

## **Практическое задание**

### **Организация облачного репозитория для хранения файлов проекта**

*Цель работы:* получение практических навыков работы с распределенной системой управления версиями Git и сервисом GitHub.

### **Методические указания**

При работе над программным проектом каждый из команды разработчиков должен иметь доступ ко всем файлам. Поэтому возникает необходимость организации виртуального хранилища программных компонент в системе контроля версий.

Существуют множество систем управления версиями. Условно их можно разделить три их главные группы:

1. в соответствии с расположением репозитория: централизованные и распределенные;
2. в соответствии с методами проверки слияния и передачи кода: блокирующие, использующие слияние до фиксации и выполняющие фиксацию до слияния;
3. системы управления версиями могут выполнять небольшие операции или операции с файлами.

Работы с системами контроля версий рассмотрим на примере распределенной системой управления версиями Git ([git-scm.com](https://git-scm.com)).

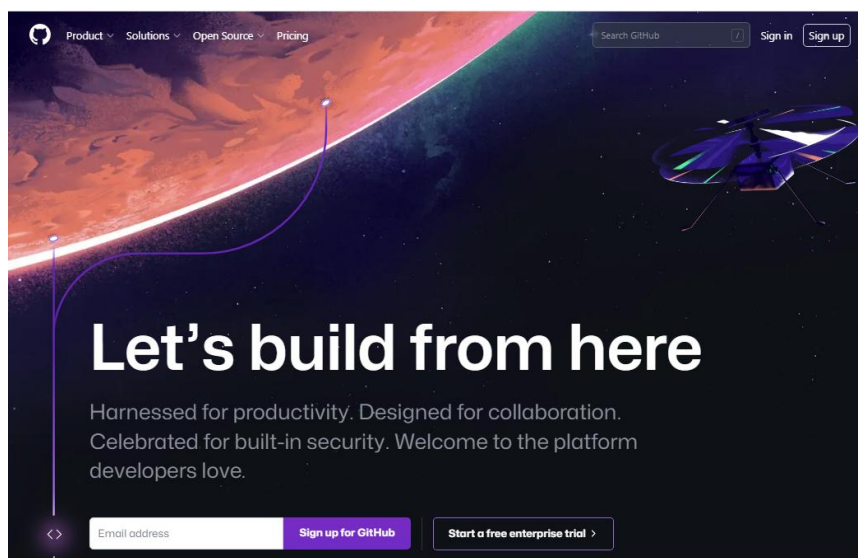
Программные средства данной системы позволяют сохранить изменения в файл или набор файлов в процессе их модификации и при необходимости вернуться к конкретной версии файла. Если над проектом работают несколько человек, то каждому из них обеспечивают доступ для совместной работы над файлом. Каждое внесенное изменение фиксируют, поэтому возникает многоверсионность разрабатываемого программного продукта [32]. Недостатком Git можно считать то, что он имеет интерфейс командной строки. Чтобы

упростить взаимодействие разработчиков с этой системой был создан графический интерфейс GitHub.

GitHub (<https://github.com/>) – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git. Ознакомиться с документацией по работе с GitHub можно на сайте GitHub Help (<https://help.github.com/en>) или ресурсе HTMLAcademy [33].

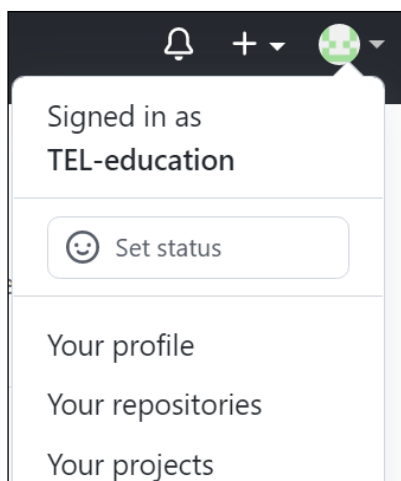
### **Упражнение. Создание аккаунта GitHub.**

1. Зарегистрируйтесь на сервисе GitHub (<https://github.com/>). Для этого перейдите на сайт и нажмите кнопку Sign up (рис.3.17). В открывшемся меню введите параметры учетной записи: имя пользователя; адрес электронной почты; пароль.

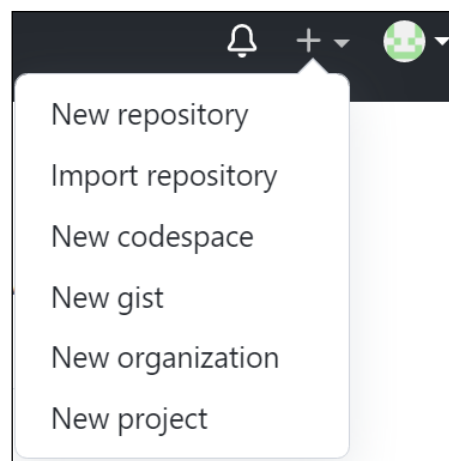


*Рис. 3.17. Регистрация на сервисе GitHub*

2. Полное имя аккаунта посмотрите на вкладке Профиля, например TEL-education.github.io (рис. 3.18.).



*Рис.3.18. Проверка имени аккаунта*



*Рис.3.19. Создание нового репозитория New repository*

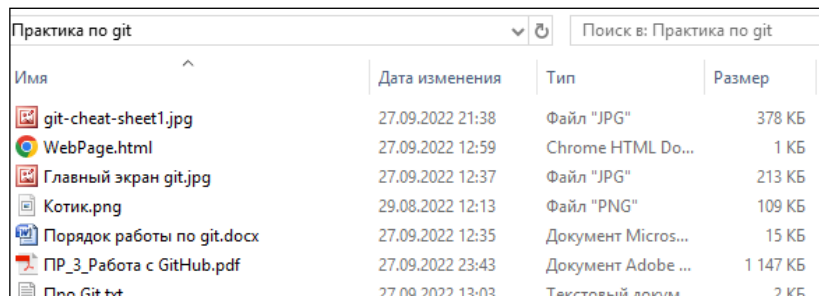
3. После регистрации создайте свой первый репозиторий. Для этого нажмите на знак Плюс в правом верхнем углу окна GitHub. Назовите репозиторий, например *Space* и нажмите кнопку **Create Repository** (рис. 3.19-3.20). Все названия проектов на GitHub пишут на английском языке. Файлы примеров названы на русском языке, чтобы упростить понимание процесса взаимодействия с удаленным репозиторием. В описание репозитория внесите: «Знакомство с GitHub».

Import a repository.'. The 'Owner' field is set to 'TEL-education' and the 'Repository name' field is set to 'Space' with a green checkmark. Below these fields is a suggestion: 'Great repository names are short and memorable. Need inspiration? How about [crispy-octo-goggles?](#)'. The 'Description (optional)' field contains the text 'Знакомство с GitHub'. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option has a subtext: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option has a subtext: 'You choose who can see and commit to this repository.'"/>

*Рис. 3.20. Создание репозитория на сервисе GitHub*

## Упражнение. Копирование файлов проекта в репозиторий на GitHub

1. Подготовьте на персональном компьютере не менее пяти файлов для копирования их в репозиторий (рис.3.21). Файлы могут иметь любой формат, созданы в любых программных средах и имеют разные размеры.



Имя	Дата изменения	Тип	Размер
git-cheat-sheet1.jpg	27.09.2022 21:38	Файл "JPG"	378 КБ
WebPage.html	27.09.2022 12:59	Chrome HTML Do...	1 КБ
Главный экран git.jpg	27.09.2022 12:37	Файл "JPG"	213 КБ
Котик.png	29.08.2022 12:13	Файл "PNG"	109 КБ
Порядок работы по git.docx	27.09.2022 12:35	Документ Micros...	15 КБ
ПР_3_Работа с GitHub.pdf	27.09.2022 23:43	Документ Adobe ...	1 147 КБ
Про Git.txt	27.09.2022 13:03	Текстовый докум...	2 КБ

Рис.3.21. Подготовленные для копирования файлы

2. После создания репозитория будет доступно окно для выбора дальнейших действий пользователя (рис.3.22). Выберите `uploading an existing file` (загрузку внешних файлов).

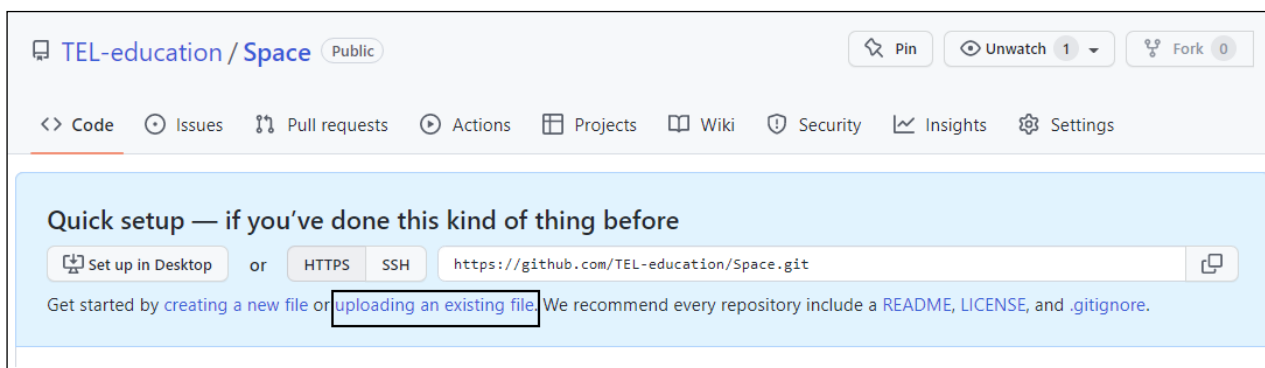
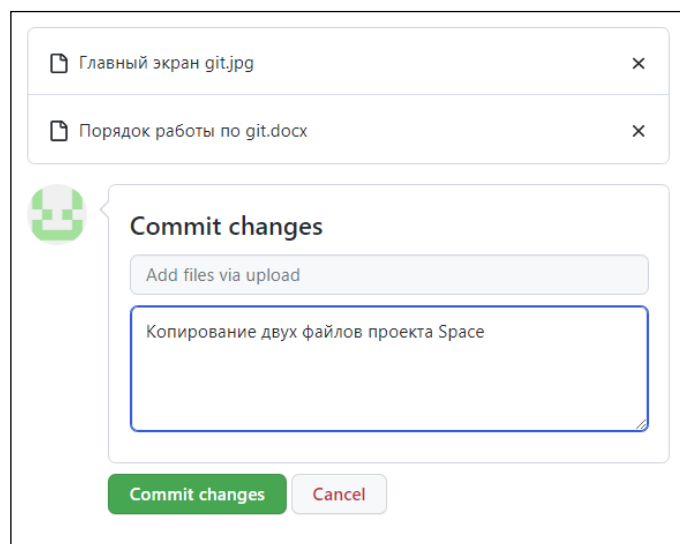


Рис.3.22. Выбор опции `uploading an existing`

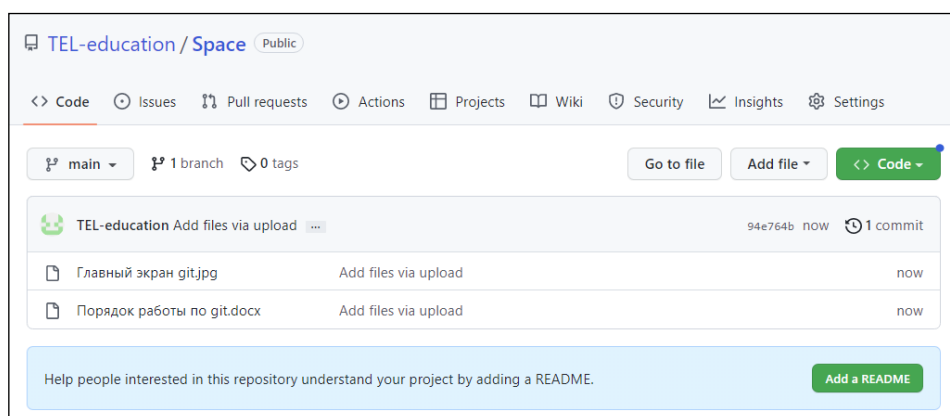
3. Затем перетяните в открывшееся окно два любых файла для копирования в основную ветвь проекта. Основная ветвь проекта по умолчанию названа `main` (рис.3.23). Обратите внимание, на автосгенерированную команду `Add files via upload`, которая появилась в разделе `Commit changes`.

Добавьте описание коммита, объясняющее, почему было сделано конкретное изменение. В рассматриваемом примере: «Копирование двух файлов проекта `Space`». Описания фиксируют историю изменений в коде проекта, чтобы другие участники проекта могли увидеть изменения и понять их назначение. Нажмите на кнопку **Commit changes** и завершите первый коммит.



*Рис.3.23. Добавление файлов проекта в репозиторий*

2. Результат копирования файлов с персонального компьютера в облачное хранилище показан на рис.3.24.



*Рис. 3.24. Ветви проекта Обучение в репозитории TEL-education*

На этом этапе в репозитории **TEL-education** создан проект **Space**, в котором присутствует только одна ветвь проекта 1branch (первая ветвь) под названием main, в которой сохранены два файла.

