Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

ПО МДК 04.02 «Обеспечение качества функционирования компьютерных систем»

|  |  |
| --- | --- |
| Выполнил студент  группы П50-7-20  Антонова М.М. | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_ К.С. Образцова  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 года |

Москва 2023

СОДЕРЖАНИЕ

[ПРАКТИЧЕСКАЯ РАБОТА №1 3](#_Toc145504636)

[ПРАКТИЧЕСКАЯ РАБОТА №2 5](#_Toc145504637)

# ПРАКТИЧЕСКАЯ РАБОТА №1

1. Продемонстрировать конфигурацию Git

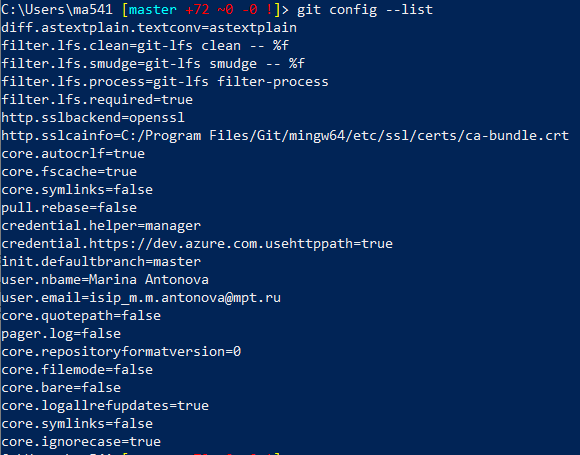


Рисунок 1. Конфигурация Git

1. Создание локального репозитория

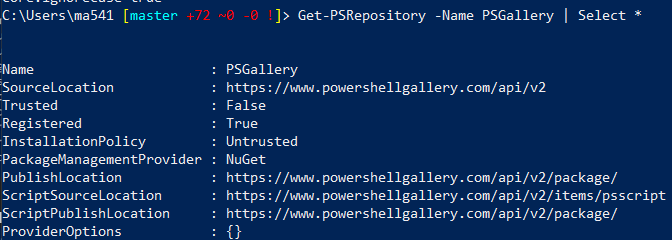


Рисунок 2. Создание репозитория

1. Выполнение простых операций



Рисунок 3. Задаем имя пользователя и адрес электронной почты



Рисунок 4. Кэширование учетных данных



Рисунок 5. Инициализация репозитория



Рисунок 6. Сравнение рабочей директории с областью подготовленных файлов



Рисунок 7. Перемещение области назад

Вывод: во время выполнения практической работы получила навыки базовой работы с репозиториями и git в целом.

# ПРАКТИЧЕСКАЯ РАБОТА №2

Цель работы: продолжить ознакомление с git, выполнить работу с ветками и познакомиться с GitHub.

Git-hosting – платформы, которые предоставляют инфраструктуру для хранения и управления репозиториями Git. Они позволяют разработчикам работать с Git репозиториями удаленно, управление версиями, отслеживать изменения и сотрудничать с другими разработчиками. (кто-то пушит проект, а другие могут подтянуть себе его обновление).

Git-hosting: GitHub, GitLab

GitLab – хостинг платформа для хранения и управления кодом. Она предоставляет функциональность, аналогичную GitHub, но с открытым исходным кодом. GitLab могут быть развернут на собственных серверах, что позволяет организациям иметь полный контроль над своими репозиториями. Он также предлагает широкий набор инструментов для управления проектами.

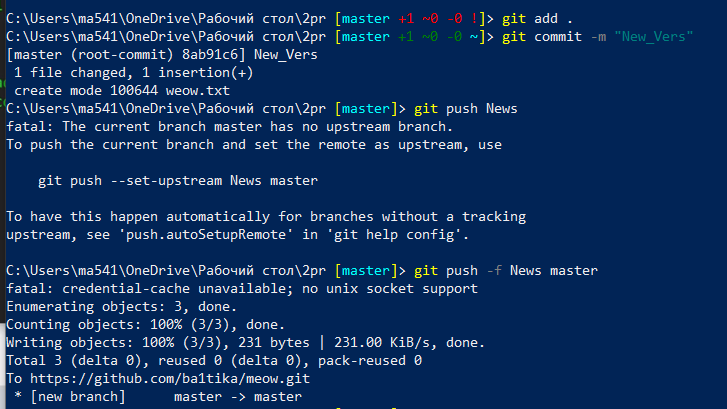


Рисунок 1. Git push

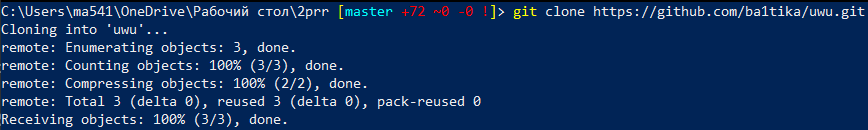


Рисунок 2. Git clone

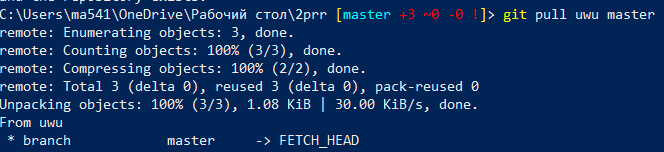


Рисунок 3. Git pull

Для взаимодействий, которые были представлены выше нужно зарегистрироваться на GitHub, сделать это очень просто:

1. Заходим на сайт: <https://github.com/>
2. Справа сверху находим кнопочку Sign Up и кликаем

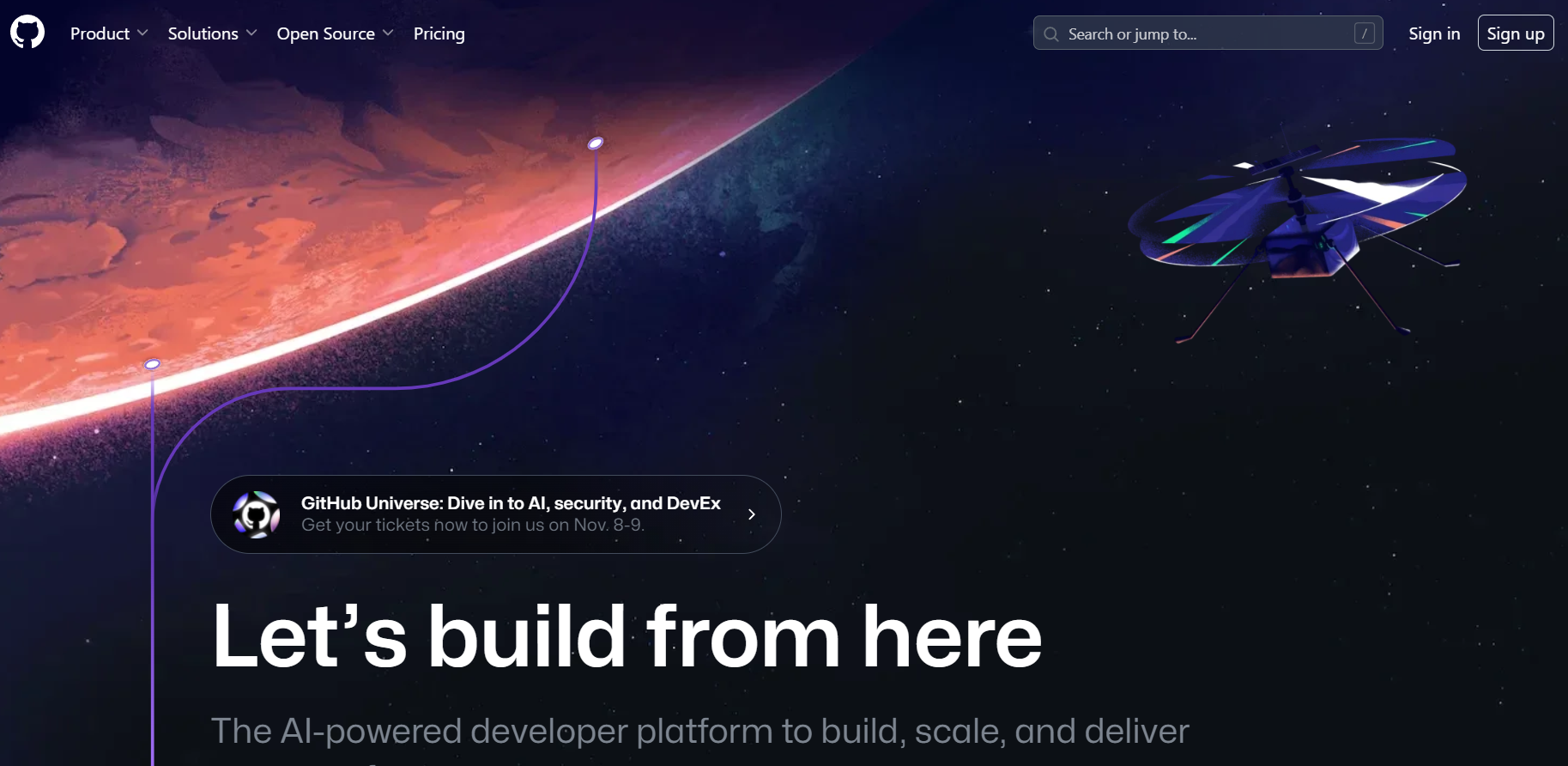


Рисунок 4. Главная страница сайта

1. Далее нужно указать почту. Затем придумываем надежный пароль и нажимаем продолжить. Придумываем ник-нейм, выбираем хотим ли получать оповещения на почту

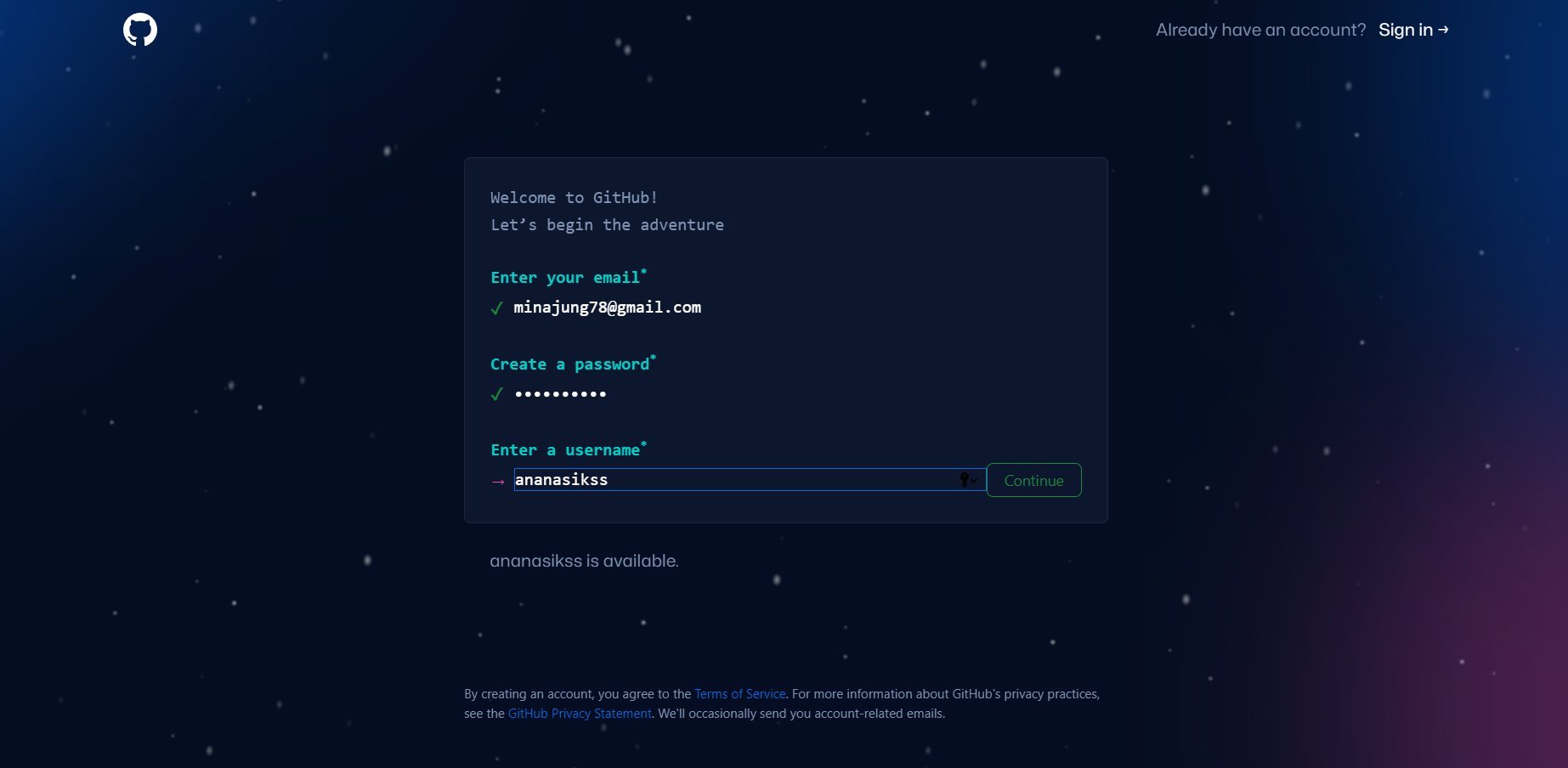


Рисунок 5. Регистрация представляет из себя заполнение полей с почтой, паролем, никнеймом

1. Последним пунктом будет проверка пользователя, пройдя её можно создать аккаунт

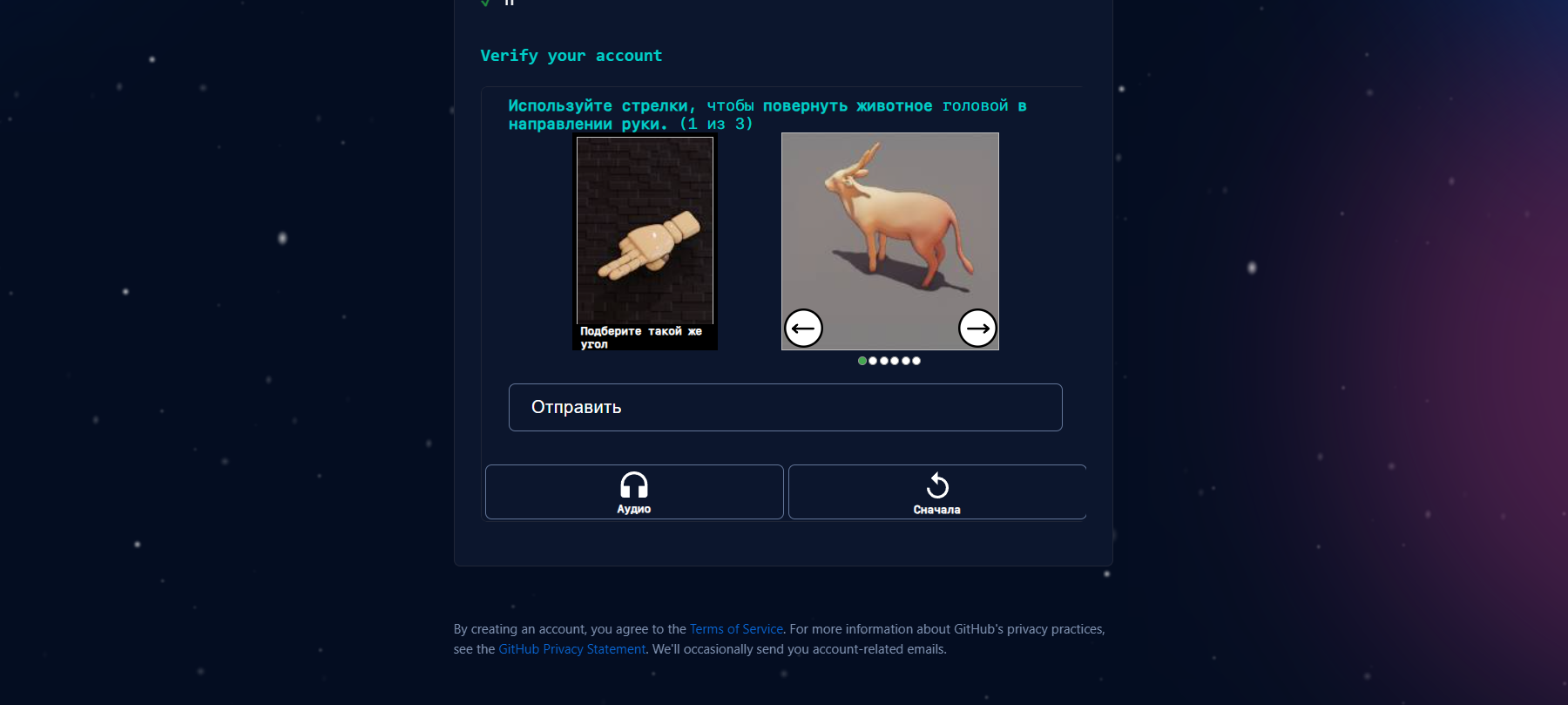


Рисунок 6. Проверка пользователя состоит из поворачивания животных

1. В конце нажимаем на заветную кнопочку

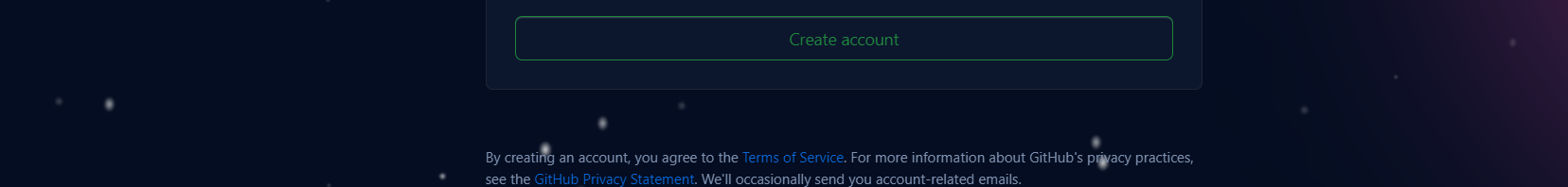


Рисунок 7. Регистрация завершиться после завершения всех пунктов, указанных выше и нажатия на кнопку Create account

В GitHub очень много интересных вкладок, в которых можно просто полазить, изначально познакомимся с боковой менюшкой. В краткости расскажу что, зачем и почему: Set status – кнопка для установления текстового статуса, классная вещь, Your profile – переместит на страницу профиля, где можно настроить профиль, смотреть репозитории и тд., Your repositories – окошко для просмотра и создания репозиториев, Your projects – окно для просмотра и создания проектов, