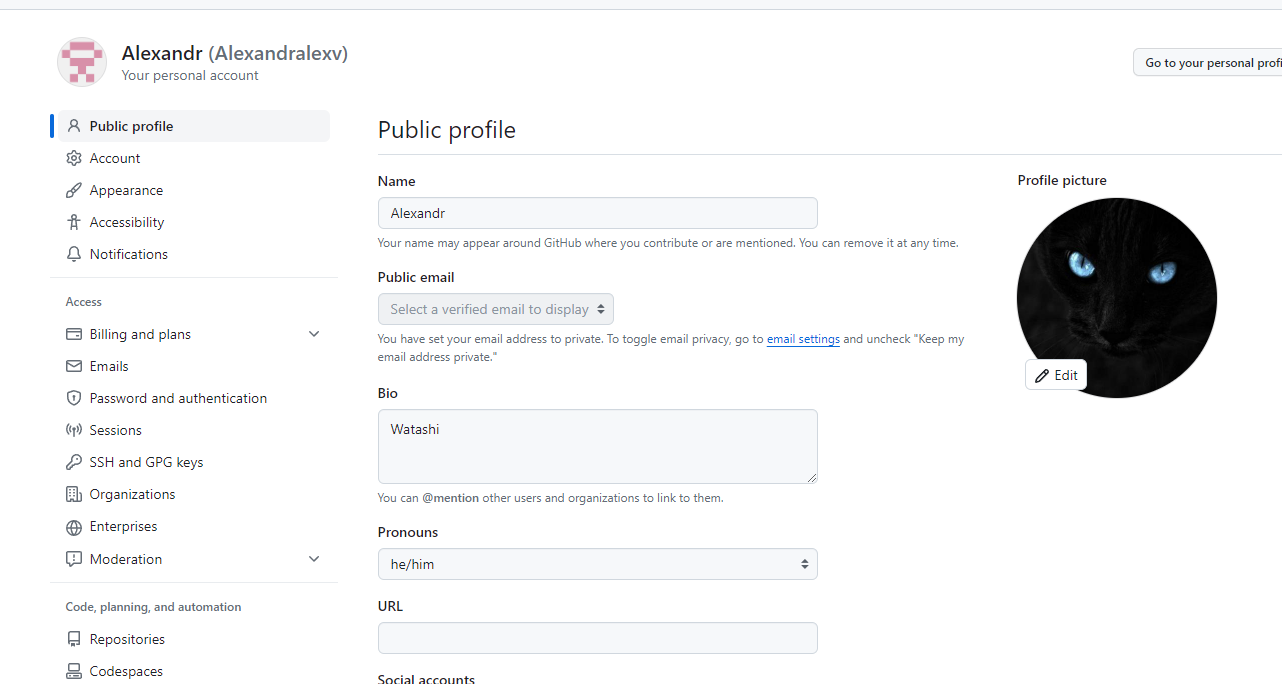
**3. Работа с удаленными репозиториями и GitHub**

**Цель работы**

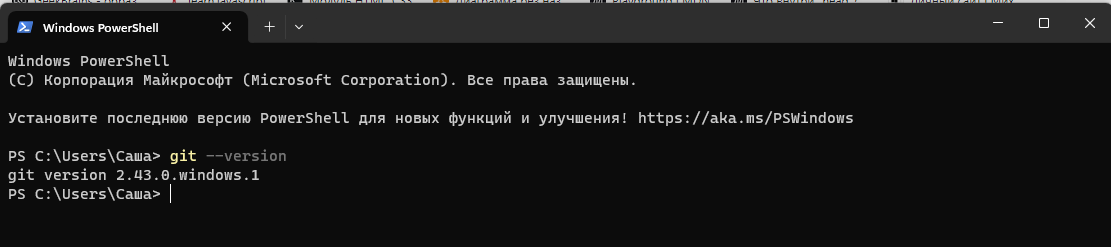
Освоить основные навыки работы с облачными и распределенными системами контроля версий, получить навыки работы с инструментальными средствами, обеспечивающими командную работу над разработкой ПО.