3. В каждом проектк предусмотрено его описание в файле README.md. Нажмите на кнопку **Add a README** для перехода в область для создания файла.

4. В текстовую область вкладки меню *Edit new file* (рис.3.25) уже внесено описание репозитория: «Знакомство с GitHub». В области *Commit new file* обратите внимание на команду Create README.md. Внесите описание коммита: «Добавление файла README».

5. Выполните слияние с основной веткой проекта main branch, нажав на кнопку **Commit new file**.

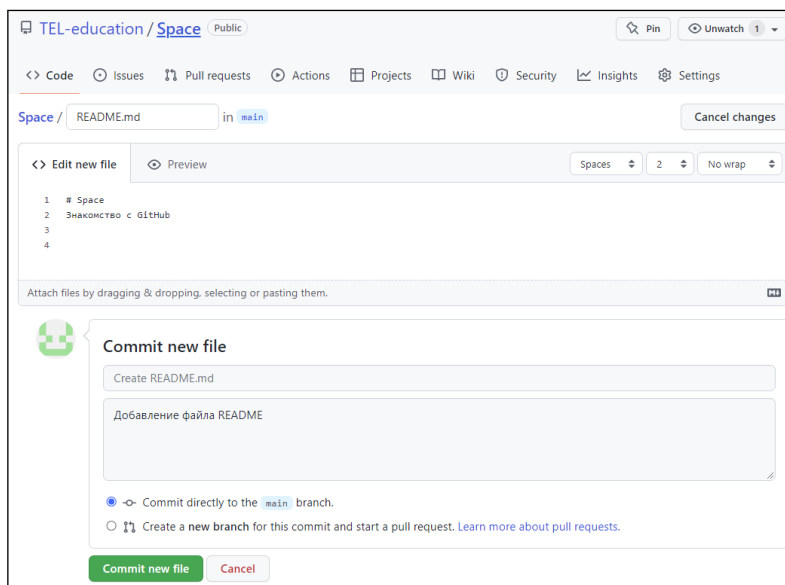


Рис.3.25. Подготовка нового коммита по добавлению файла README.md.

После этого шага в репозитории Space сохранены три файла (рис.3.26).

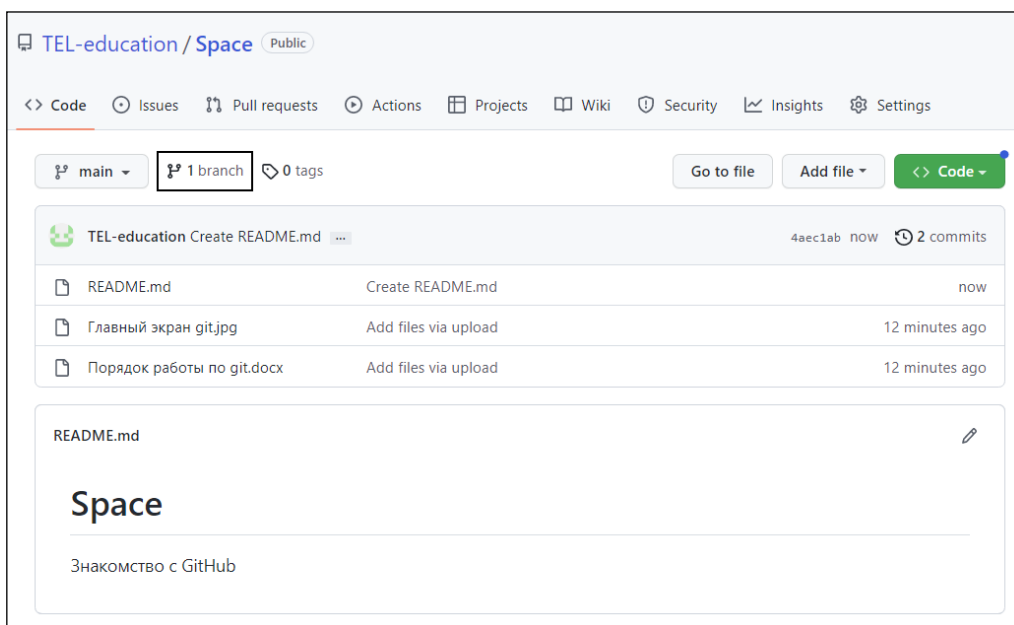


Рис. 3.26. Количество файлов в ветке main после первого коммита

### Упражнение. Добавление второй ветви проекта Space

1. Кликните по пиктограмме 1 branch (рис. 3.26).
2. Нажмите на кнопку **New branch** для добавления новой ветви проекта (3.27).

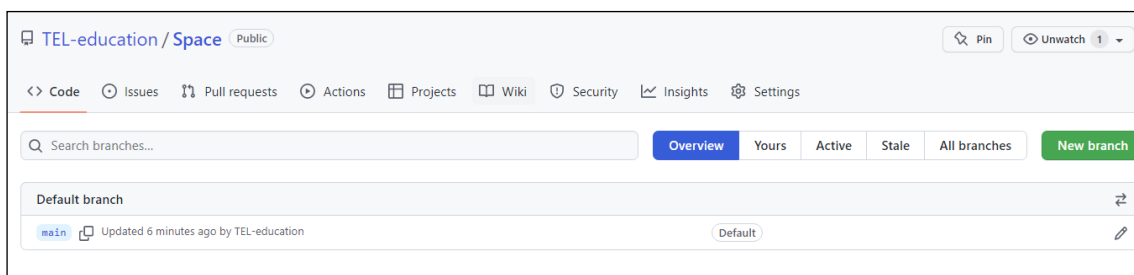


Рис. 3.27. Начало создания новой ветви в репозитории Space

3. Дайте новой ветви понятное и лаконичное название, например ВтораяВетвь. Источником данных этой ветви будут данные ветви main – три файла проекта Space. Нажмите **Create branch** (рис.3.28). После создания новой ветви в репозиторий включает две ветви (рис. 3.29). На данном шаге в каждой ветви сохранения по три одинаковых файла.

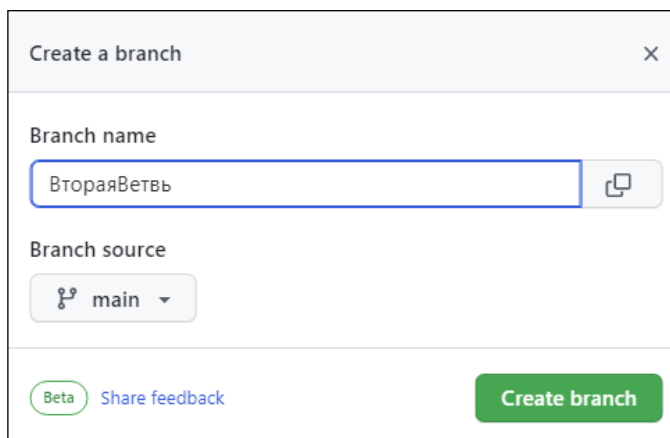


Рис. 3.28. Создание второй ветви в репозитории Space

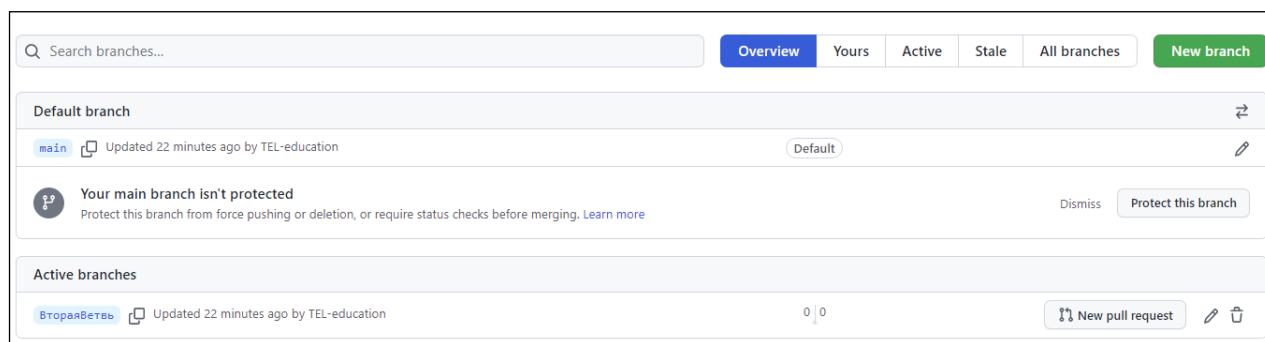


Рис. 3.29. Две ветви репозитория Space

При создании второй ветви GitHub потребует уточнений в политике безопасного слияния ветвей проекта. Нажмите **Protect this branch**. Будет произведен перход на новую страницу с политиками GitHub. В учебном проекте

не активуйте чек-боксы, оставьте все настройки без изменений и примите соглашение о политике управления интеграцией.

4. Нажмите на название репозитория Space (рис.3.26). Затем выберите из раскрывающегося меню название ВтораяВетка (рис.3.27).

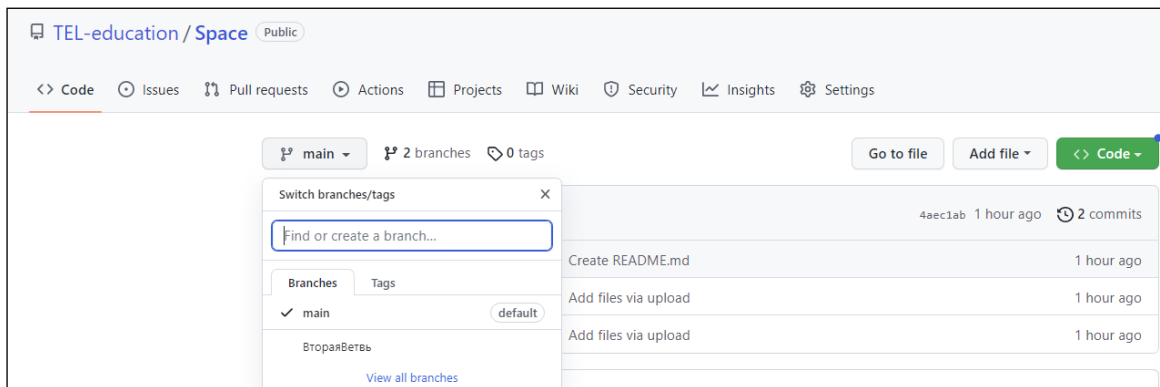


Рис. 3.27. Переход к данным, расположенным во второй ветви

5. Добавьте еще два файла во вторую ветку репозитория. Для этого нажмите на **Add files** и выберите из раскрывающегося списка **Upload files** (рис.3.28).

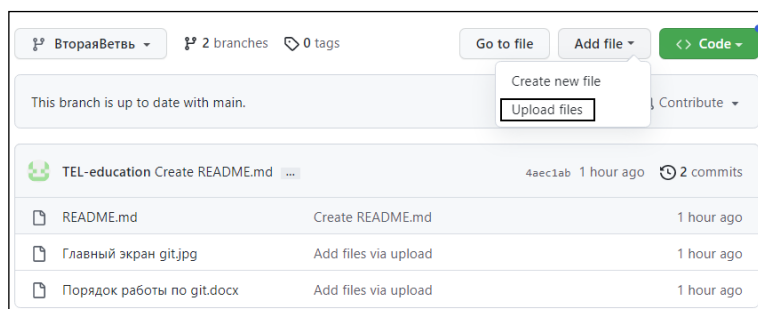


Рис.3.28. Выбор варианта загрузки файлов

6. Скопируйте два файла из папки, подготовленной на персональном компьютере. Добавьте описание коммита и нажмите **Commit change** (рис. 3.29).

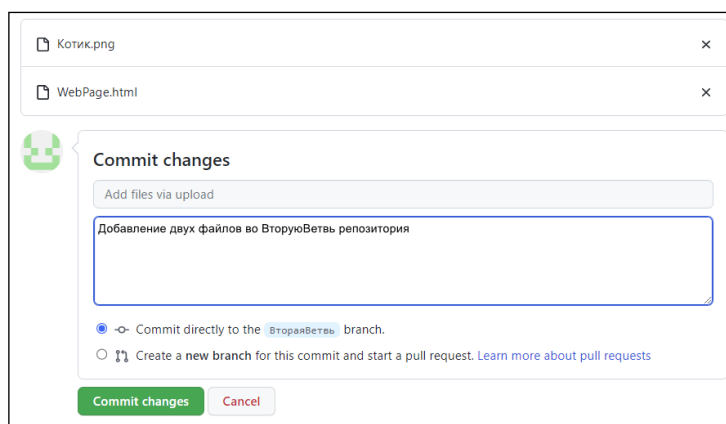




Рис. 3.29. Добавление файлов во вторую ветку

7. Нажмите на название репозитория Space и посмотрите на результат изменения количества файлов (рис.3.30).

