート(I)

ワーキング・セット

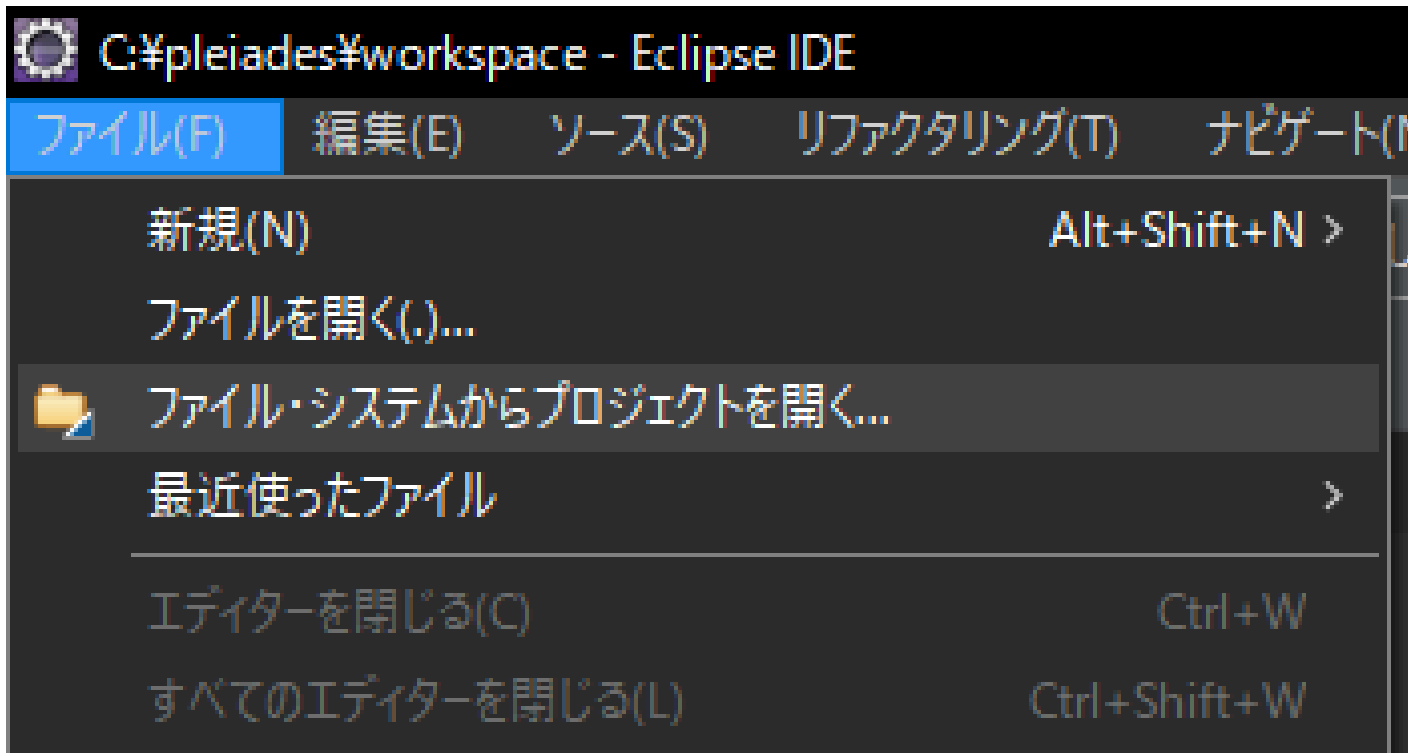
☐ ワーキング・セットにプロジェクトを追加(I) 新規(W)...

ワーキング・セット(O): 選択(E)...

? < 戻る(B) 次へ(N) > 完了(F) キャンセル

## リモートリポジトリのクローン

- 「Java」のパーспекティブを開き、「ファイル・システムからプロジェクトを開く」を選択する。



## リモートリポジトリのクローン

- リモートリポジトリをクローンしたローカルリポジトリをプロジェクト先に指定する。

ファイル・システムまたはアーカイブからプロジェクトをインポート

ファイル・システムまたはアーカイブからプロジェクトをインポート  
このウィザードはプロジェクトを検索するフォルダーまたはアーカイブ・ファイルの内容を解析し、IDE へインポートします。

インポート・ソース(I): C:\Users\new-kensyu\git\

フィルター入力

フォルダー	インポート
<input checked="" type="checkbox"/>	

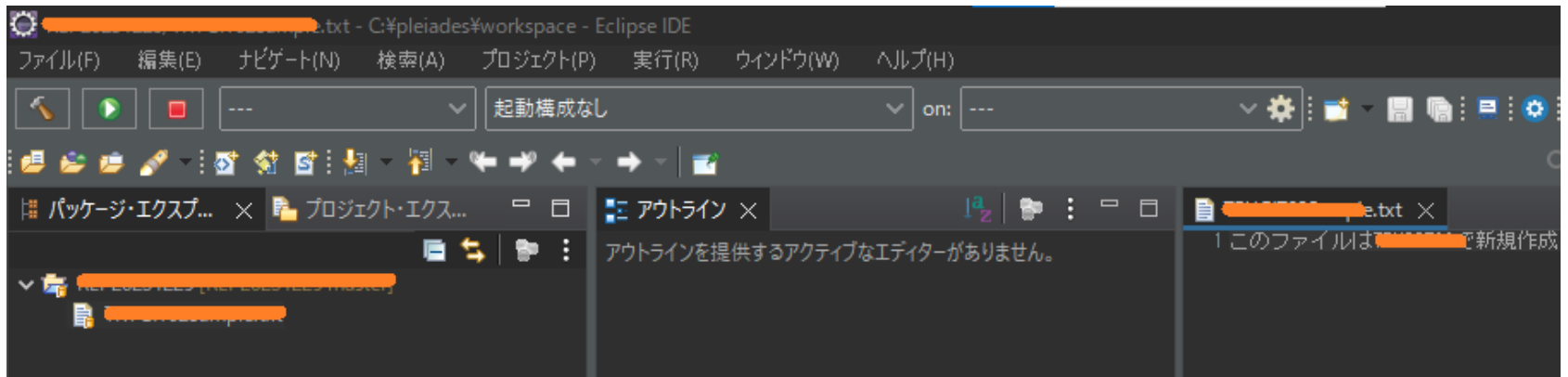
1 から 1 を選択  
☐ すでに開いて

☐ 完了時に新しくインポートされたプロジェクトを閉じる(O)  
[インストールされたプロジェクトのコンフィグレーターを使用:](#)  
☒ ネストされたプロジェクトの検索(P)  
☒ プロジェクト・ネーチャーの検出および構成(C)

ワーキング・セット

## リモートリポジトリのクローン

- 「Java」パーспекティブで、クローンしたリポジトリの内容が編集可能なことを確認する。



## リモートリポジトリのクローン

- 以降の「ローカルリポジトリへのコミット」→「リモートリポジトリへのプッシュ」は、ローカルリポジトリの操作と同じ。