**Задания для выполнения**

1. Зарегистрироваться на сайте github.com

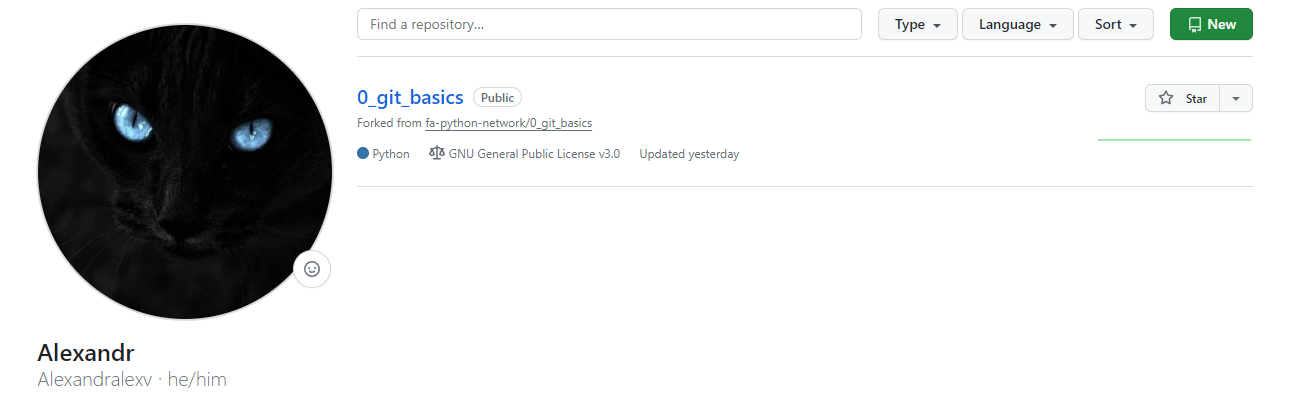


1. Установить на компьютере программу Git



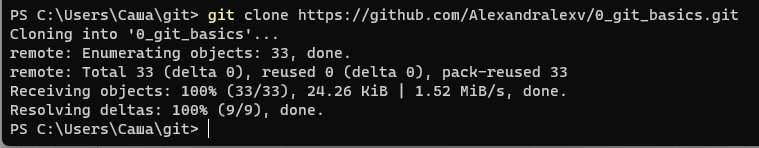
1. Форкнуть данный репозиторий в свой аккаунт





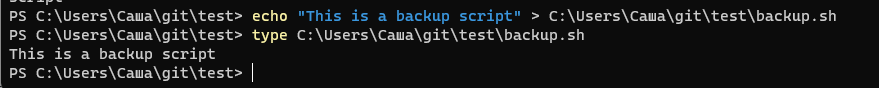
1. Склонировать созданный удаленный репозиторий в директорию ~/git/test



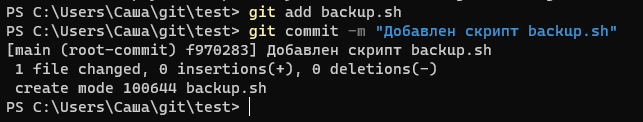


1. На локальной машине пишем скрипт ~/git/test/backup.sh, с произвольным содержанием

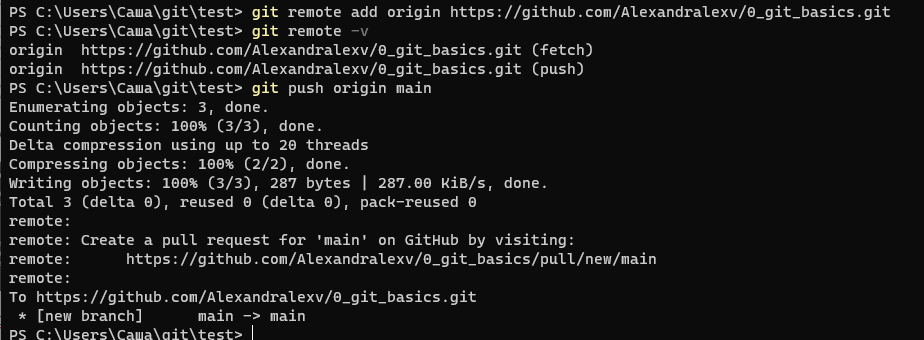




1. Фиксируем скрипт в репозитории (делаем коммит)

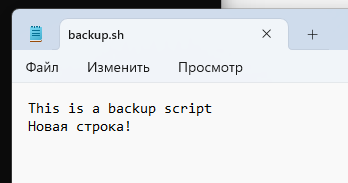


1. Обновляем удаленный репозиторий репозиторий (делаем пуш)

