

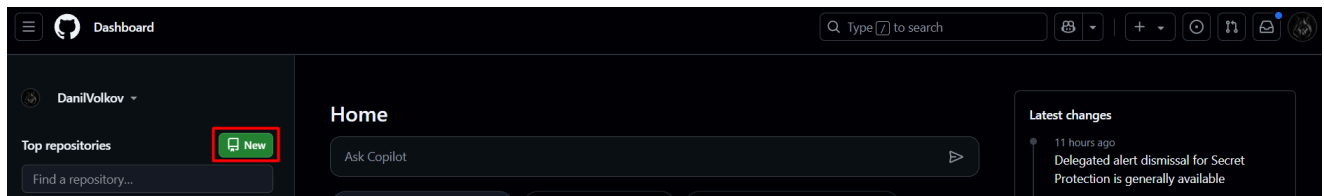
Домашнее задание 3

Домашнее задание состоит из 3 частей:

1. Создание репозитория в GitHub
2. Клонирование репозитория
3. Отработка взаимодействия

Часть 1: Создание репозитория в GitHub

1. Перейдите в GitHub: <https://github.com/>
2. Нажмите на кнопку `New` для создания репозитория.




3. Настройте новый репозиторий:
 - Введите имя репозитория: `TestRepository`
 - Выберите `Private`
 - Добавьте файл `.gitignore` для Python
 - Нажмите кнопку `Create repository`

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 DaniilVolkov ▾

Repository name *

/ TestRepository

✔ TestRepository is available.

Great repository names are short and memorable. Need inspiration? How about **congenial-octo-succotash** ?

Description (optional)

☐

Public

Anyone on the internet can see this repository. You choose who can commit.

☒

Private

You choose who can see and commit to this repository.

Initialize this repository with:

☐

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: Python ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

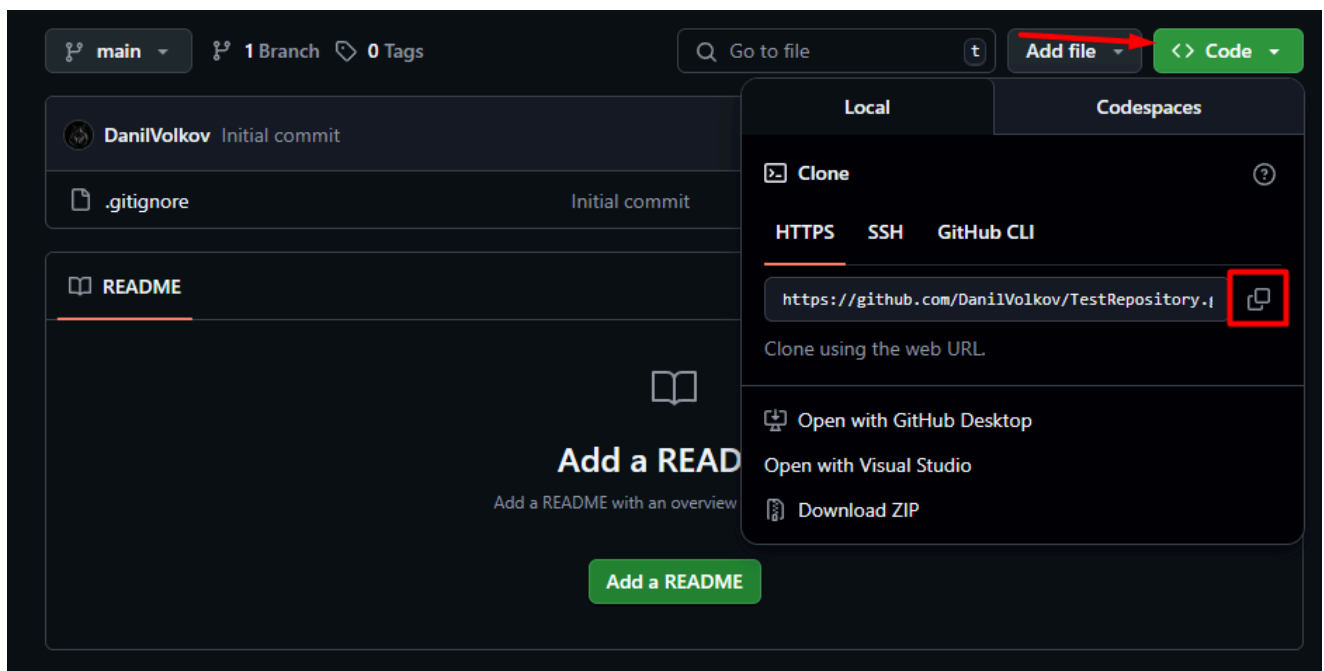
 You are creating a private repository in your personal account.

Create repository

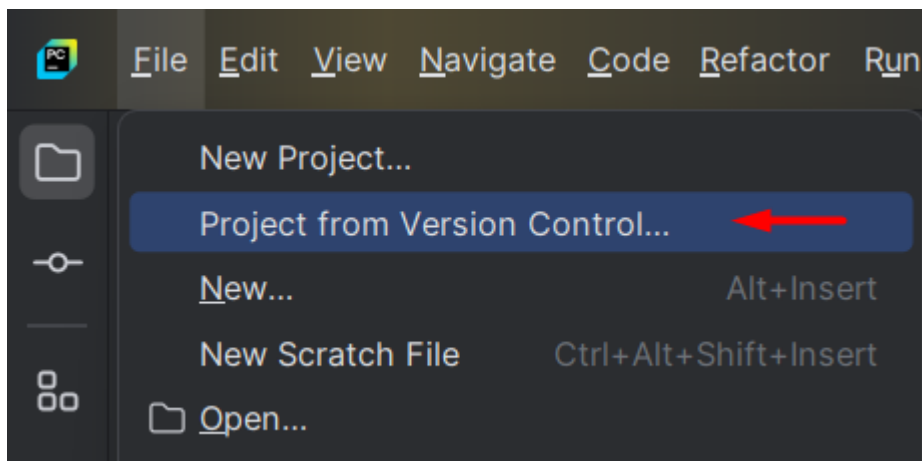
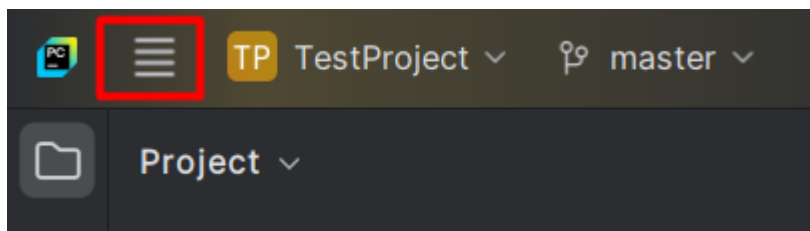
Поздравляем! Репозиторий создан!

Часть 2: Клонирование репозитория

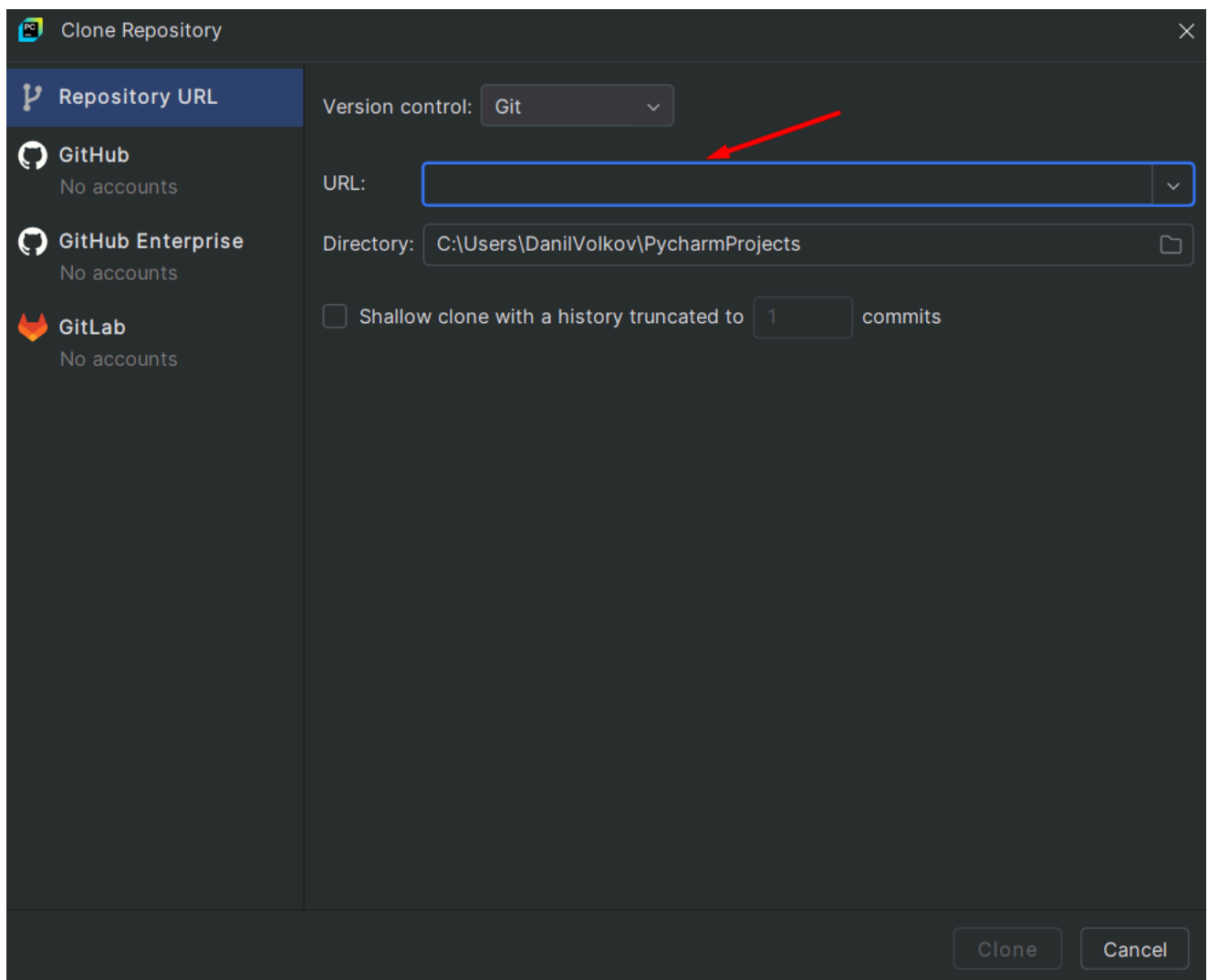
1. В новом репозитории нажмите кнопку `Code` и скопируйте HTTPS-ссылку.



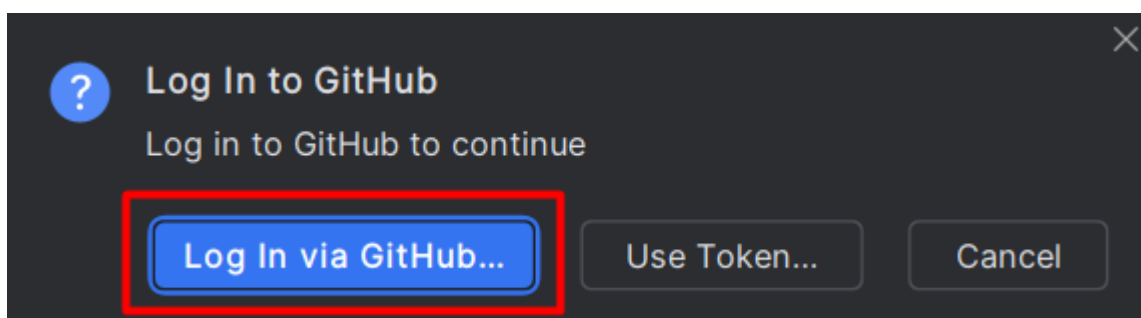
2. Откройте PyCharm и в меню **File** выберите **Project from Version Control...**



3. Вставьте скопированный URL и нажмите **Clone**



4. При первом подключении из PyCharm потребуется авторизация в GitHub.
5. Для авторизации можно использовать токен или вход через github.
Выбираем вход через GitHub.



6. Данная кнопка перенаправляет на web-страницу с авторизацией.
На странице нужно нажать кнопку `Authorize in GitHub`

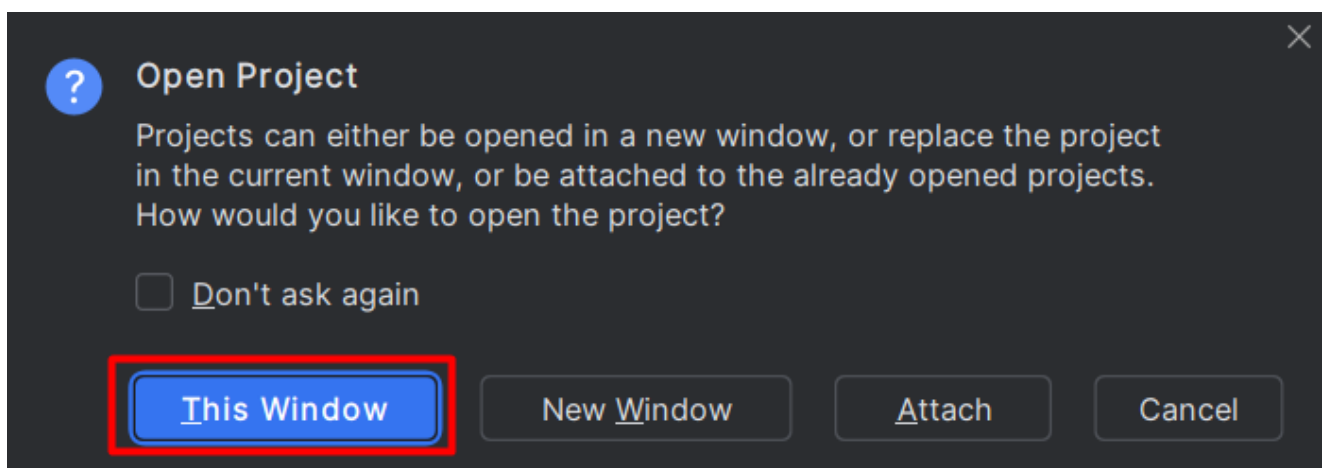
Please continue only if this page is opened from a [JetBrains IDE](#).

Authorize in GitHub

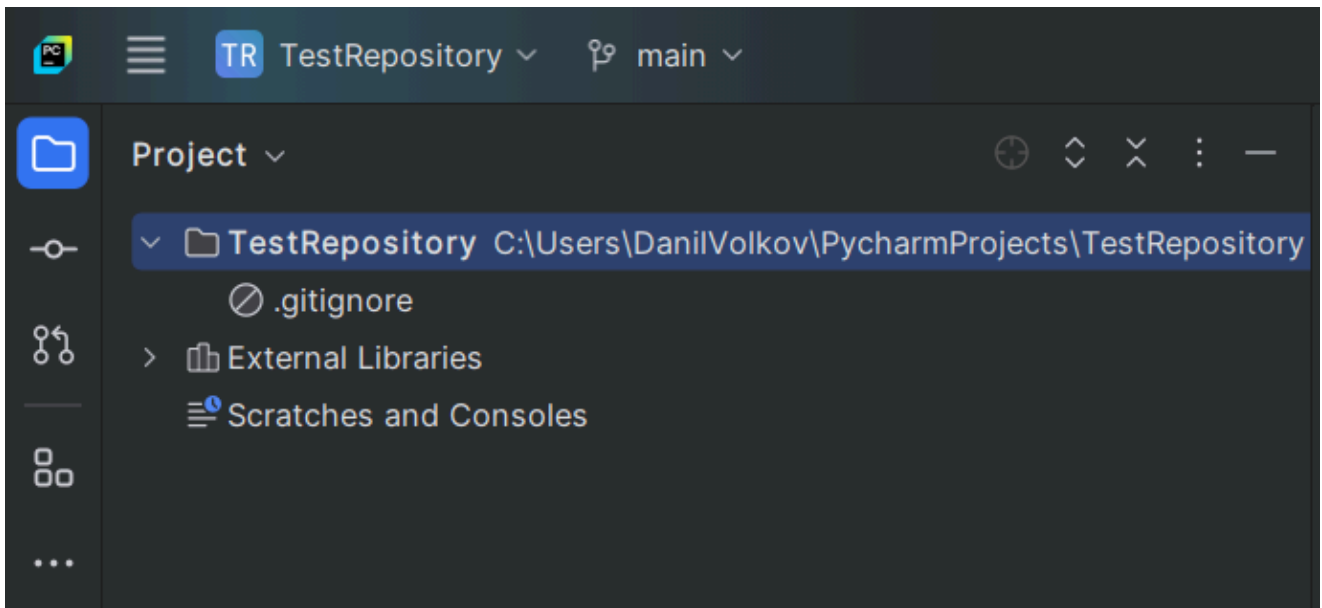
7. После должно появиться сообщение об успешной авторизации.

You have been successfully authorized in GitHub. You can close the page.

8. Теперь можно перейти в PyCharm и открыть проект.



9. В открытом проекте должен лежать 1 файл - `.gitignore`



10. Теперь GitHub и PyCharm связаны.

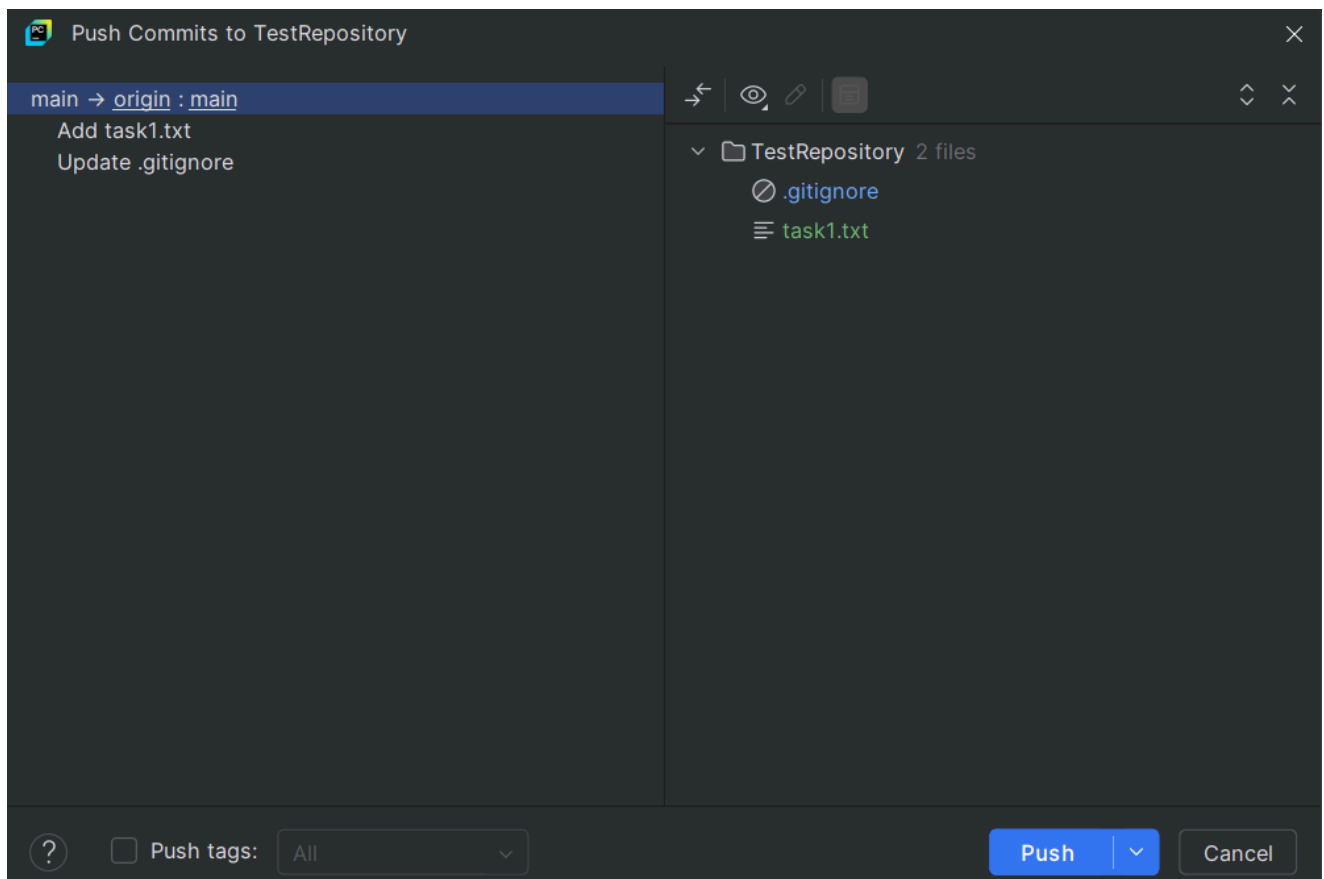
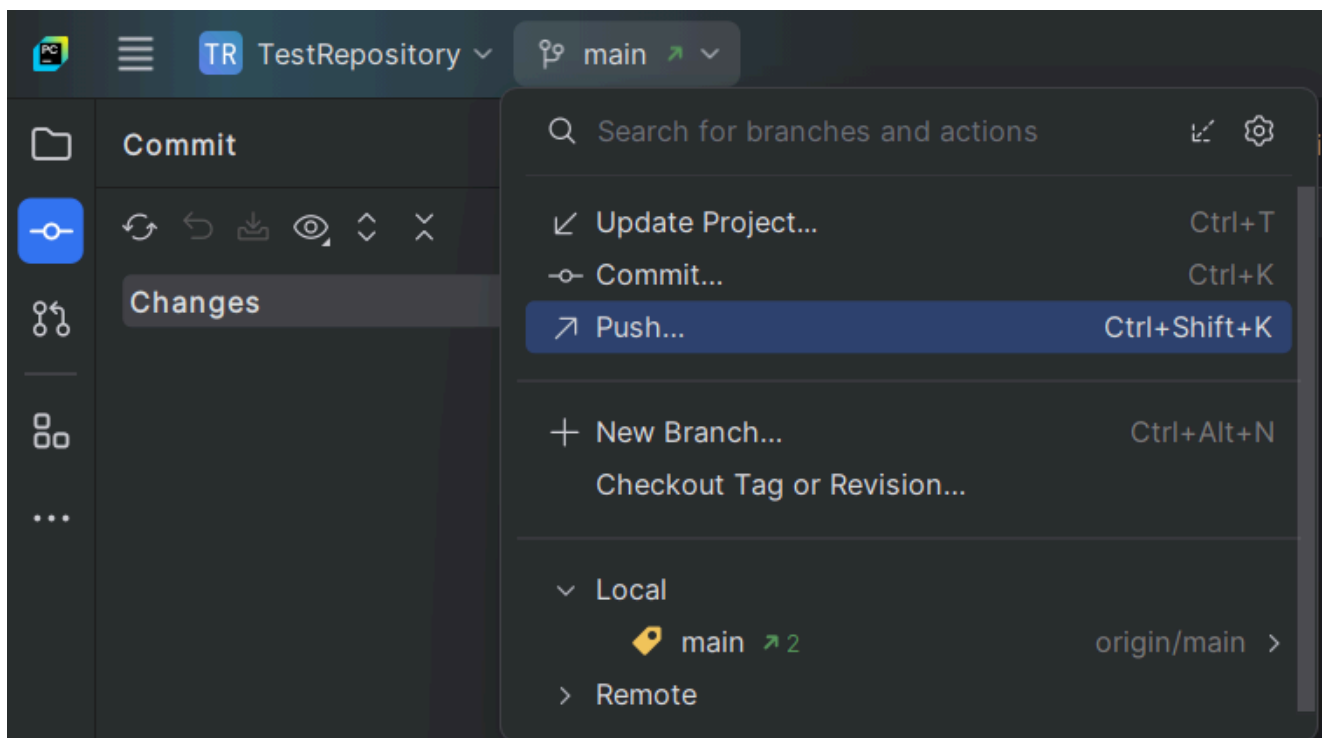
Добавить или удалить аккаунт можно в пункте меню `Settings -> Version Control -> GitHub`

Часть 3: Отработка взаимодействия

1. Откройте файл `.gitignore`. Найдите пункт `PyCharm`. В данном пункте раскомментируйте строку `.idea/`.
Готовый вариант должен выглядеть как на изображении ниже.

```
# PyCharm
# JetBrains specific template is maintained in a separate JetBrains.gitignore that can
# be found at https://github.com/github/gitignore/blob/main/Global/JetBrains.gitignore
# and can be added to the global gitignore or merged into this file. For a more nuclear
# option (not recommended) you can uncomment the following to ignore the entire idea folder.
.idea/
```

2. Сделайте коммит с описанием `Update .gitignore`
3. Создайте файл `task1.txt` и добавьте его в отслеживаемые.
4. Добавьте в файл `task1.txt` содержимое: Это было бы мученье, но я настроил подключение
5. Сделайте коммит с описанием `Add task1.txt`
6. Теперь при помощи команды `Push` отправьте изменения в локальной ветке `main` на удаленный репозиторий, также в ветку `main`.



7. Перейдите в GitHub, обновите страницу и проверьте изменения.

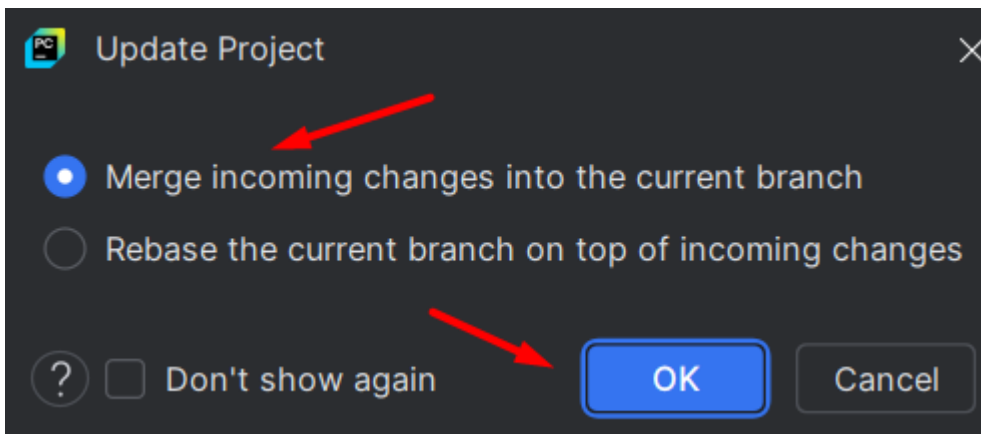
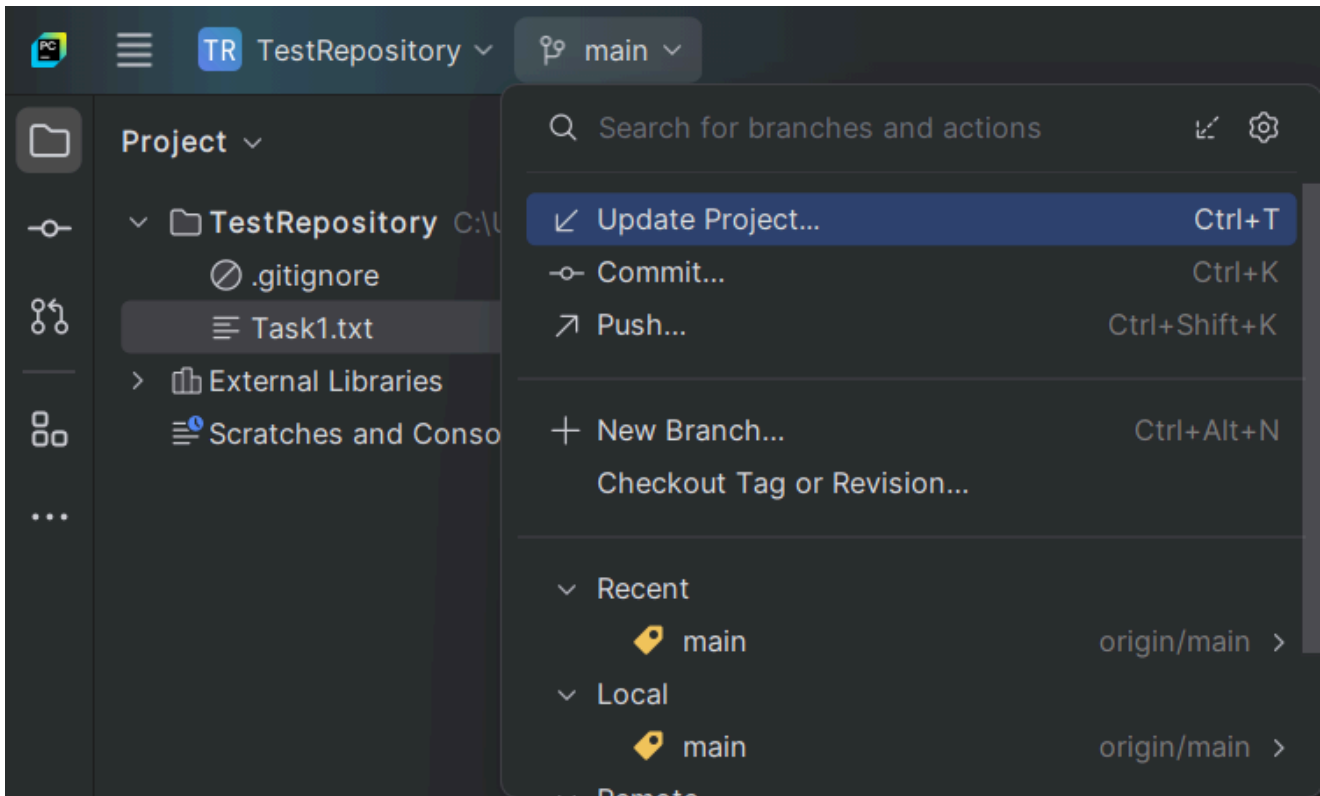
Запомните! Отправлять коммиты сразу в ветку main - плохая практика!
В дальнейшем, в процессе выполнения домашних заданий на курсе так делать запрещено!

Если в ходе выполнения домашних работ вы случайно зальете изменения сразу в ветку main,

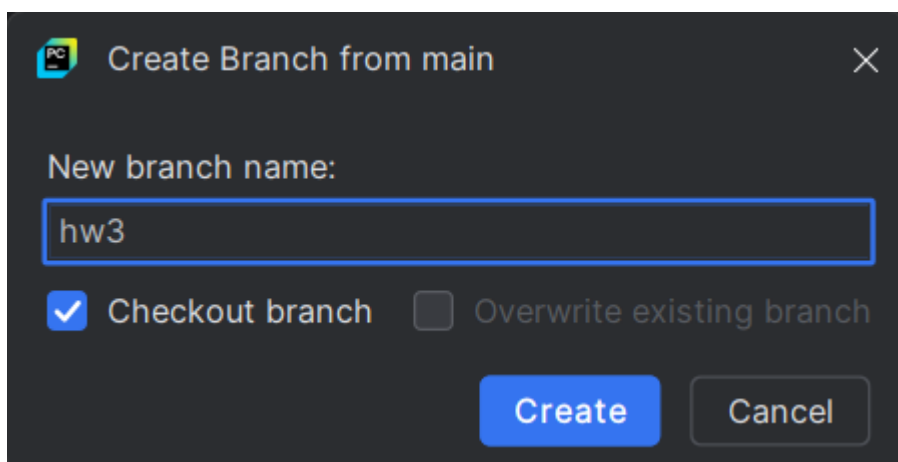
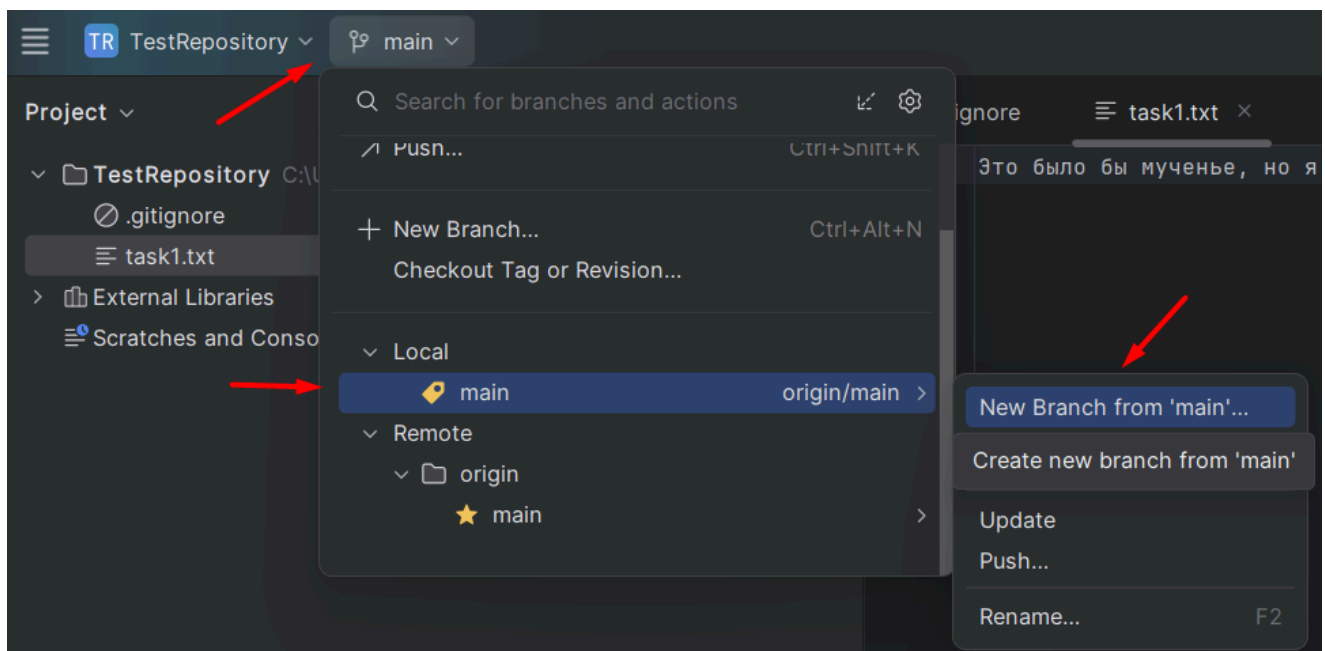
нужно будет откатить изменения и сделать корректный Pull Request.

8. Перед тем, как создавать новую ветку, нужно загрузить обновленную ветку main из удаленного репозитория.

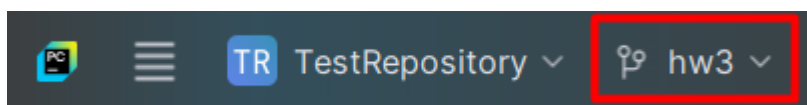
Для этого нажмите на ветку main и выберите Update Project...



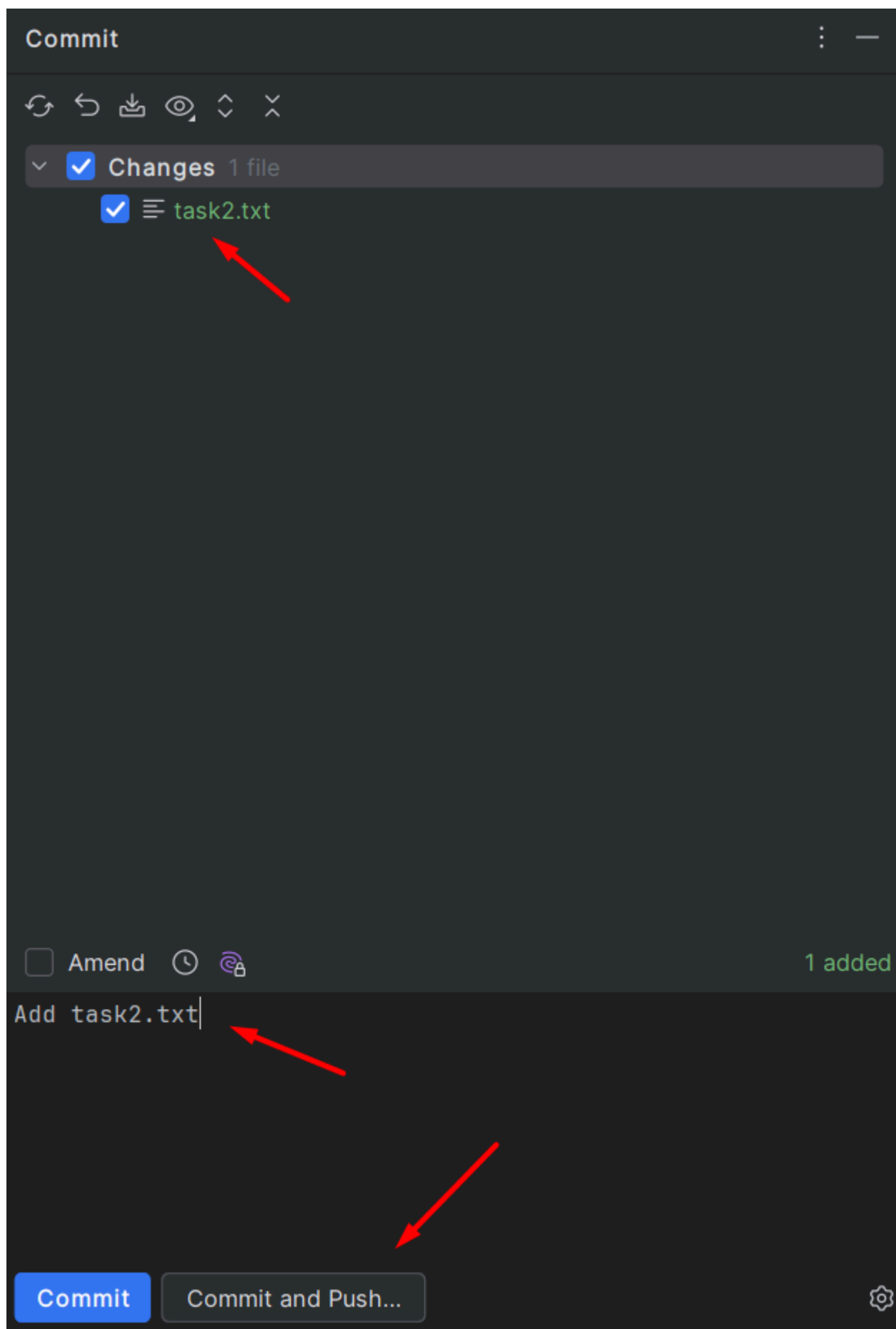
9. Создайте ветку hw3 от ветки main



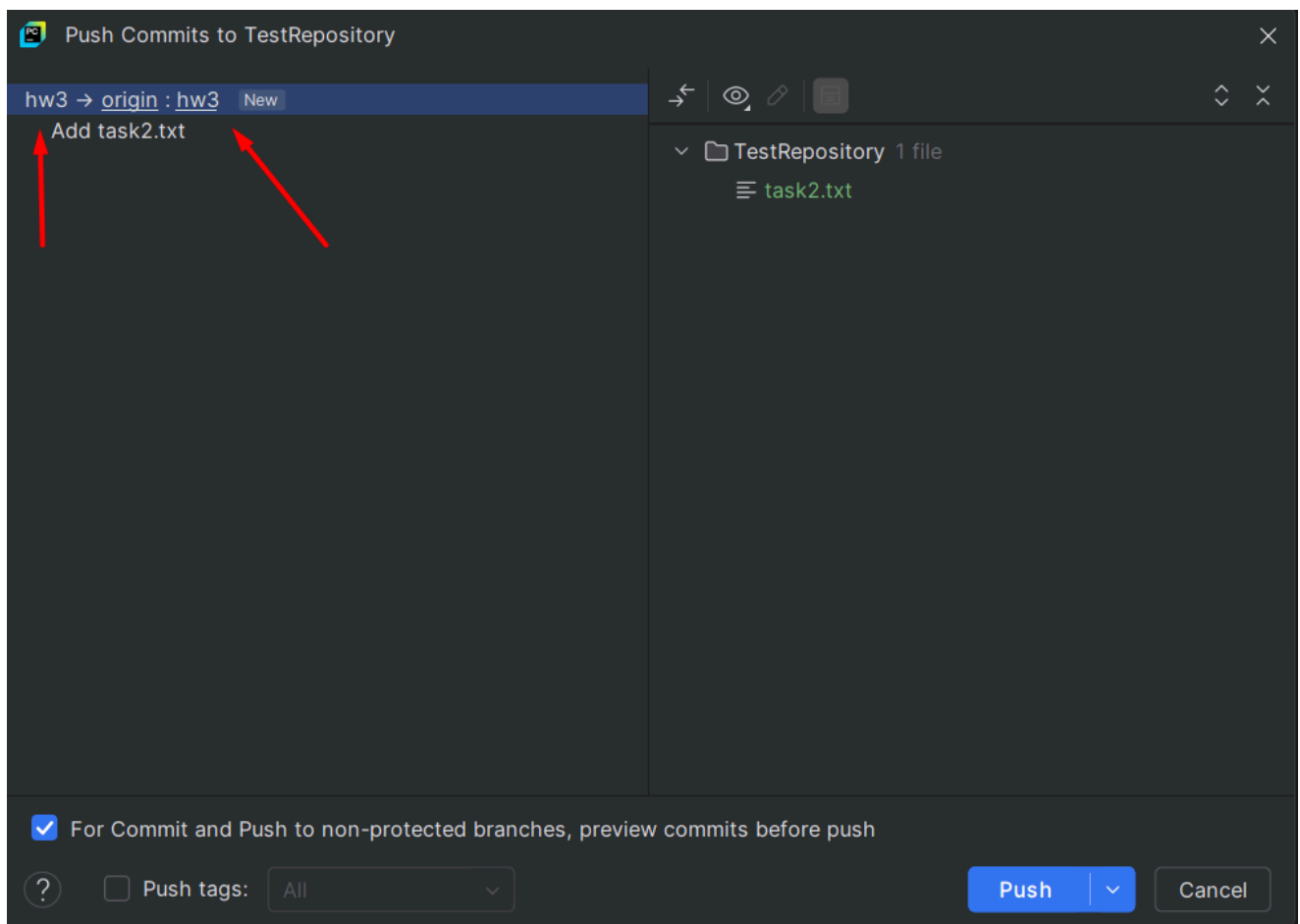
10. Проверьте, что вы находитесь на ветке `hw3`



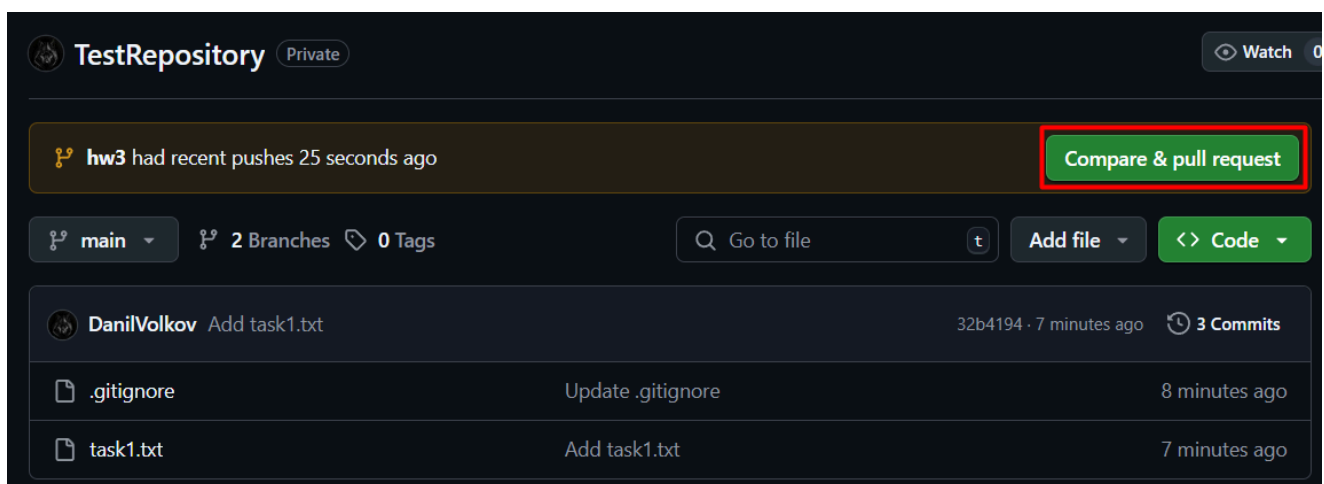
11. Создайте файл `task2.txt` со следующим содержимым: `Изучаю, познаю, до конца точно дойду` и добавьте его в stage git.
12. Сделайте коммит с описанием `Add task2.txt` и отправьте изменения на удаленный репозиторий в GitHub.
Данные действия можно сделать одной кнопкой - `Commit and Push...`



13. Убедитесь, что на удаленном репозитории будет создана новая ветка hw3



14. Перейдите в GitHub и откройте Pull Request (PR).

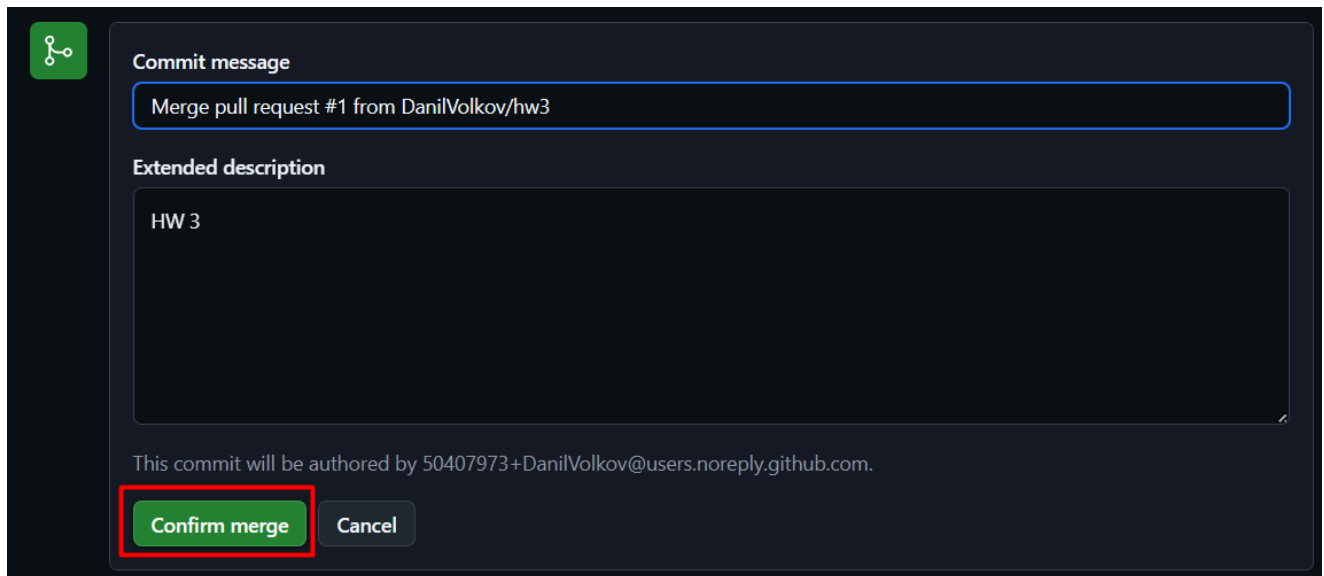


15. Настройте PR. Убедитесь, что изменения будут отправляться из ветки `hw3` в ветку `main`.

Введите название `hw 3`. В меню `Reviewers` вы в будущем будете выбирать ваших преподавателей.

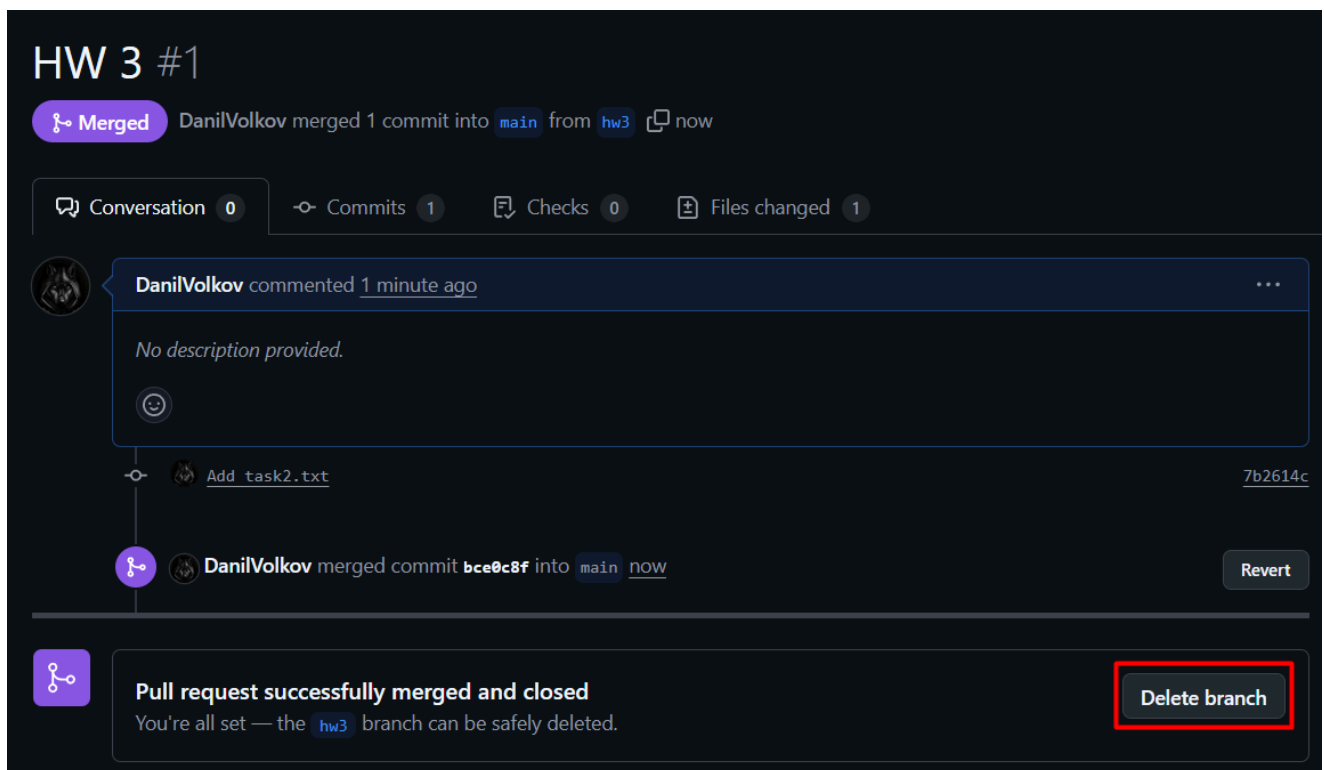
Пока что оставьте это поле без изменений. Нажмите кнопку `Create pull request`

После можете ввести сообщение к коммиту, описание и нажать кнопку `Confirm merge`



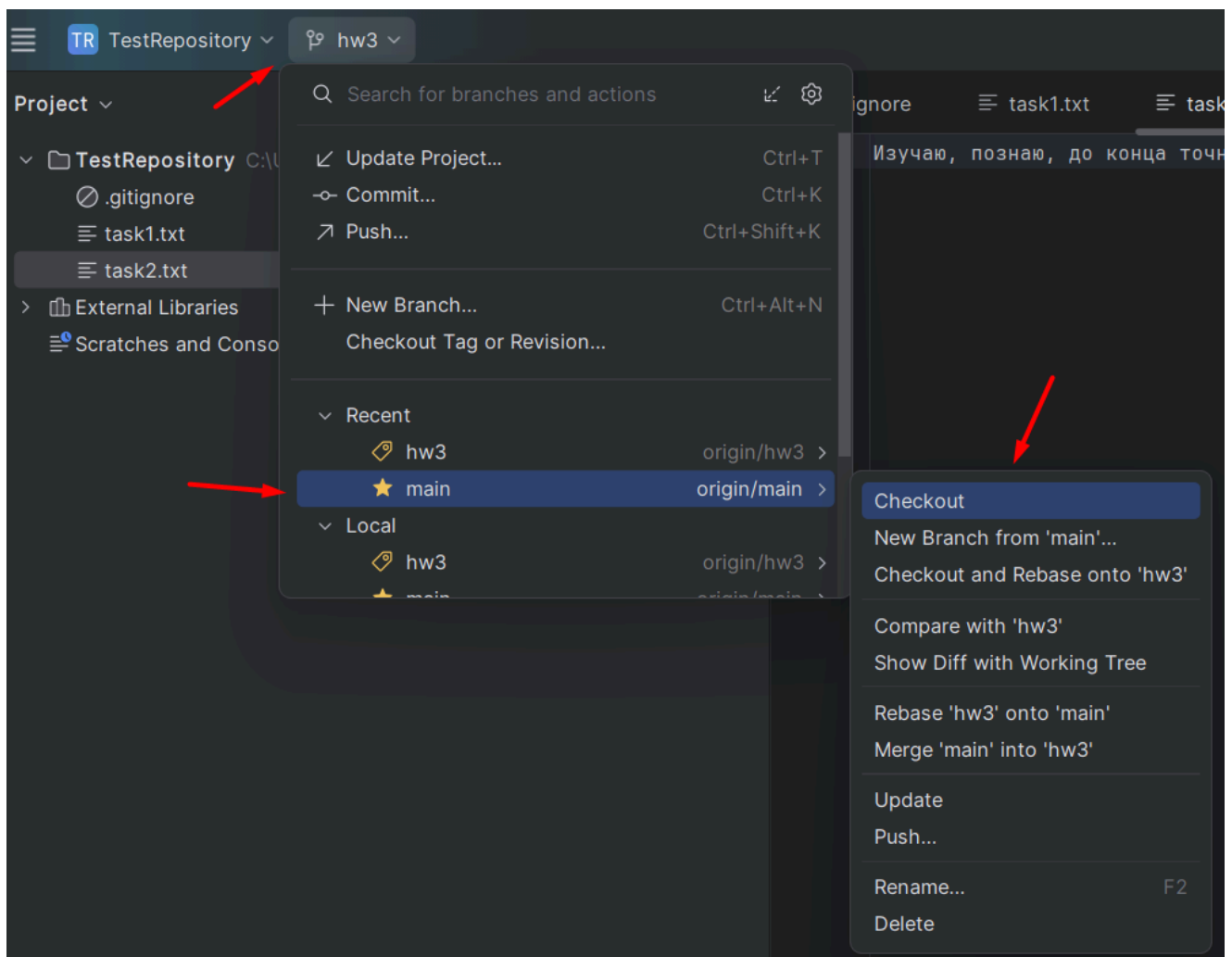
The screenshot shows the GitHub merge confirmation interface. At the top, there's a green icon with a branch symbol. Below it, the 'Commit message' field contains 'Merge pull request #1 from DanilVolkov/hw3'. The 'Extended description' field contains 'HW 3'. Below these fields, a line of text states: 'This commit will be authored by 50407973+DanilVolkov@users.noreply.github.com.'. At the bottom, there are two buttons: 'Confirm merge' (highlighted with a red rectangle) and 'Cancel'.

18. Теперь можно безопасно удалить ветку `hw3` и на Github, и в локальном репозитории.

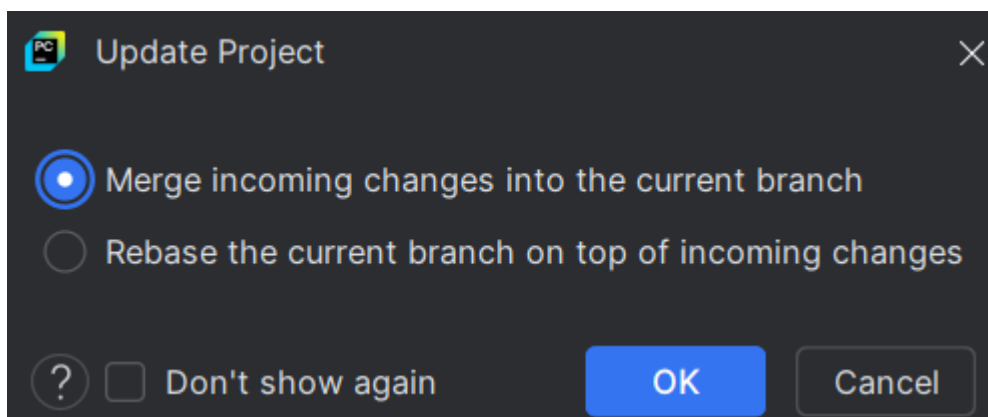


The screenshot shows the GitHub pull request page for 'HW 3 #1'. The status is 'Merged'. Below the status, it says 'DanilVolkov merged 1 commit into main from hw3 now'. There are tabs for 'Conversation' (0), 'Commits' (1), 'Checks' (0), and 'Files changed' (1). A comment by 'DanilVolkov' is visible, stating 'No description provided.' Below the comment, a commit is shown: 'Add task2.txt' with commit hash '7b2614c'. Below the commit, it says 'DanilVolkov merged commit bce0c8f into main now'. At the bottom, there's a message: 'Pull request successfully merged and closed. You're all set — the hw3 branch can be safely deleted.' To the right of this message, there is a button 'Delete branch' (highlighted with a red rectangle).

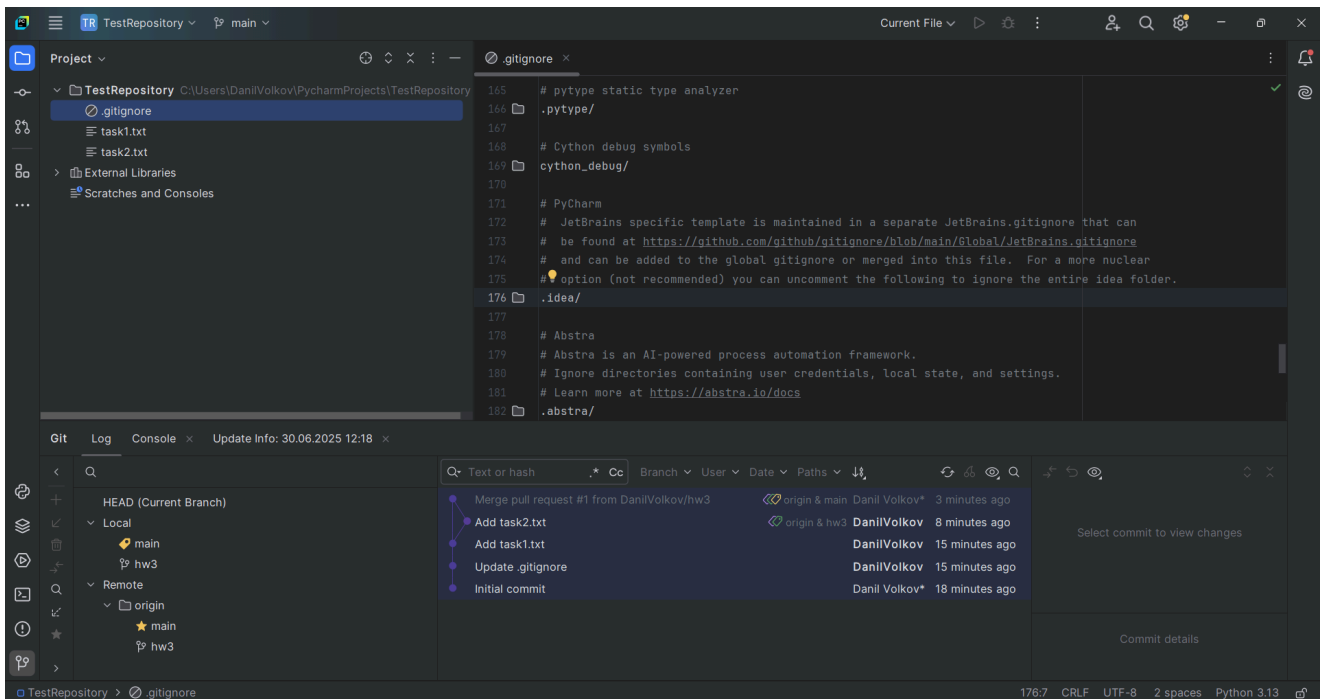
19. Перейдите в PyCharm и переключитесь на ветку `main`



20. Файлик `task2.txt` пропал, так как в локальной ветке `main` еще нет изменений из ветки `main` на Github.
21. Обновите локальную ветку `main` с помощью `Update Project...`

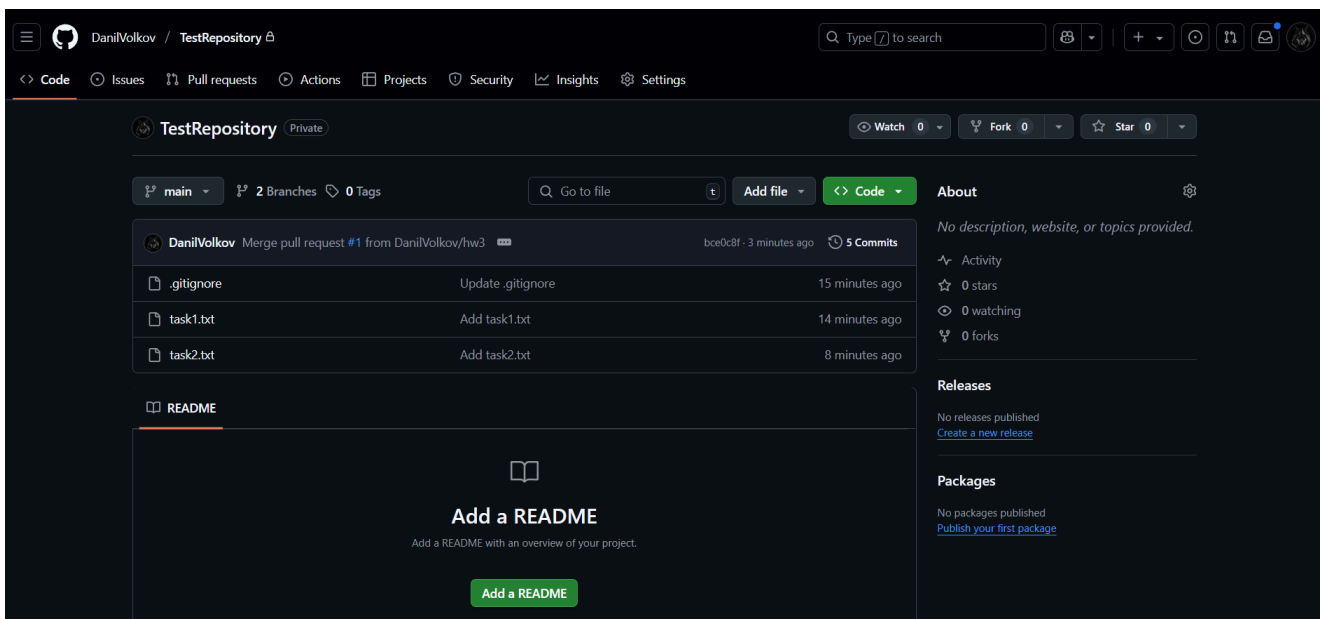


22. Приложите в iSpring скриншот PyCharm с меню `Git` и файловой структурой.
Пример:



23. Также приложите скриншот с GitHub.

Пример:



Полезные материалы:

- Интерактивный курс по git: https://learngitbranching.js.org/?locale=ru_RU.
- Руководство по работе с Git и GitHub через интерфейс PyCharm: <https://teletype.in/@kurilkv/CodimGitHub#H084>
- [Что делать если отправил изменения сразу в ветку main](#)
- [Что делать если сделал Merge pull request до проверки](#)