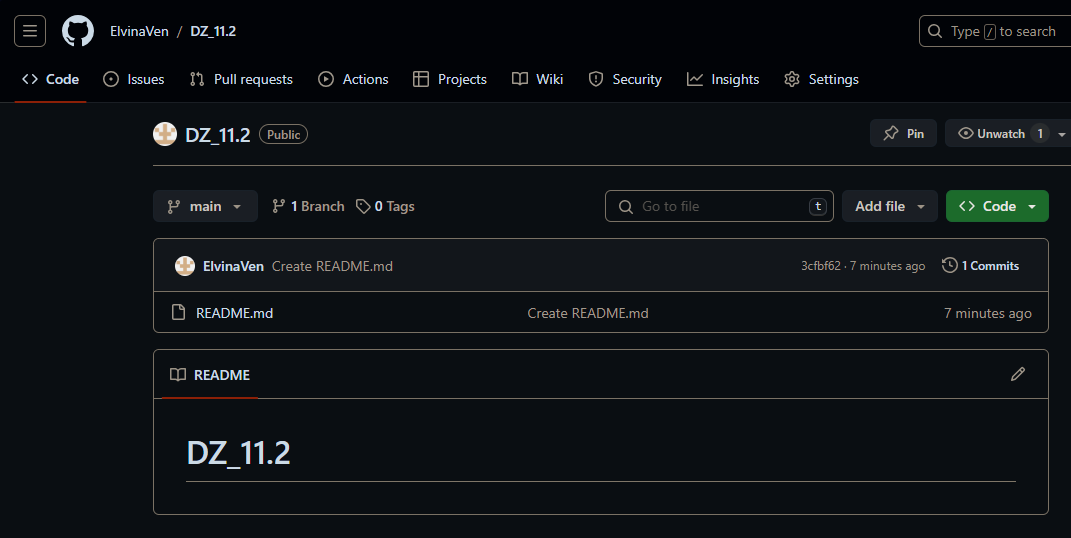
**Задание 1. Получение проекта**

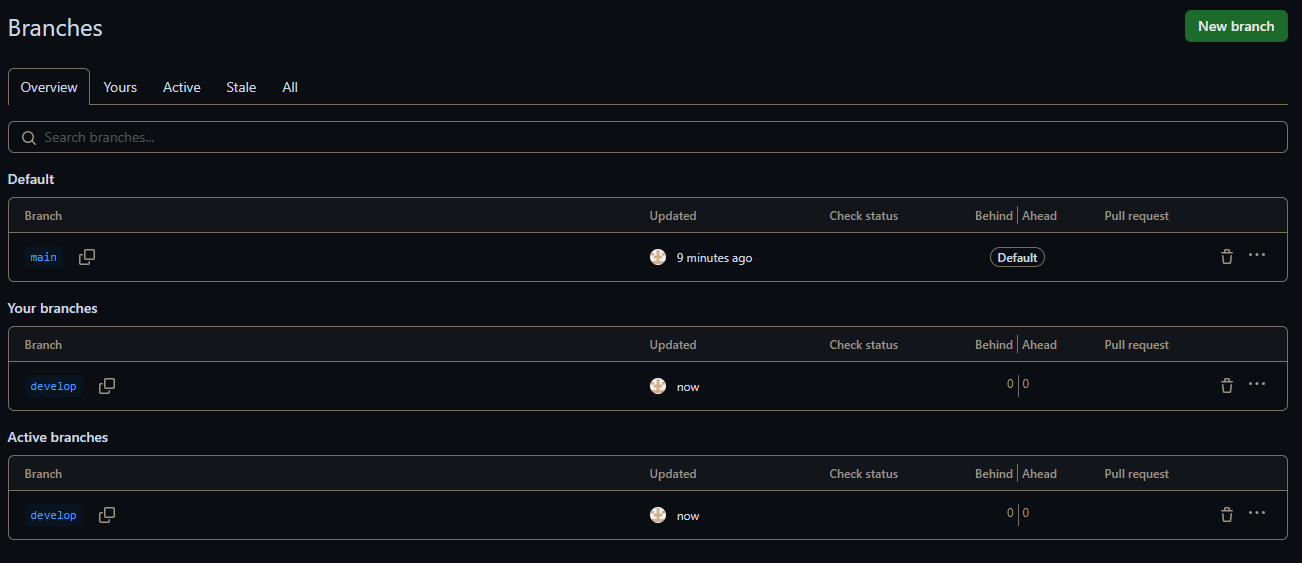
Выполните следующие шаги:

**На GitHub:**

1. Создайте новый репозиторий с файлом readme.md.



1. Создайте ветку develop.

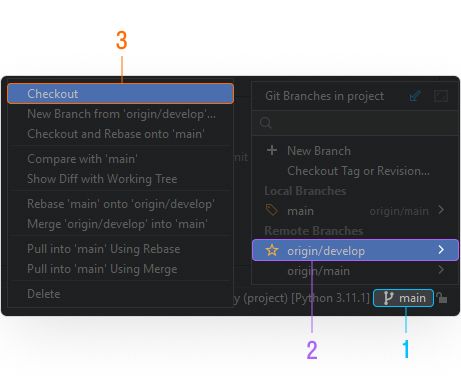


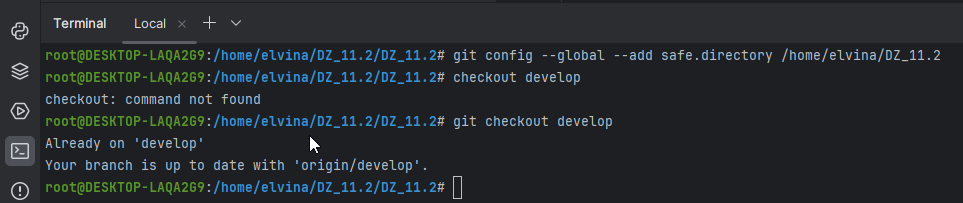
**Локально:**

1. Клонируйте репозиторий с GitHub на компьютер.



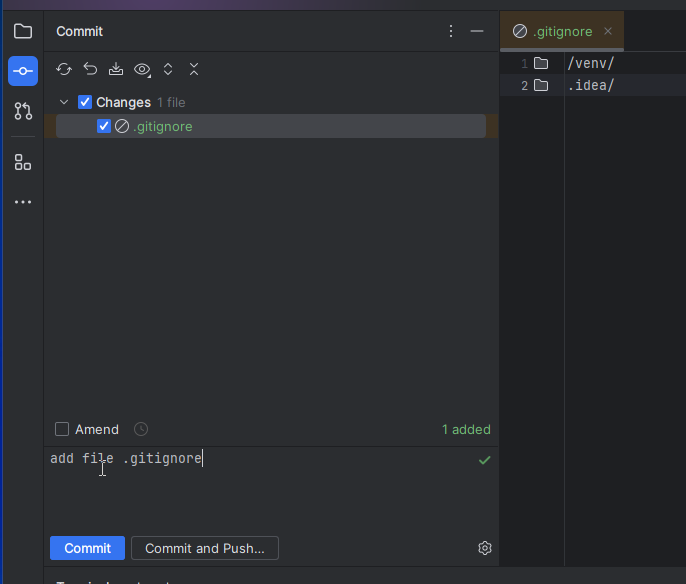
1. Откройте проект в PyCharm (3-й и 4-й пункт можно сразу сделать из PyCharm).
2. Сделайте checkout удаленной ветки develop (станет текущей).



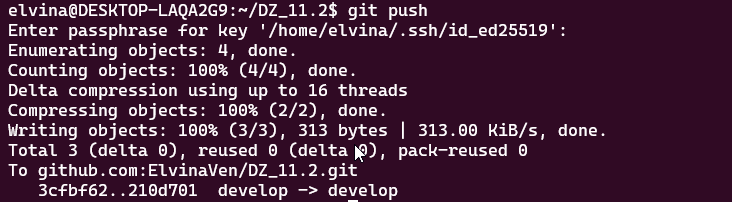


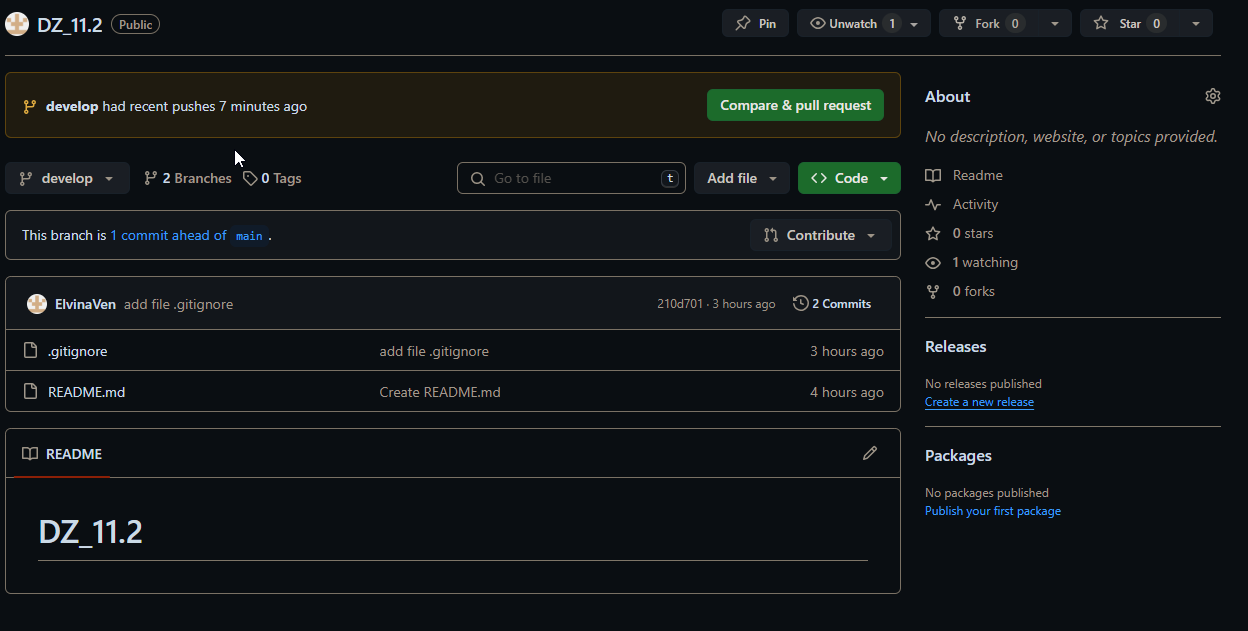
1. Создайте, заполните и закоммитьте файл .gitignore.

Подумайте, какие файлы надо игнорировать при работе из PyCharm. Важно, чтобы лишние файлы, которые создает PyCharm, не попадали в коммиты.



1. Отправьте изменения на GitHub.





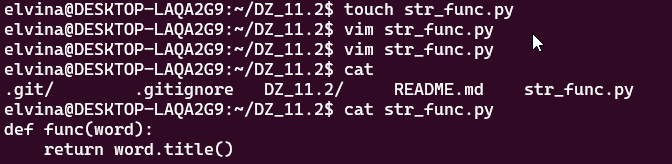
**Задание 2. Первый пул-реквест**

Выполните следующие шаги:

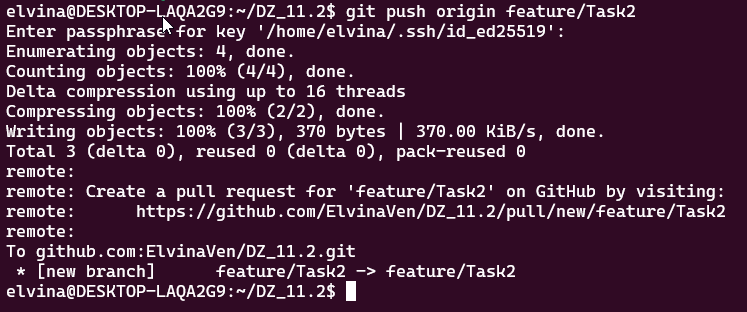
1. Создайте и переключитесь на ветку feature/Task2.

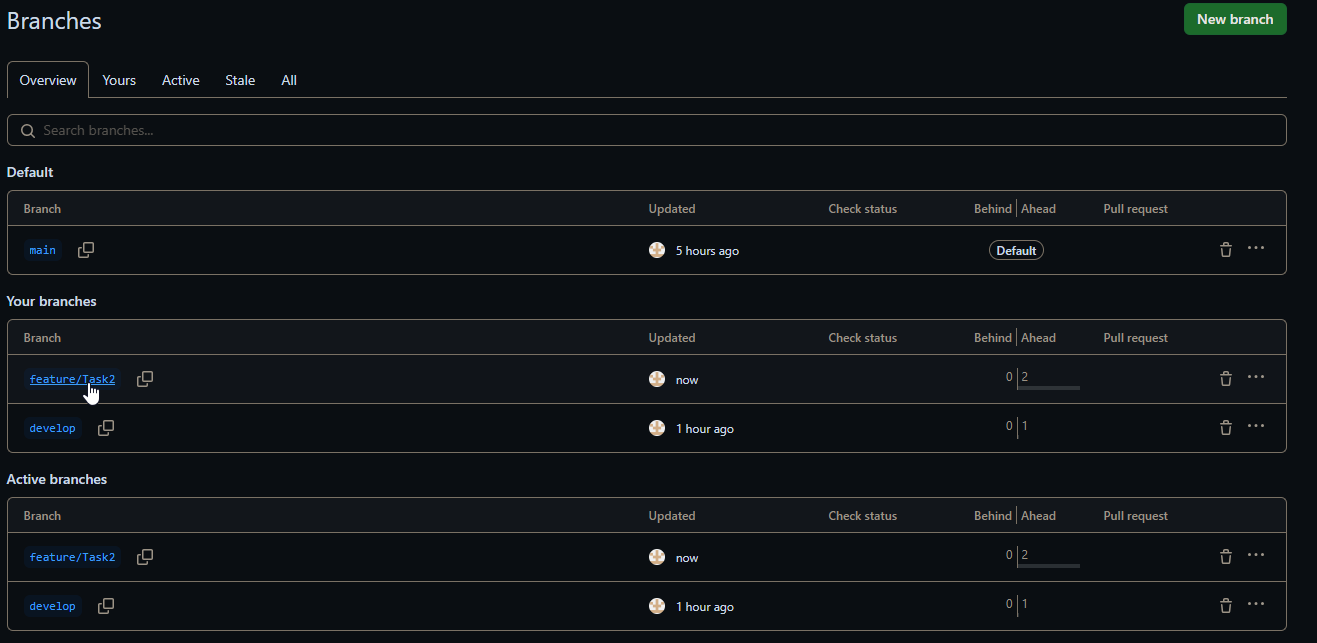


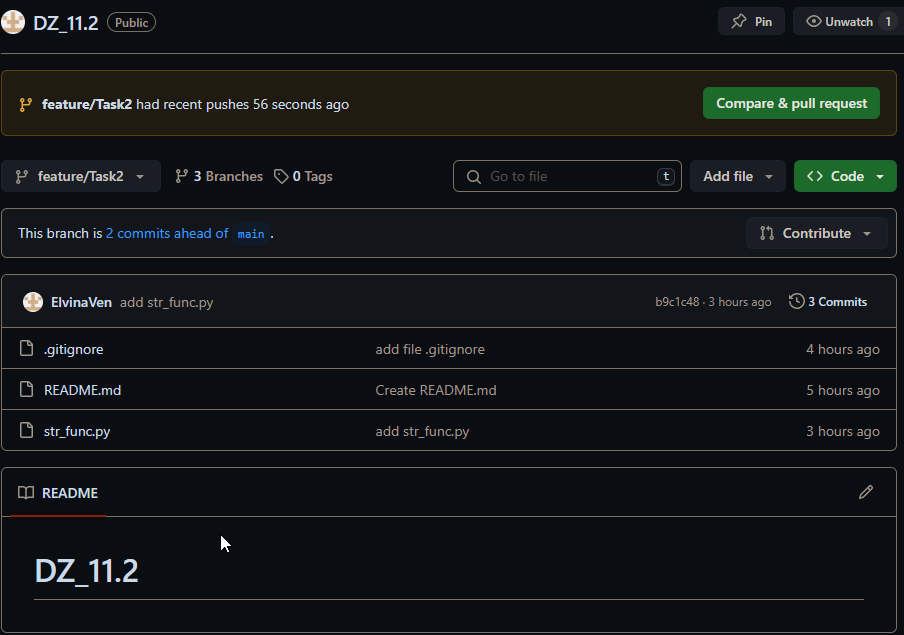
1. Создайте файл str\_func.py и в нем функцию, которая принимает на вход строку и возвращает ее со всеми заглавными буквами.



1. Закоммитьте проделанную работу и отправьте изменения на GitHub.





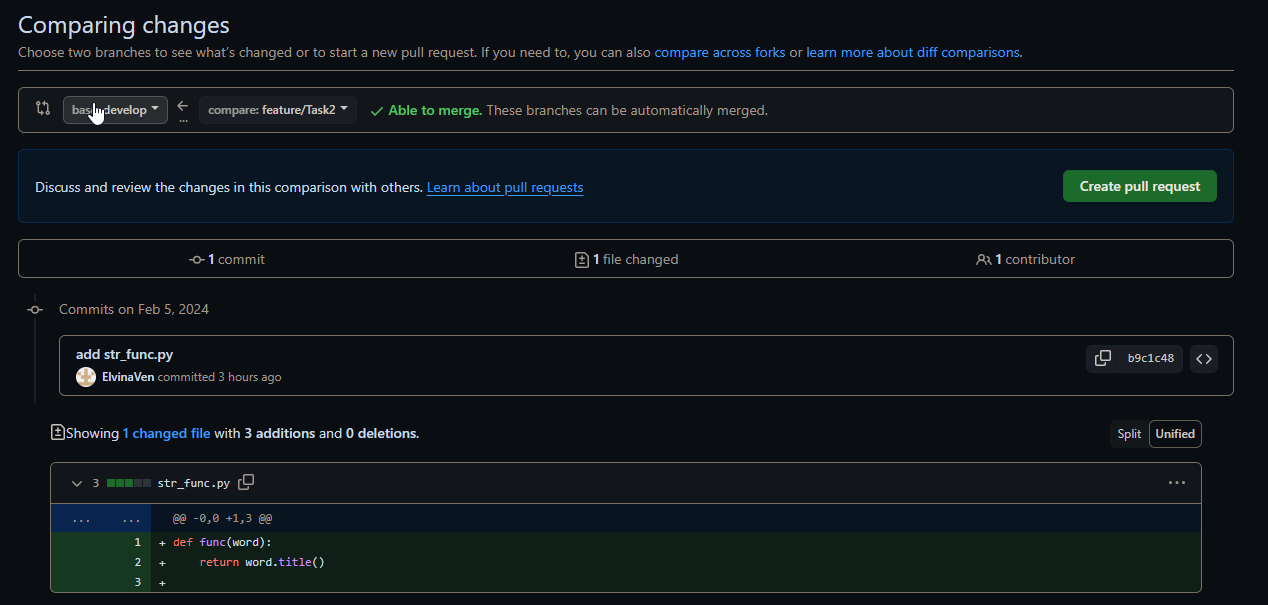


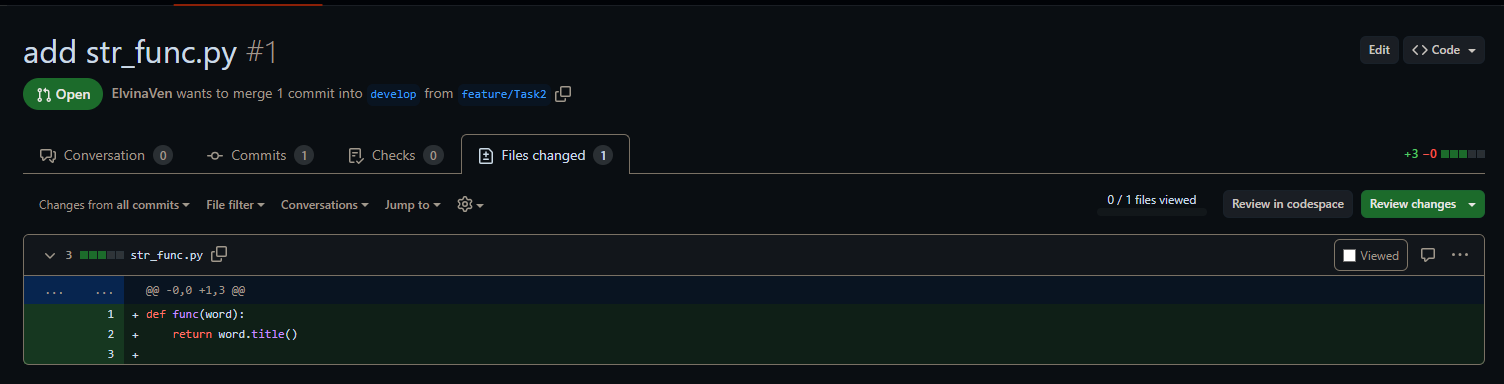
**На GitHub:**

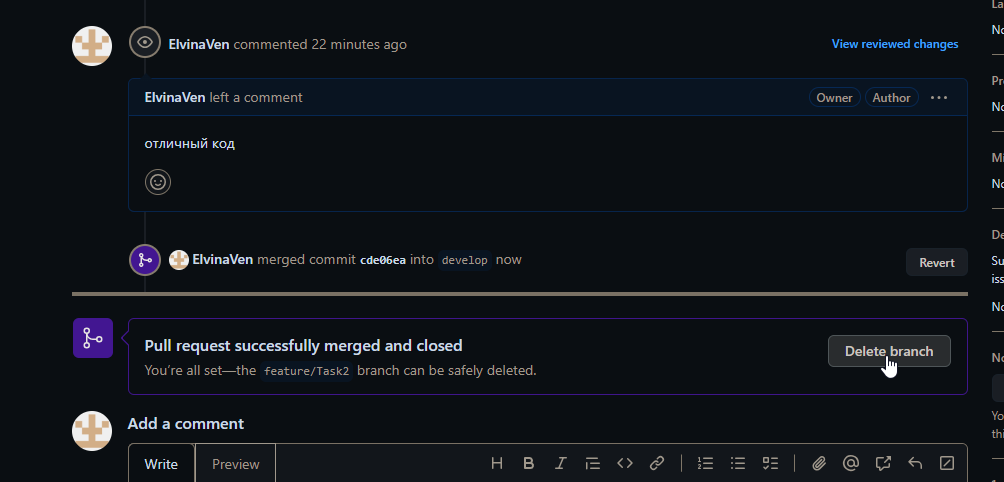
1. Сделайте pull request из ветки feature/Task2 в develop.

После pull request вашему тимлиду приходит уведомление. Он смотрит, что было сделано, вносит комментарии, отклоняет или принимает пул-реквест.

В учебных целях действия тимлида вы делаете самостоятельно.



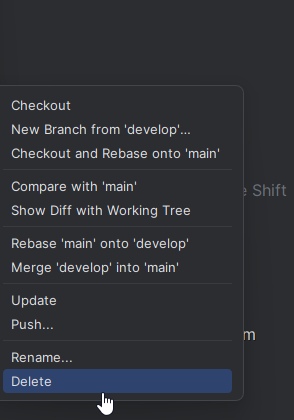




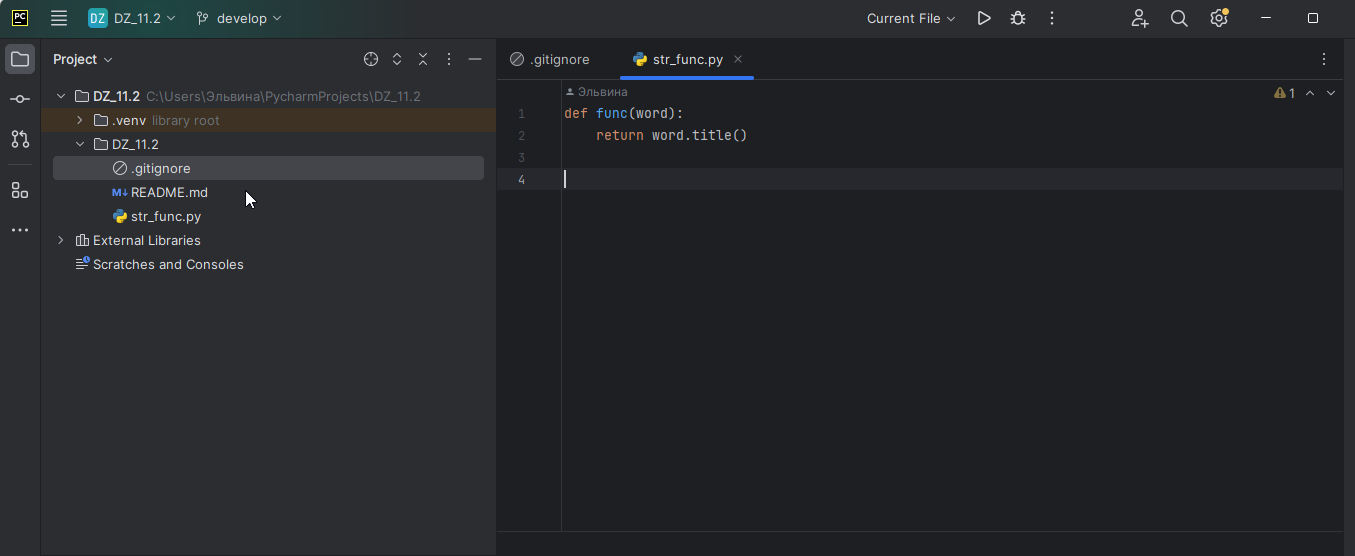
1. Сделайте за тимлида мердж с удалением ветки feature/Task2.

**Локально:**

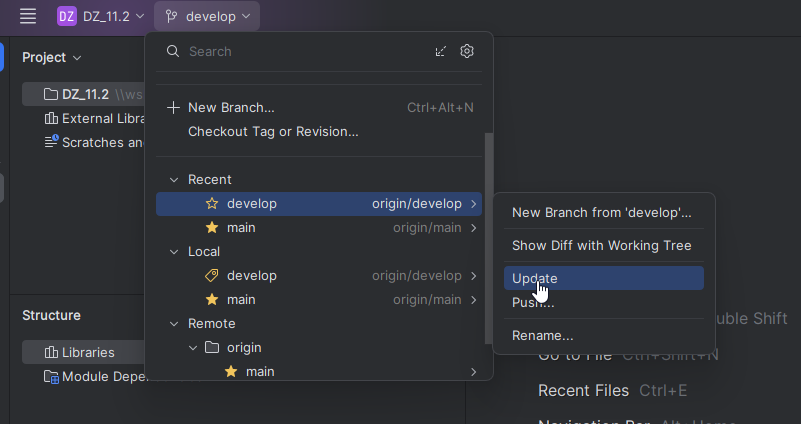
1. Удалите локальную ветку feature/Task2.



1. Обновите локальную ветку develop.







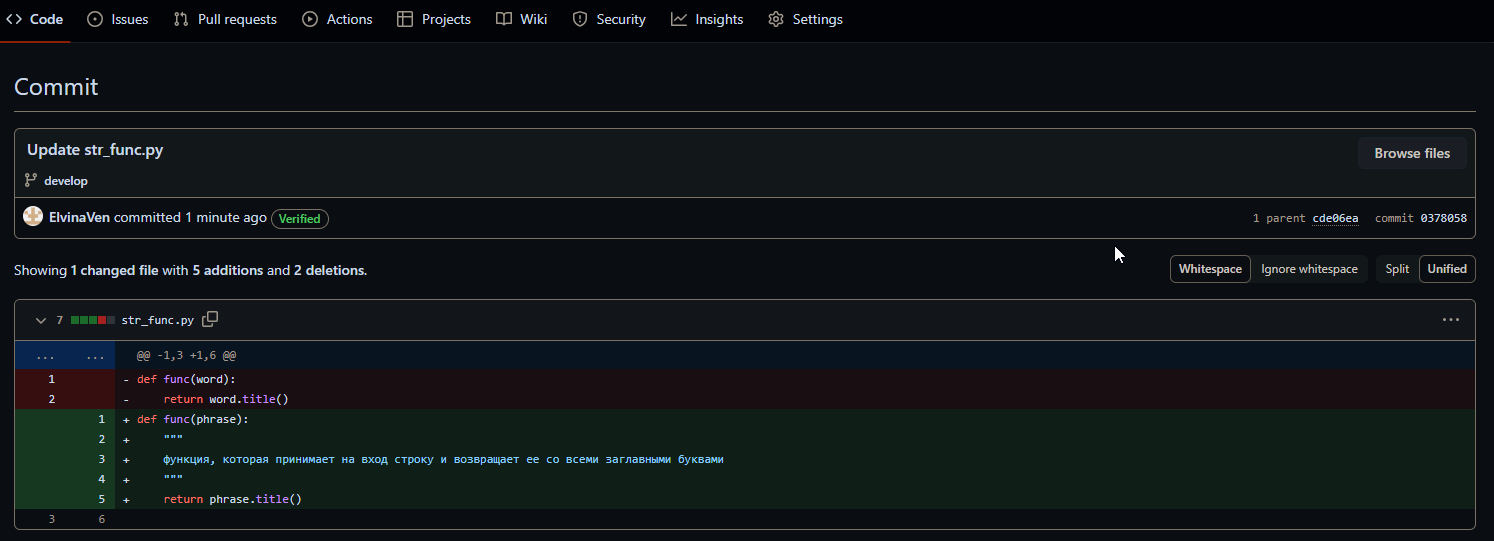
**Задание 3. Наращивание функционала**

Тимлид дал задачу написать еще одну функцию и попросил не забывать добавлять описание к функциям (docstring).

Выполните следующие шаги:

**На GitHub:**

1. Добавьте docstring к функции из предыдущего задания (прямо в ветке develop). Этим вы сымитируете проделанную работу другого разработчика над этим же модулем.



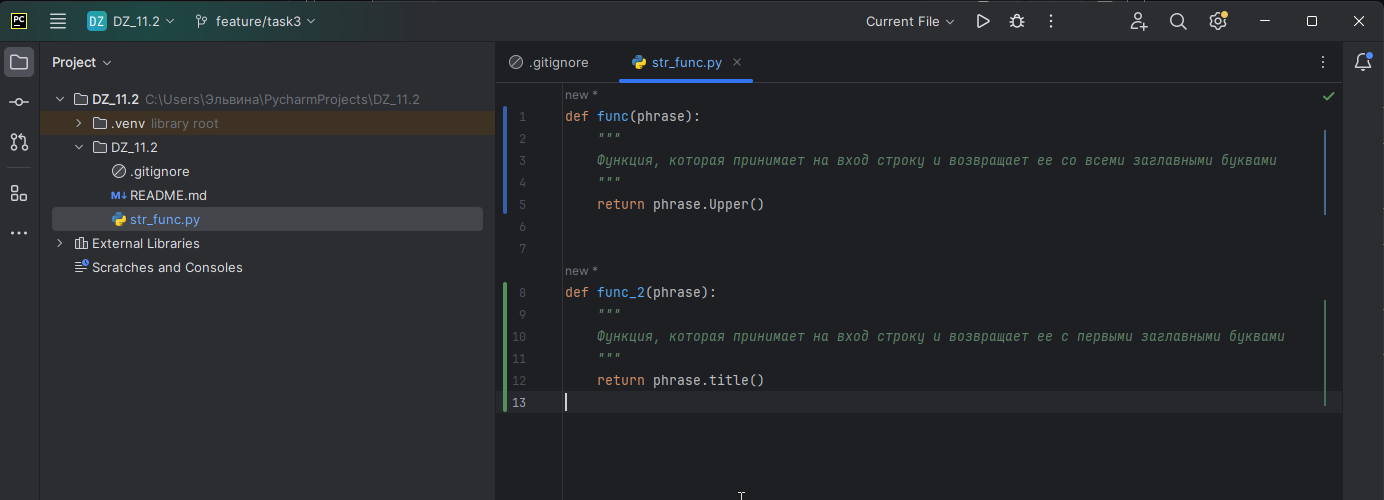
**Локально:**

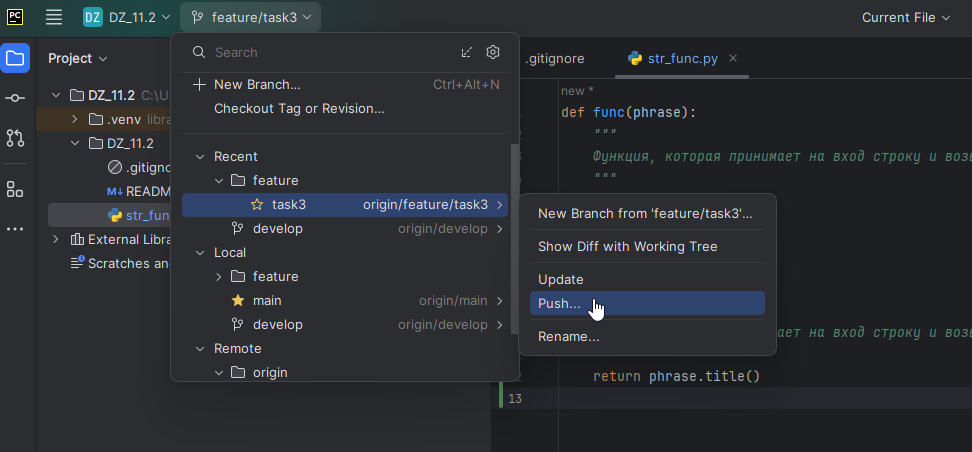
1. Напишите в файле str\_func.py вторую функцию, которая делает заглавными первые буквы каждого слова в строке, поступившей на вход функции. Добавьте к функции docstring.

Выполняйте это задание в новой ветке. Назовите ее, например feature/task3

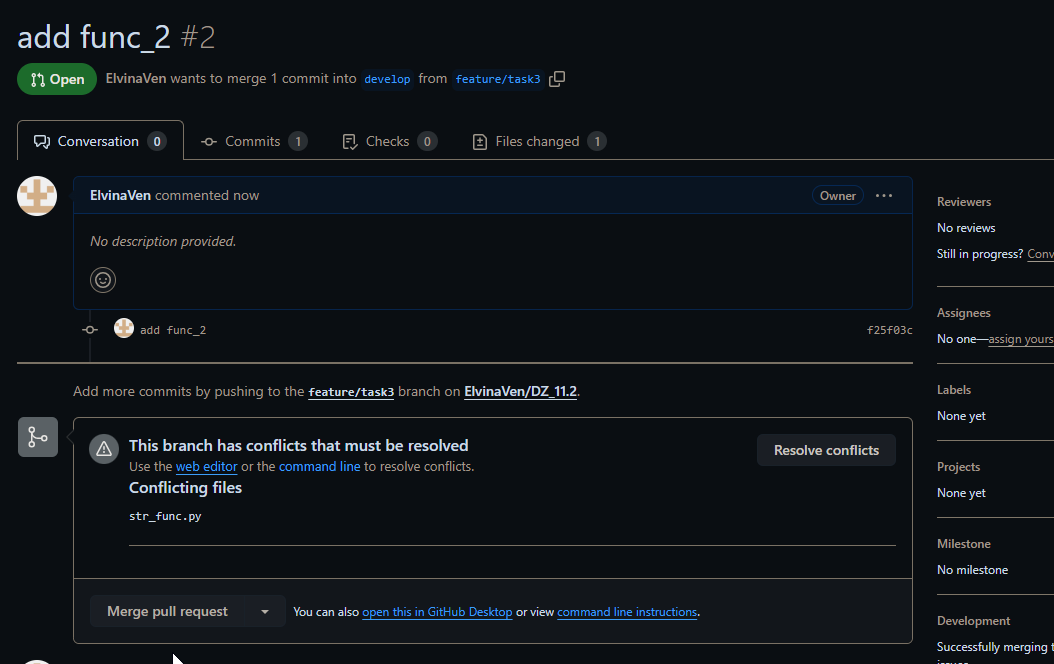
1. Добавьте docstring к функции из Задания 2.

Делаем коммит и пушим





1. Отправьте проделанную работу коллегам на код-ревью (т. е. pull-request, который должен показать, что есть конфликт: **This branch has conflicts that must be resolved**).

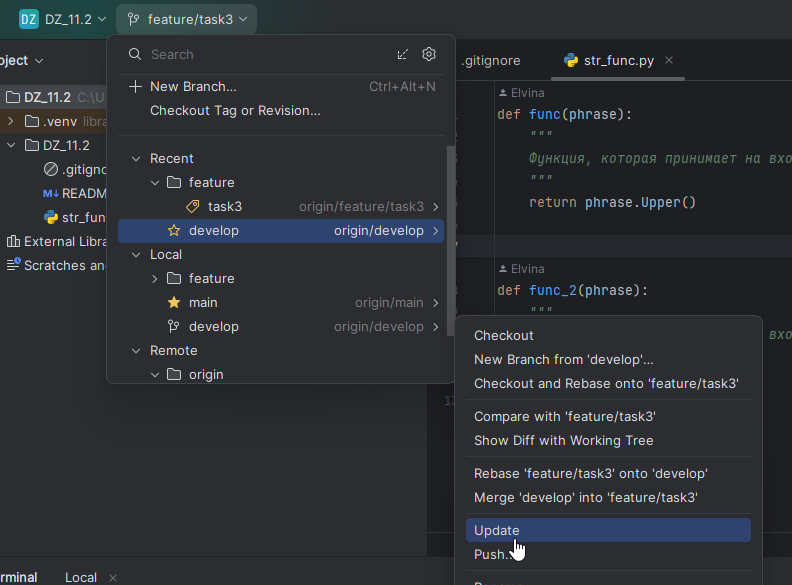


**Задание 4. Решение конфликтов**

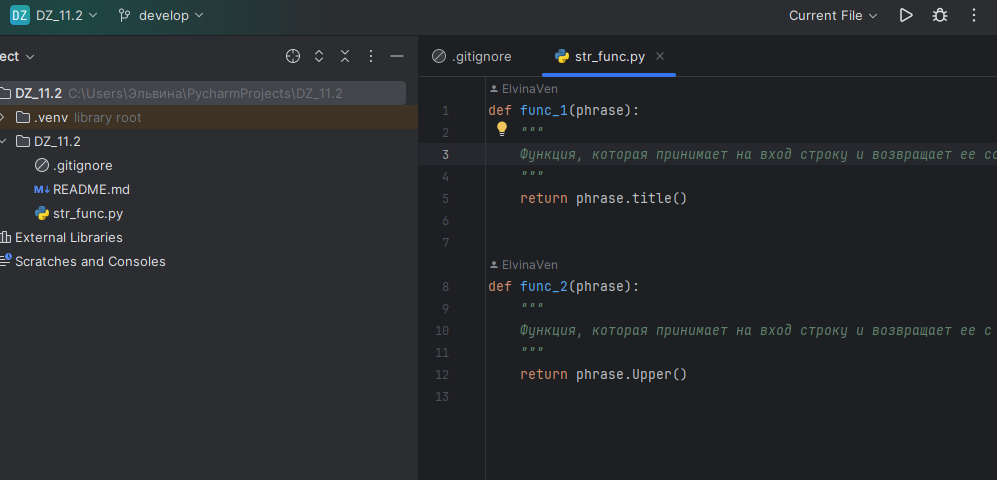
Ваш пул-реквест не прошел, так как есть конфликт. Другой программист уже внес описание к вашей первой функции, пока вы работали над второй.

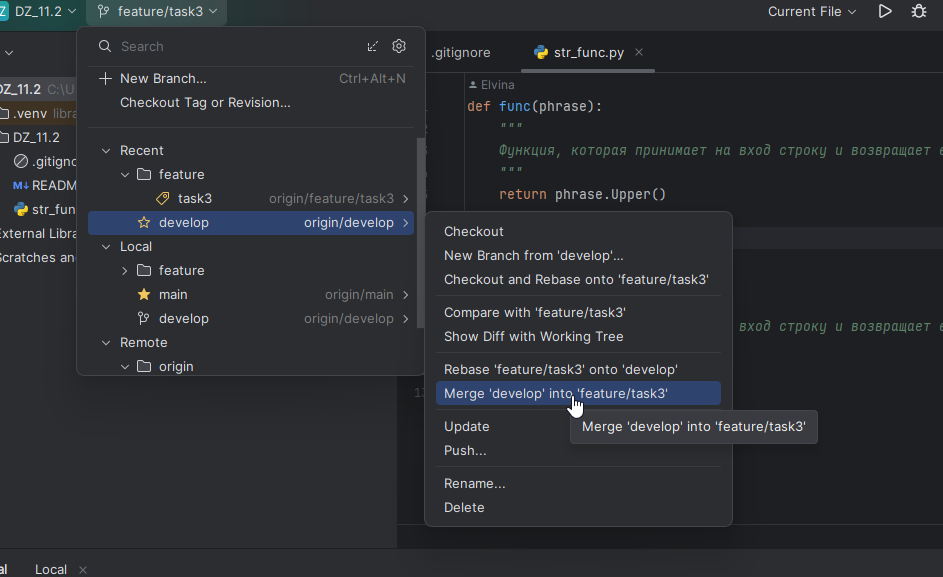
Выполните следующие шаги:

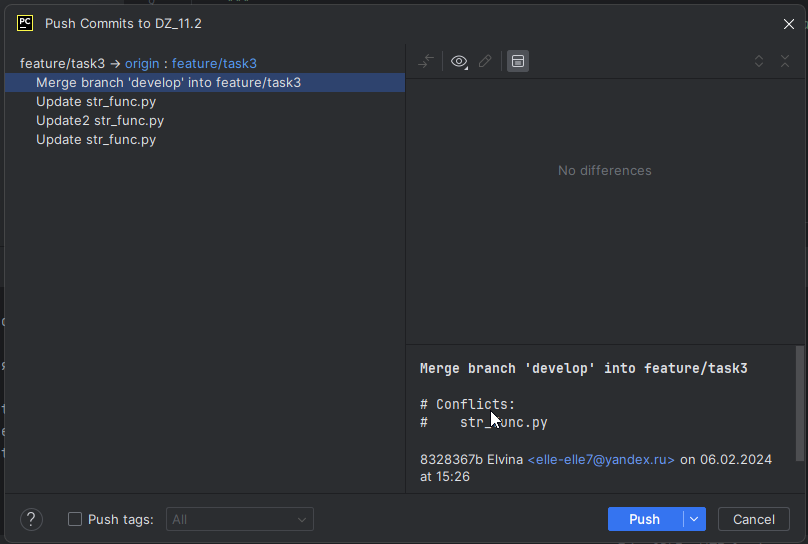
1. Подтяните изменения в ветку develop (у вас должны появиться правки другого программиста).



Появились правки

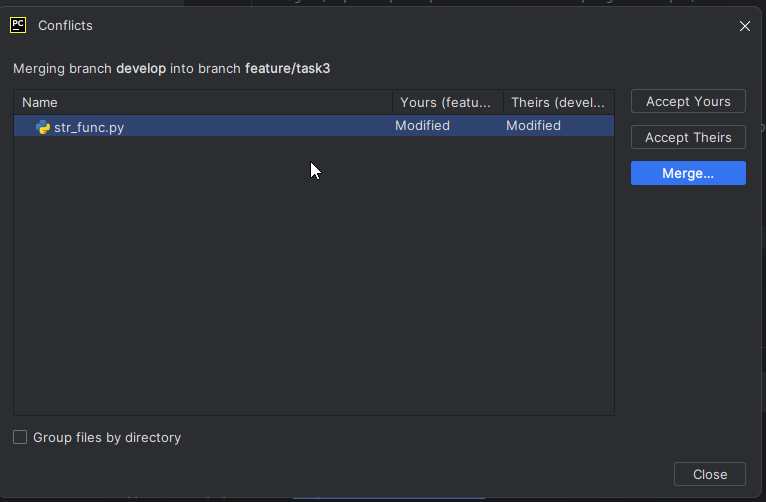


1. Сделайте мердж develop в текущую ветку, где вы ведете разработку (вам предложат порешать конфликт). 

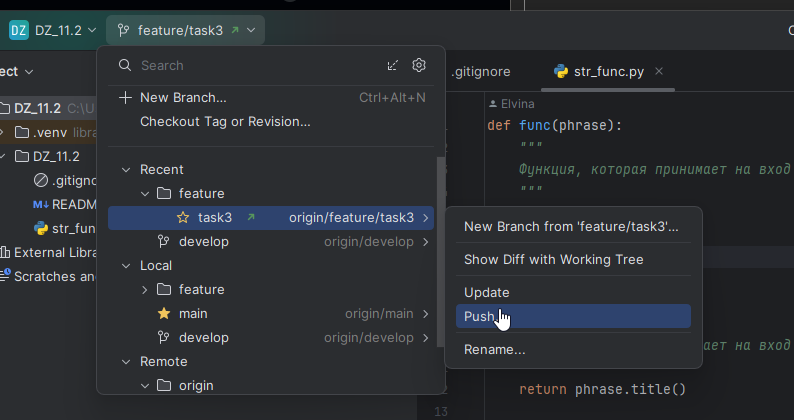


1. Решите конфликт, отклонив описание функции другого программиста.

Жмем Accept yours



1. Сделайте пуш.

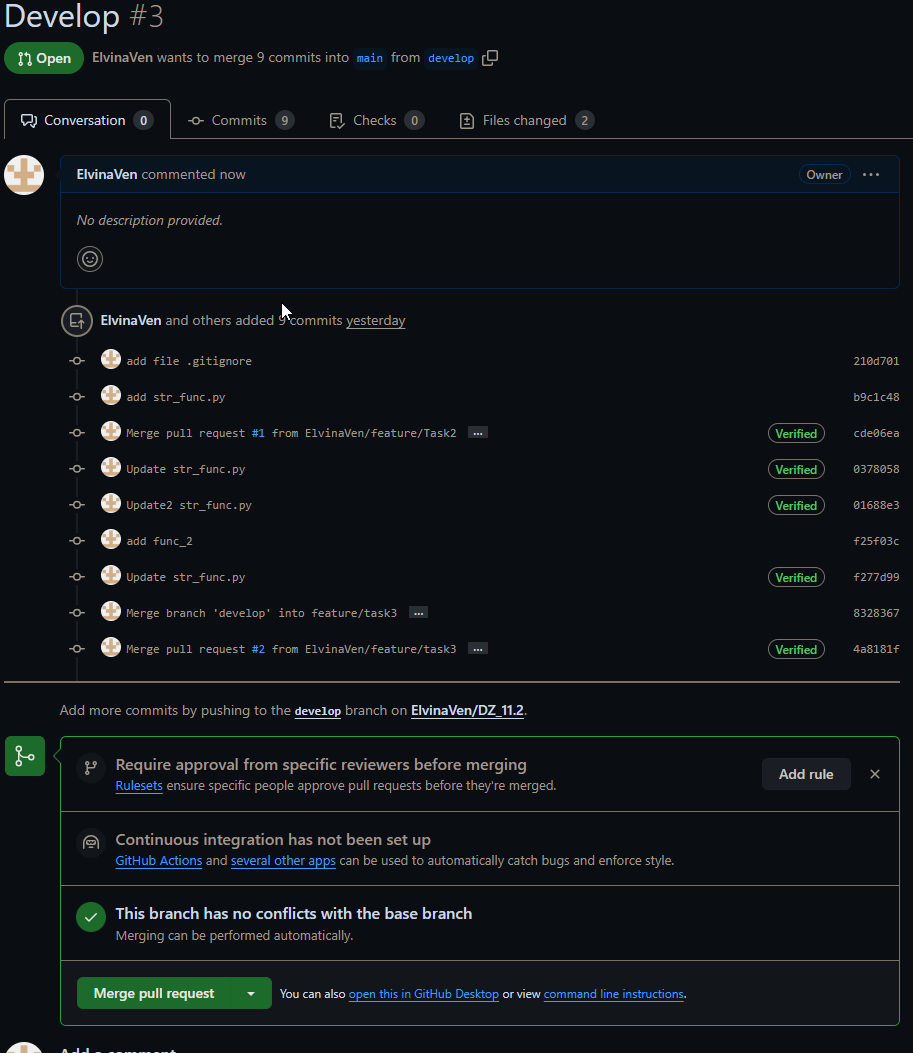


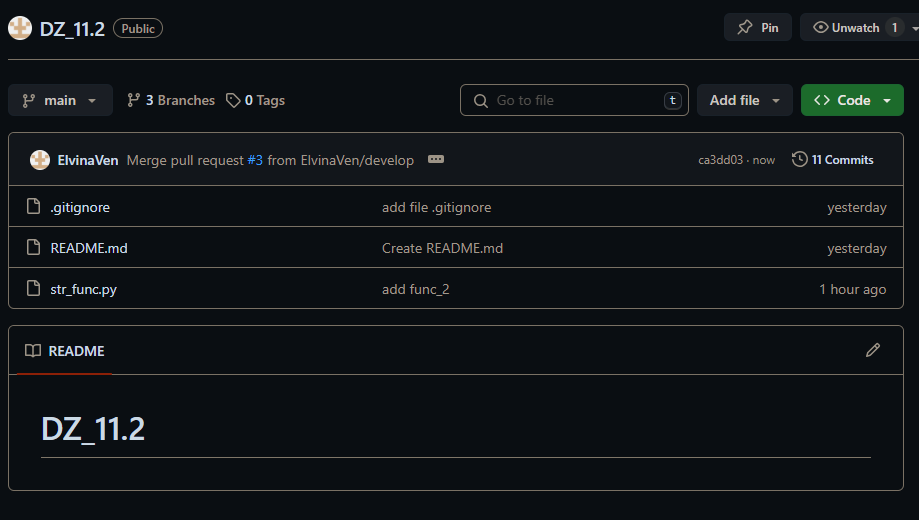
1. За тимлида запрувьте пул-реквест на GitHub.

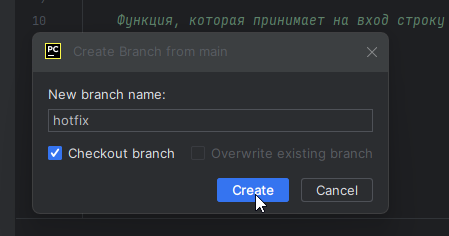
**\* Дополнительное задание**

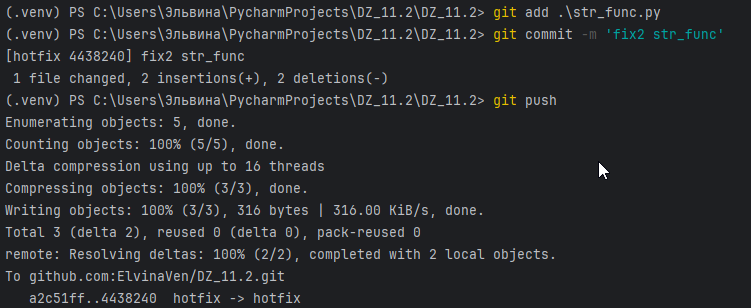
Команды разработчиков постоянно коммитят новый функционал в ветку develop. Но периодически эти наработки надо выкатывать и конечным пользователям. Подготовленные для пользователей версии продуктов (релизы) коммитят в ветку main. Пользователи могут находить какие-то баги в приложении, эти баги быстро чинятся в ветке hotfix.

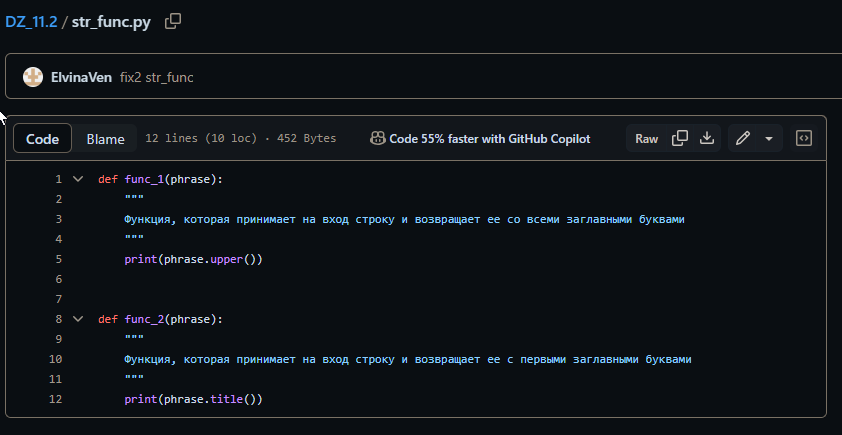
Сделайте мердж ветки develop в main. Из main сделайте checkout в hotfix ветку, сделайте в коде минимальное улучшение, сымитировав фикс бага, и закоммитьте всё обратно в main.

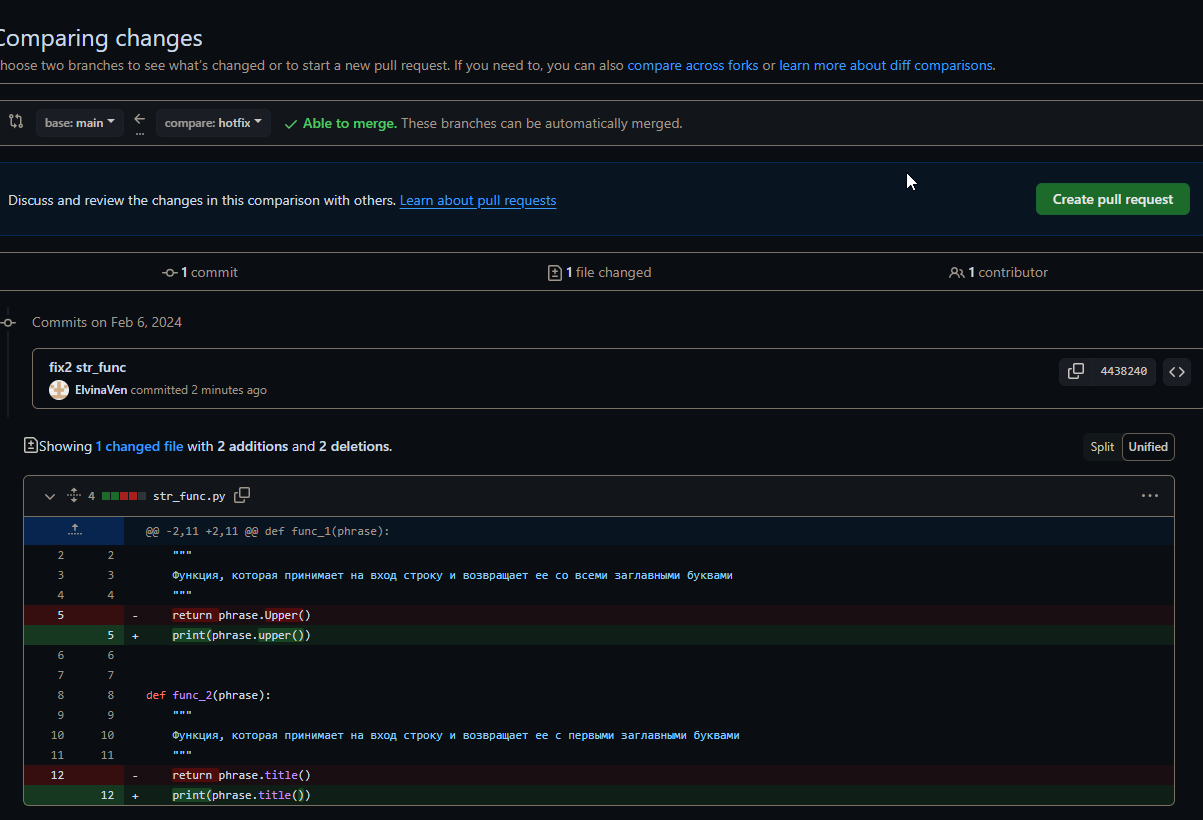












Итог