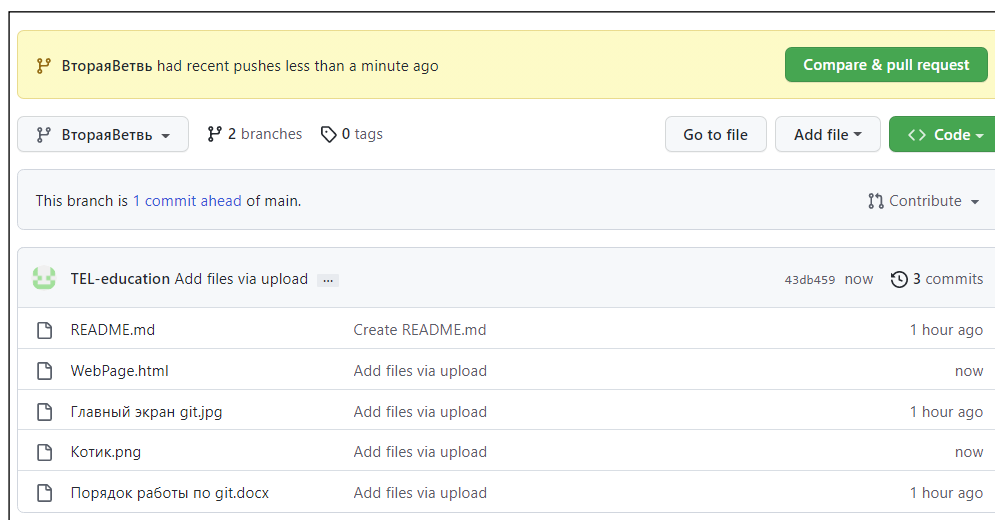


Рис.3.30 Просмотр новых данных во второй ветви

Обратите внимание, что количество ветвей проекта увеличилось на единицу и вместо 1 branches записано 2 branches.

### Упражнение. Добавление третьей ветви проекта Space

1. Нажмите на пиктограмму 2 branches, перейдите в область создание новой ветви (повторите шаги 2 и 3 предыдущего упражнения). Назовите третью ветку репозитория ТретьяВетвь (рис.3.31). Укажите, что источником данных для нее является также ветвь main.

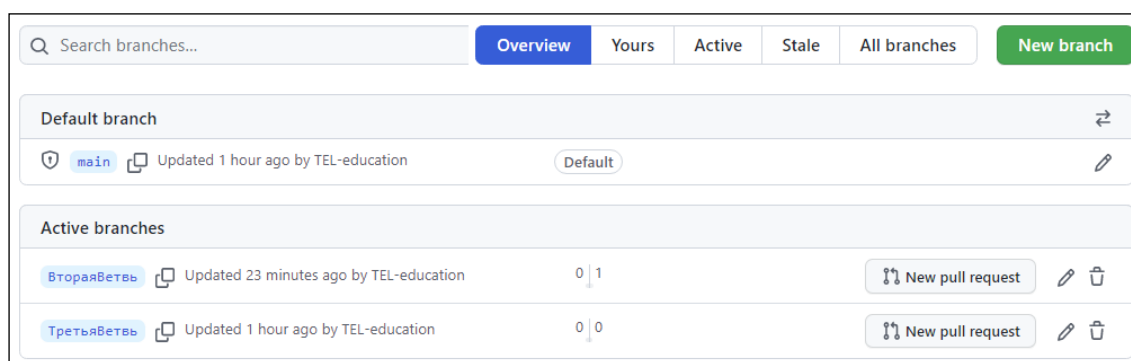


Рис. 3.31. Три ветви проекта

2. Перейдите в третью ветку и создайте в ней текстовый файл. Для этого нажмите на **Add files** и в раскрывшемся списке выберите **Create New file** (рис. 3.28). Добавьте следующий текст: «GitHub – крупнейший веб-сервис для

хостинга IT-проектов и их совместной разработки. Web-сервис основан на системе контроля версий Git. Ознакомиться с документацией по работе с GitHub можно на сайте GitHub Help».

Назовите файл, например Назначение СКВ.txt. Заполните комментарий и нажмите на кнопку **Commit new files** (рис.3.32).

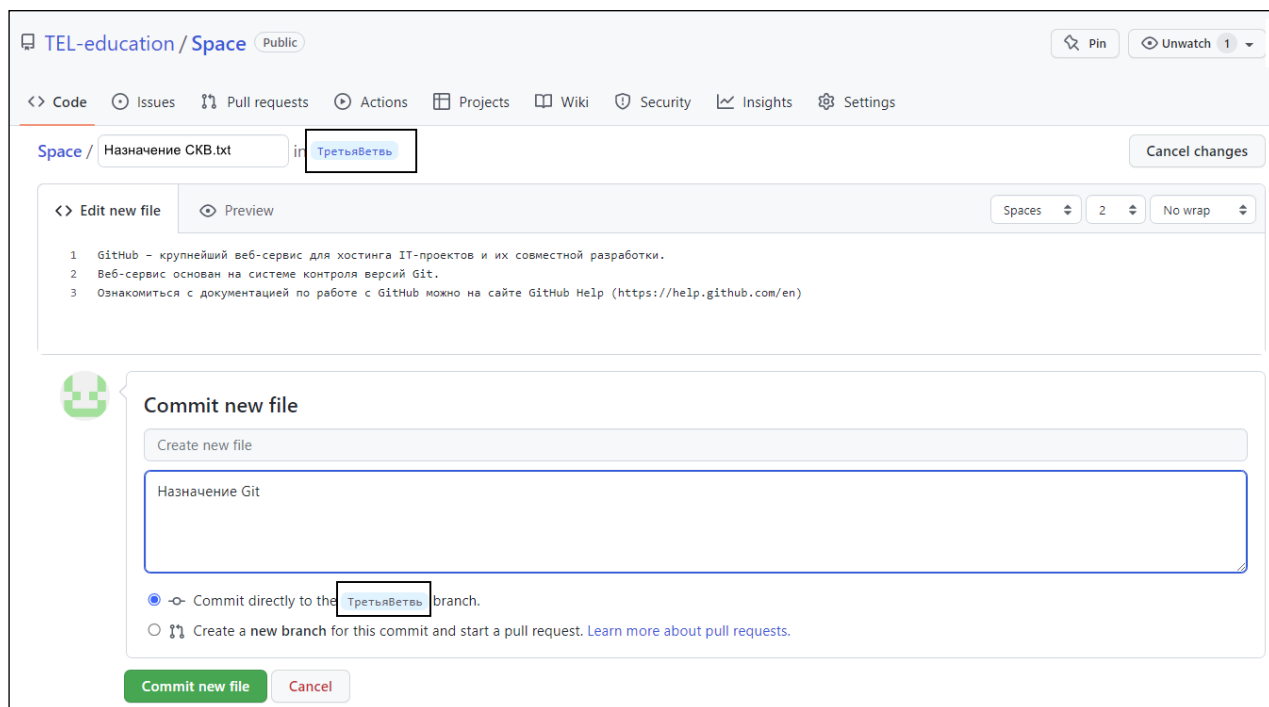


Рис. 3.32. Добавление текстового файла в третью ветку репозитория

Обратите внимание, что добавление файла происходит в ветку под названием ТретьяВетвь.

3. В репозитории созданы три ветви: main, ВтораяВетвь, ТретьяВетвь (рис.3.33).

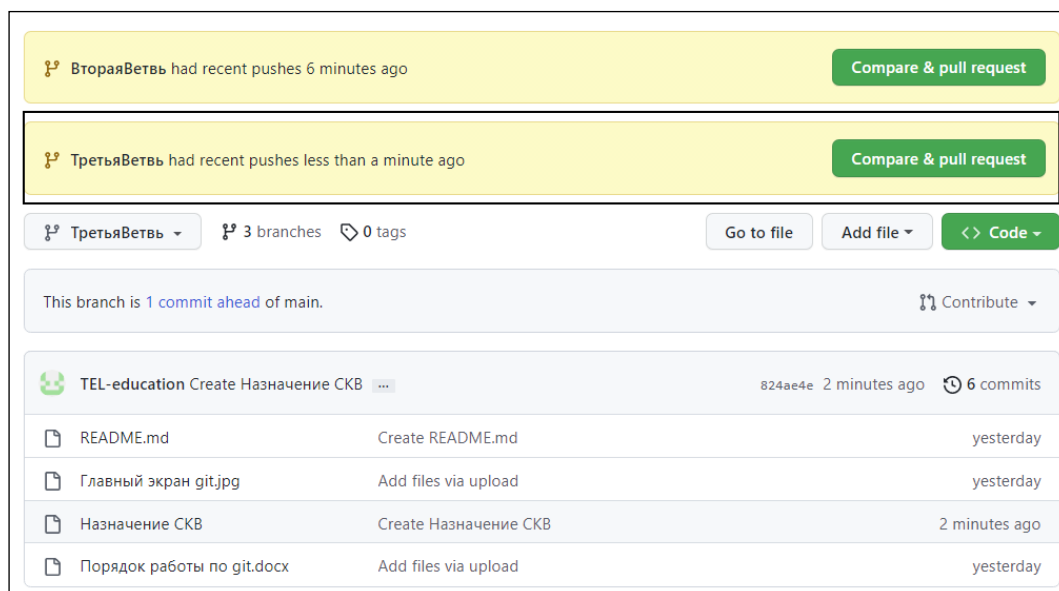


Рис. 3.33. Три ветви проекта

В ветке `main` сохранены исходные файлы проекта, в двух других – альтернативы варианты развития проекта, в которых к исходным файлам добавлены другие. На следующем этапе разработчики могут выбрать, какой из двух вариантов выбрать и, выполнив запрос на слияние, сделать основным.

### Упражнение. Объединение ветвей проекта Space

1. Проведите слияние третьей ветки проекта с основной веткой. Нажмите на кнопку **Compare & pull request** (рис. 3.33). Pull request (пул-реквест) – это запрос на интеграцию изменений в выбранную ветку проекта.

2. Проверьте правильность интеграционного потока: из ТретьейВетви в основную `main` (рис.3.34). Будет проведено автоматическое сравнение данных в двух ветвях и добавление файлов, которые присутствуют только в ТретьейВетви, в основную ветвь.

3. Нажмите **Create Pull request**.

4. Подтвердите запрос на интеграцию нажатием на кнопку **Merge Pull request** (рис.3.35).

5. Кнопка **Merge Pull request** изменила название **Confirm merge** (рис.3.36).

6. Результат успешной интеграции показна на рис.3.37.

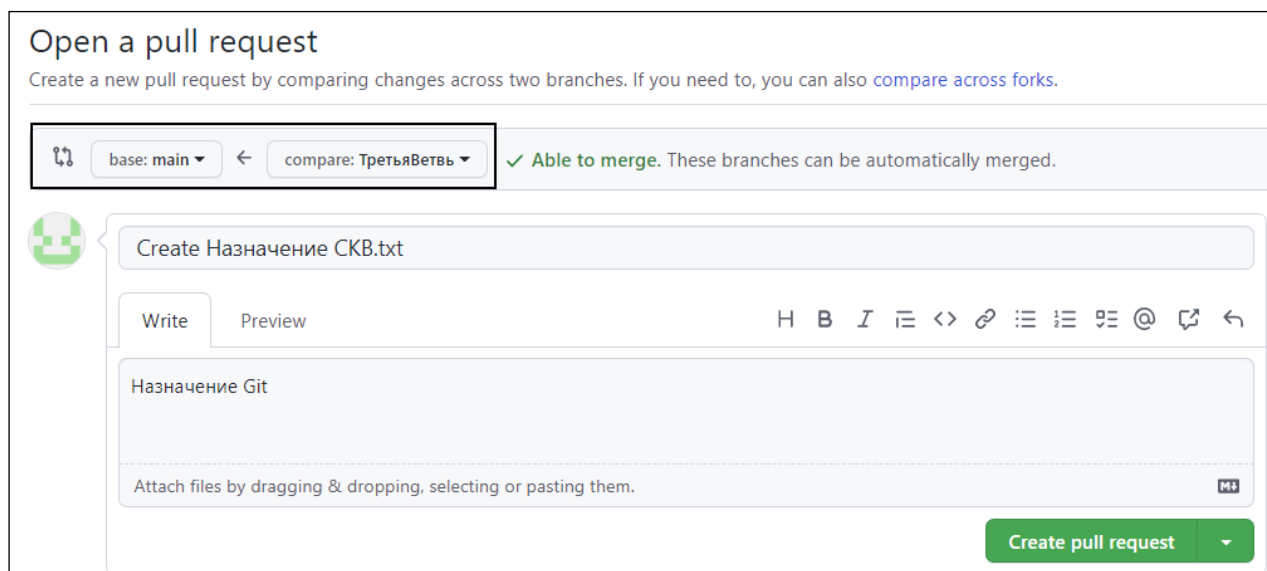


Рис. 3.34. Интеграция данных третьей ветви и основной

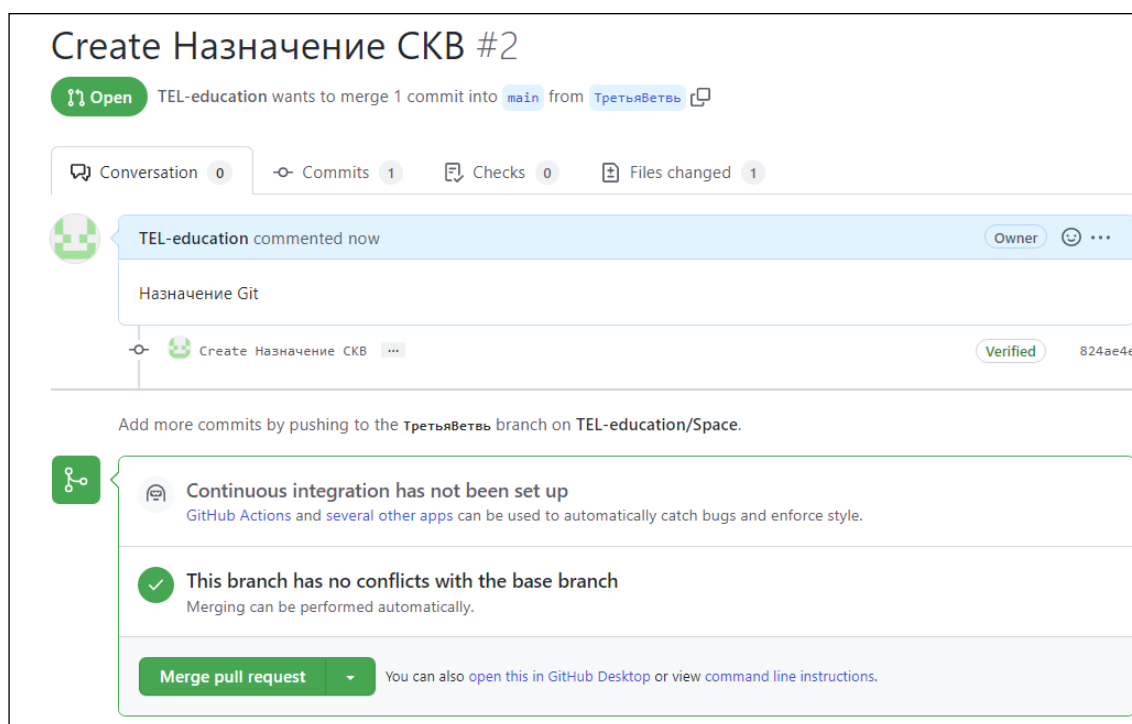


Рис. 3.35. Подтверждение запроса на интеграцию

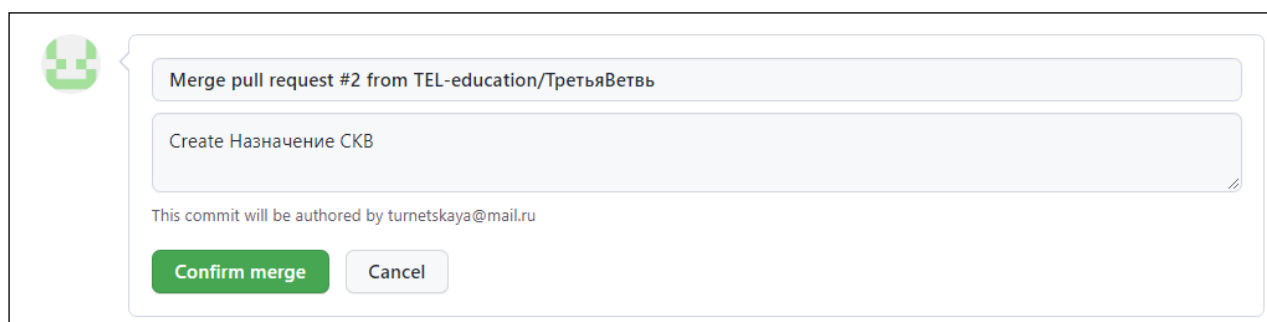


Рис. 3.36. Принятие подтверждения нажатием на кнопку **Confirm merge**

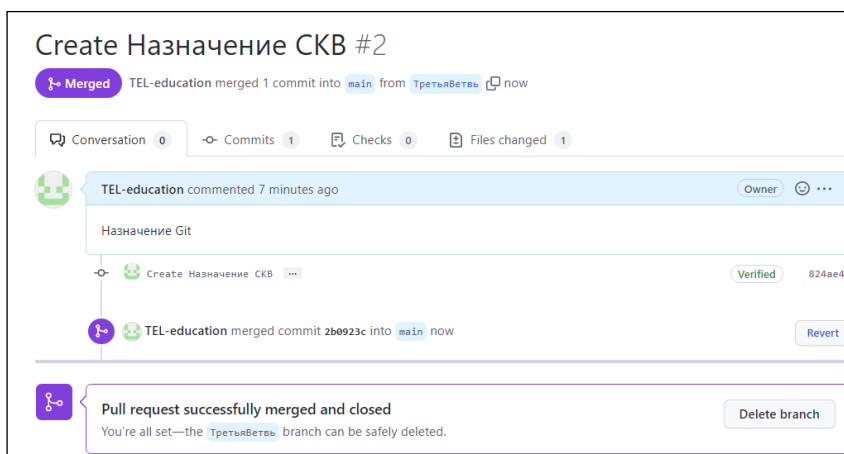


Рис.3.37. Сообщение об успешной интеграции данных в основную ветвь

7. Перейдите в режим просмотра ветвей проекта: **Нажмите на Space>3 branches** (рис. 3.38). В комментариях ТретьейВетви появилась информация о выполненной интеграции.

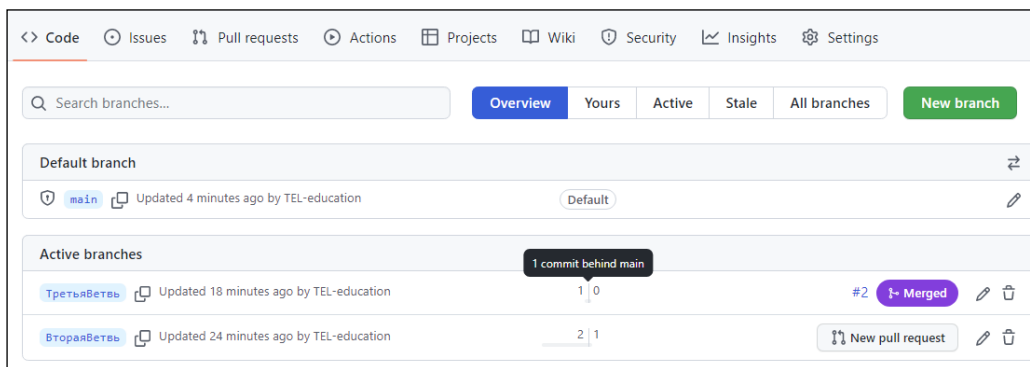


Рис.3.38. Информация о действиях по ветвям репозитория

### Порядок выполнения

1. Зарегистрироваться на сервере GitHub.
2. Создать не менее трех ветвей проекта, в каждой из которых сохраните файлы разных форматов: код программы, выполненный на любом языке программирования, или текстовый файл, например отчеты по лабораторным работам, графические материалы и т.д.
3. Каждое изменение (новую ветвь, дополнение репозитория) фиксировать в системе контроле версий.
4. Форкните любой программный проект, который хранится на GitHub, например библиотеку pandas, в свой удаленный репозиторий.

5. При выполнении заданий следующих практических заданий выкладывайте файлы разрабатываемого проекта в репозиторий. В каждом отчете предоставляйте ссылку на данный удаленный репозиторий.

При выполнении работы рекомендуется использовать приложение на GitHub Desktop для внесения изменений в исходные файлы, хранящиеся на сервере распределенной системы управления версиями Git.

### Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Назначения GitHub.
4. Прогресс по реализации проекта в Kaiten.
5. Созданный аккаунт на GitHub с календарем активностей (рис.3.39-3.40).

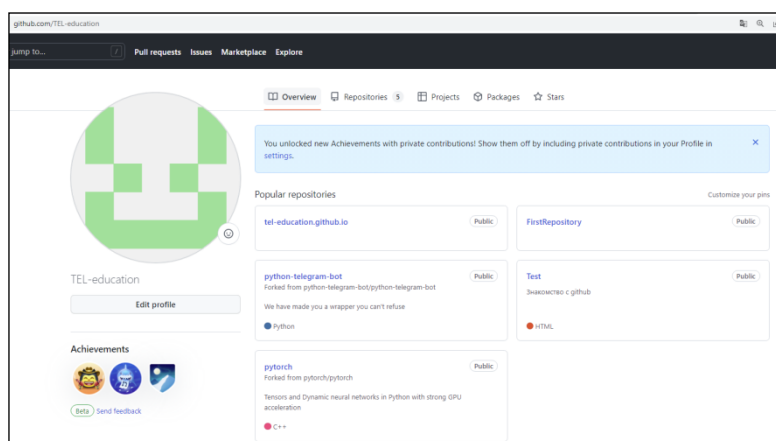


Рис. 3.39. Пример аккаунта

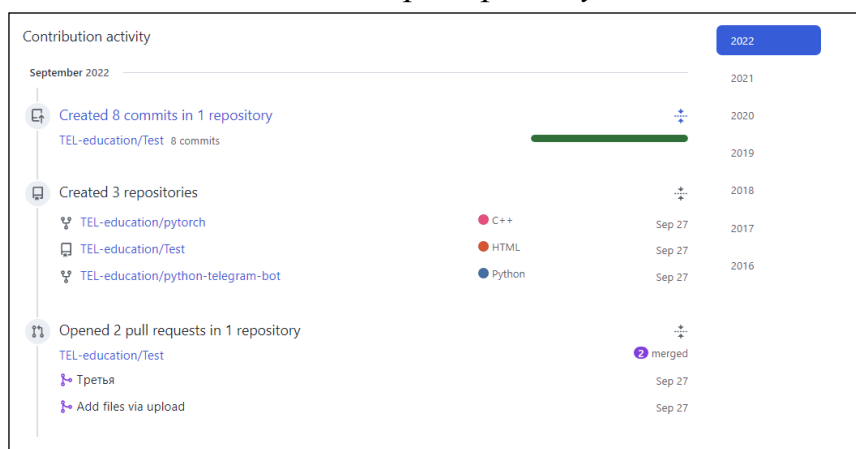


Рис. 3.40. Активность аккаунта

6. Скриншоты двух ветвей проекта на удаленном сервере GitHub (рис.3.41-3.42).

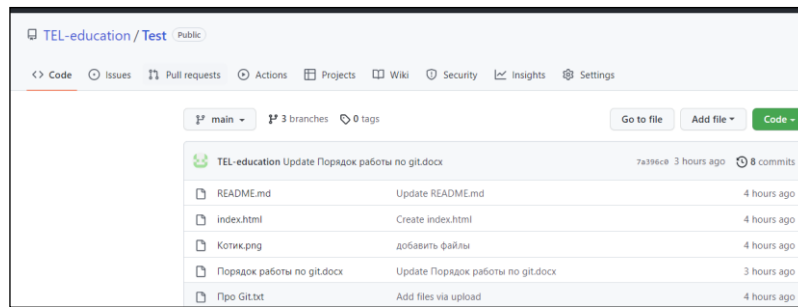


Рис. 3.41. Ветвь main

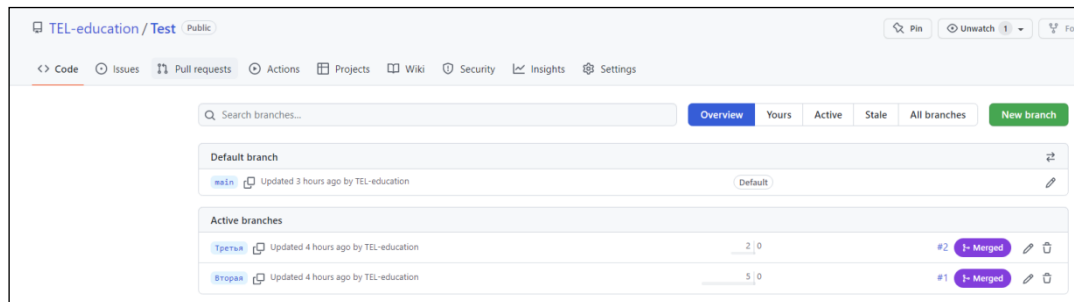


Рис. 3.43. Ветви проекта

7. Скриншот форка выбранного проекта с полезными библиотеками (рис.3.43).

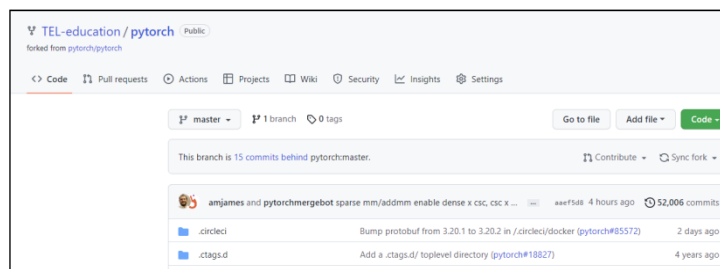


Рис. 3.43. Пример скриншота форка библиотеки pytorch

8. Ссылка на созданный удаленный репозиторий на GitHub.

9. Вывод в формате эссе, в котором требуется объяснить назначение работы, что нового узнали при работе над проектом (или все знали), с какими проблемами столкнулись при выполнении работы и способы их преодоления.

10.Список использованных источников.

Обратите внимание, что в репозитории в последствии должны быть сохранены файлы проекта, который вы разрабатываете в рамках изучаемой дисциплины.