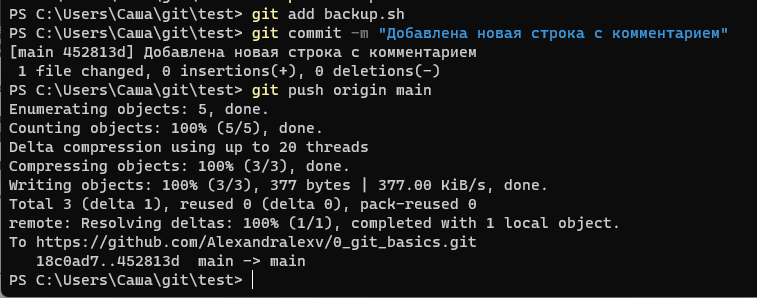


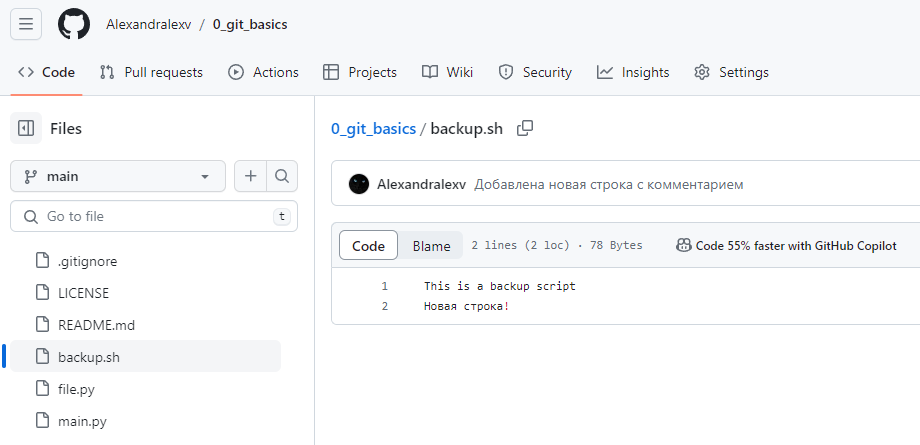
1. Через текстовый редактор добавить любую новую строку с комментарием



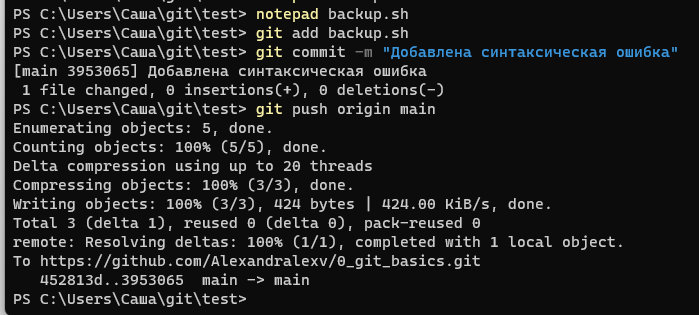


1. Сделать коммит

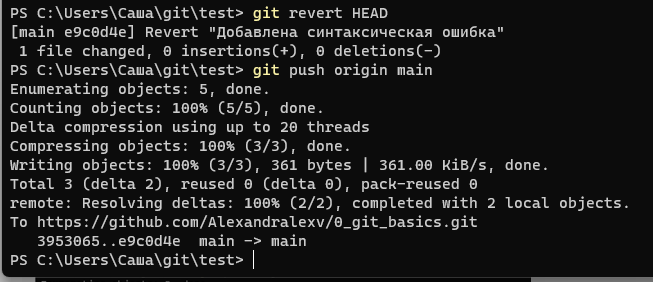


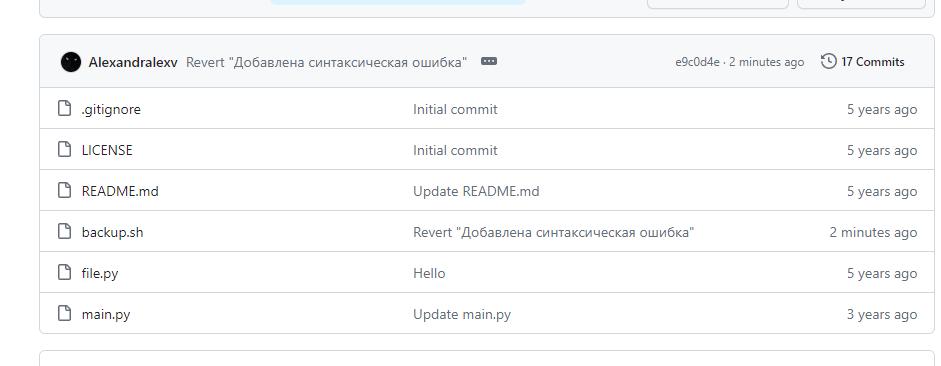


1. Вности синтаксическую ошибку в скрипт
2. Сделать коммит ошибочного скрипта

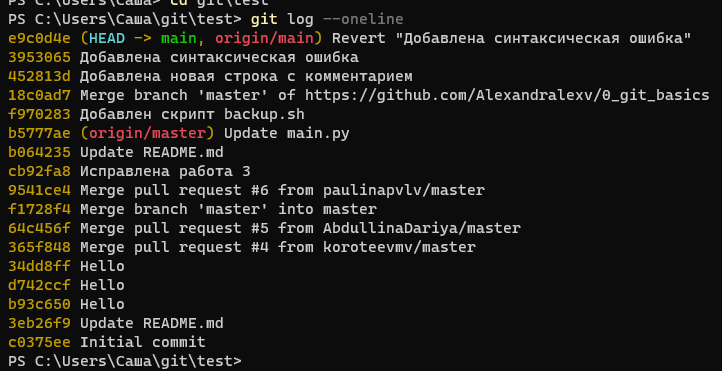


1. Откатываем до последней рабочей версии

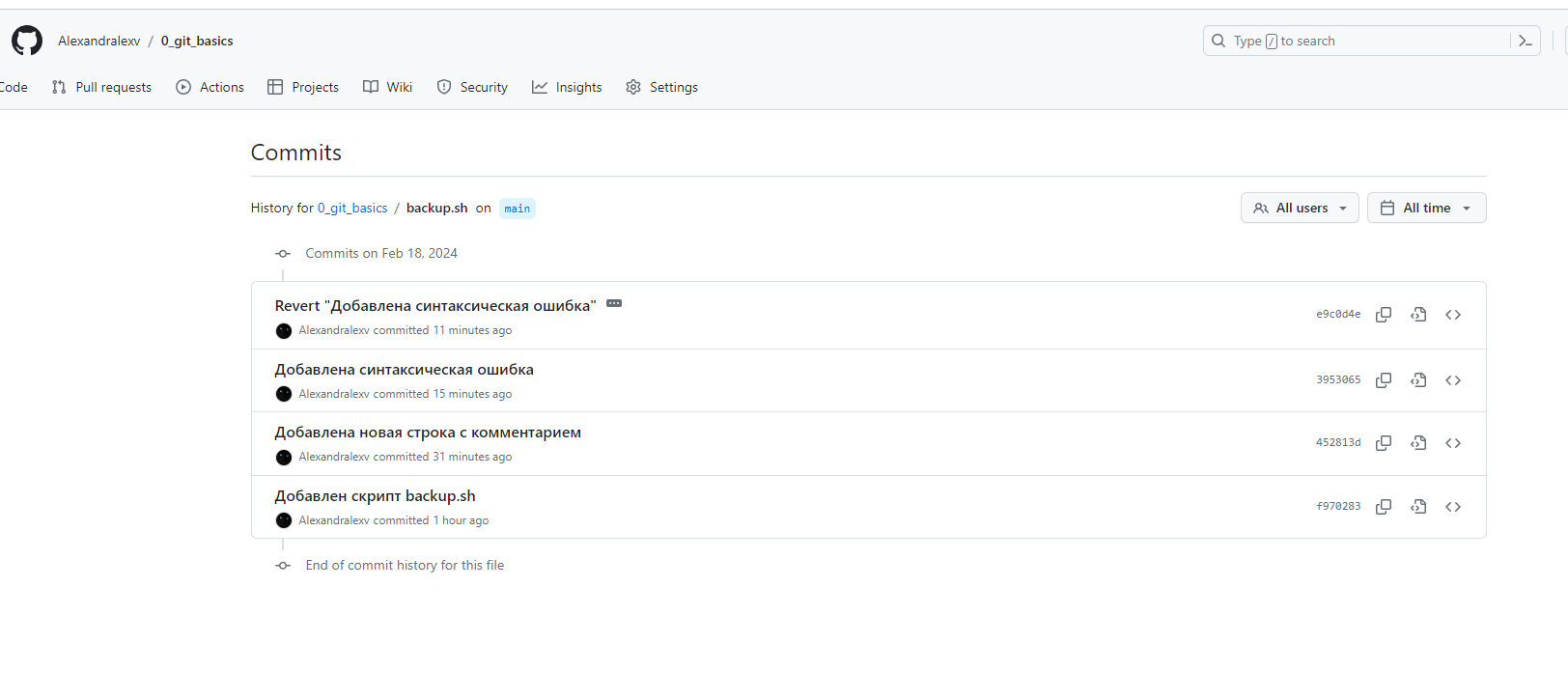




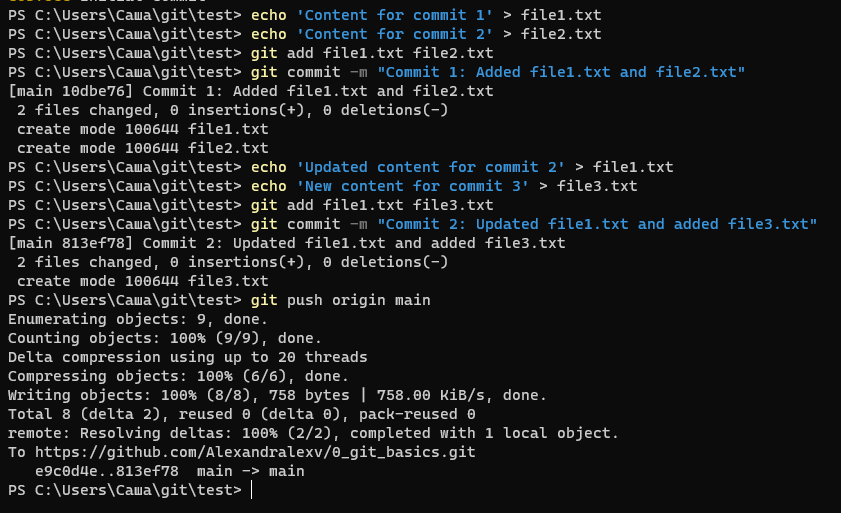
1. Просмотреть историю коммитов



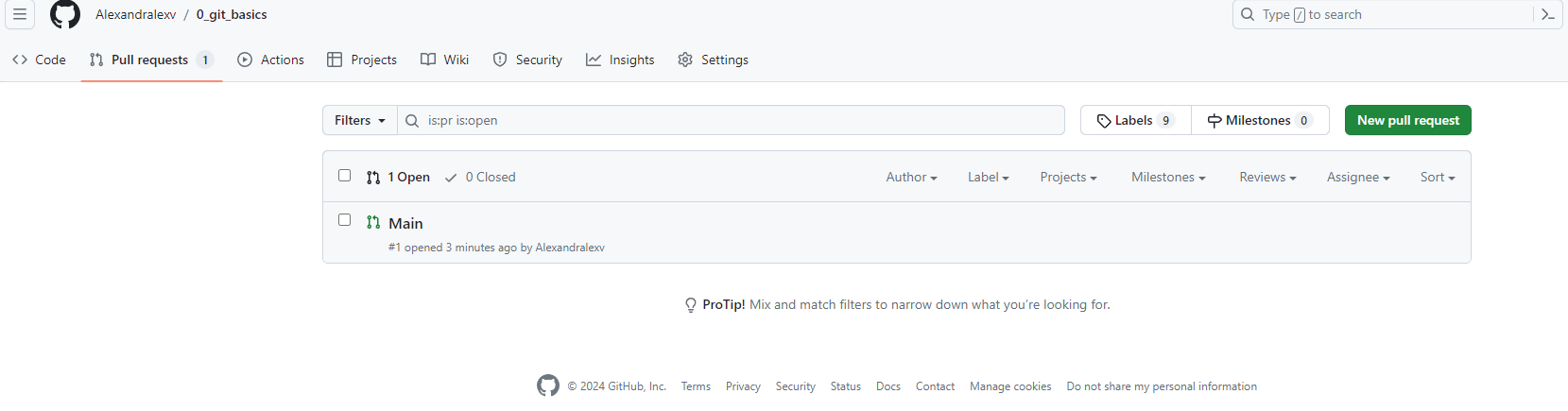
Или на гитхабе: backup.sh -> History

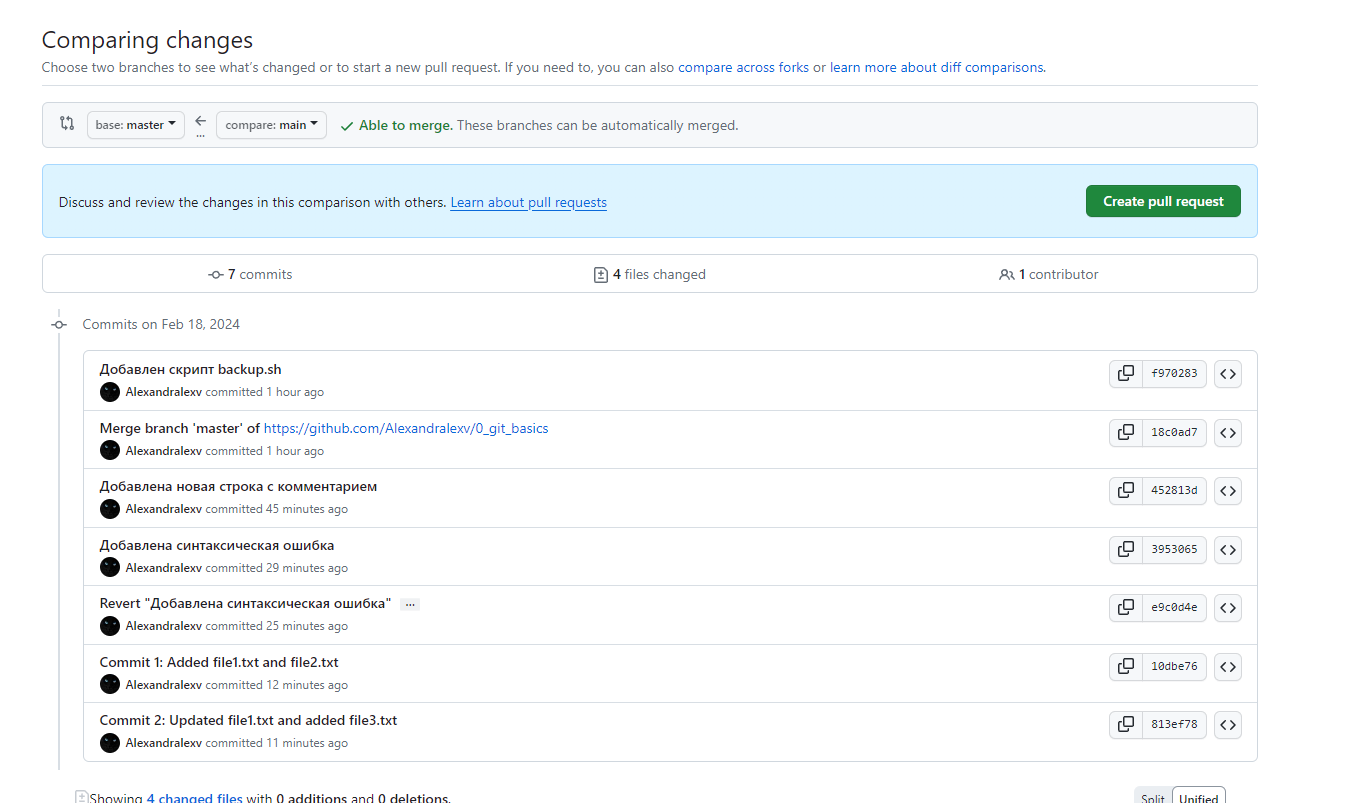


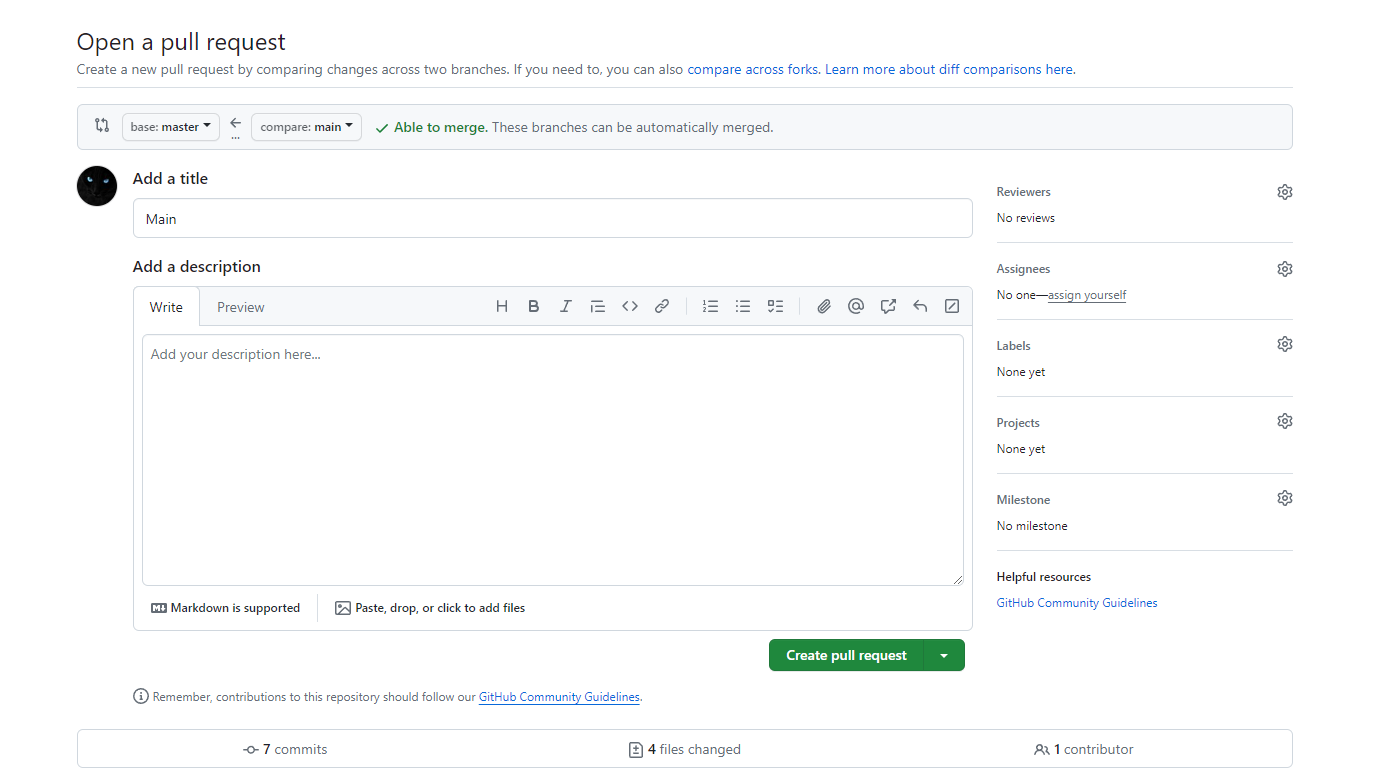
1. Добавить несколько коммитов произвольного содержимого

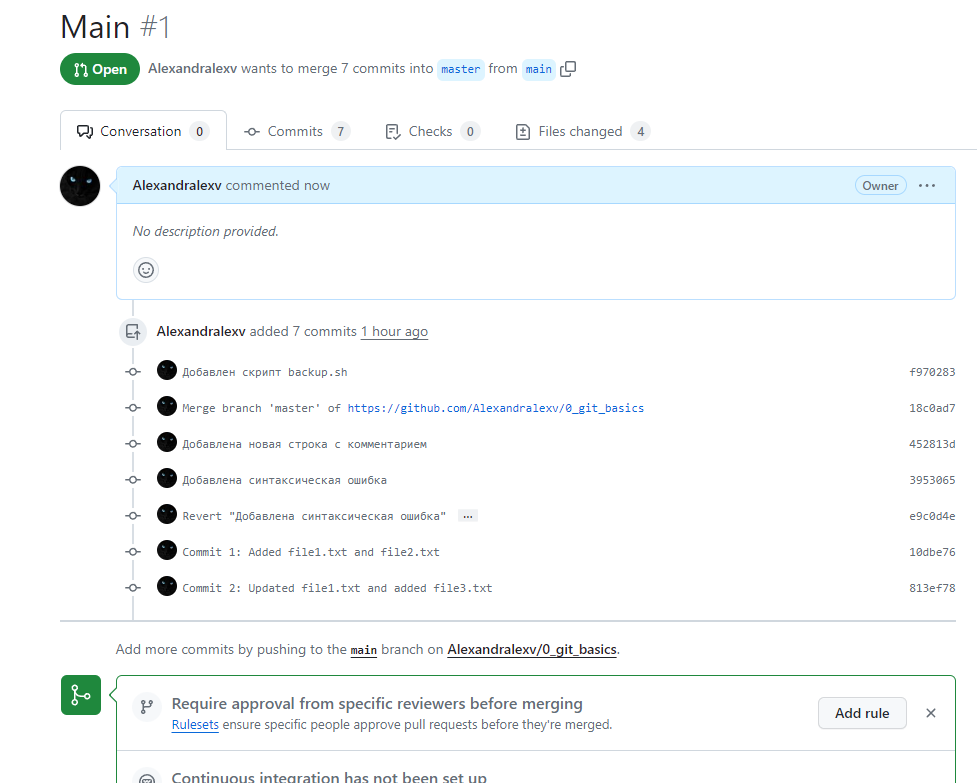


1. Создать пулл реквест в данный репозиторий









**Контрольные вопросы**

1. Зачем нужен облачный хостинг репозиториев?

Облачные хостинги репозиториев, такие как GitHub, GitLab, Bitbucket и другие, предоставляют удобное и надежное хранилище для ваших проектов с системами управления версиями, такими как Git. Вот несколько причин, почему облачный хостинг может быть полезен:

- \*\*Доступность данных:\*\* Репозитории хранятся в облаке, что обеспечивает доступность данных из любого места и устройства с подключением к интернету.

- \*\*Коллаборация:\*\* Облачные хостинги предоставляют инструменты для совместной разработки, что позволяет командам эффективно работать вместе, отслеживать изменения и рецензировать код.

- \*\*Интеграция:\*\* Многие облачные платформы интегрированы с другими сервисами и инструментами разработки, такими как CI/CD, системы отслеживания ошибок, чаты и документация.

- \*\*Управление правами доступа:\*\* Вы можете легко управлять доступом к своим репозиториям, предоставляя разные уровни доступа для участников проекта.

1. Какими основными функциями обладает сайт github.com?

- \*\*Хостинг репозиториев:\*\* GitHub предоставляет облачный хостинг для ваших Git-репозиториев, что облегчает совместную разработку.

- \*\*Отслеживание изменений:\*\* GitHub позволяет просматривать историю коммитов, вносить изменения в код, а также оставлять комментарии и рецензировать код.

- \*\*Pull Requests (Пулл-реквесты):\*\* Этот механизм позволяет предложить изменения из одной ветки в другую, обсуждать эти изменения и, при необходимости, объединять их.

- \*\*Issues (Задачи):\*\* GitHub предоставляет систему отслеживания задач и багов, которая помогает организовать их обработку.

- \*\*Actions и Workflows:\*\* Инструменты для настройки автоматизированных процессов с использованием Continuous Integration (CI) и Continuous Deployment (CD).

- \*\*Коллаборация и социальные функции:\*\* Возможность фолловить других разработчиков, обсуждение изменений в коде, общение в Issues и Pull Requests, а также другие социальные возможности.

1. Как организовать командную работу над открытым проектом?

- \*\*Использование Forks:\*\* Участники могут форкнуть (сделать копию) основного репозитория, внести изменения в своем форке, а затем предложить их для объединения через Pull Request.

- \*\*Управление доступом:\*\* Организаторы проекта могут управлять правами доступа, определять роли участников, что обеспечивает безопасность и контроль над кодом.

- \*\*Использование Issues:\*\* Для организации работы и обсуждения задач проекта можно использовать систему Issues, где каждая задача имеет уникальный номер и может быть привязана к коду.

- \*\*Pull Requests и Рецензии:\*\* Участники могут предлагать изменения через Pull Requests, а другие участники могут просматривать, комментировать и рецензировать эти изменения.

- \*\*Использование CI/CD:\*\* Настройка автоматизированных процессов с использованием Continuous Integration и Continuous Deployment для проверки и автоматической поставки кода.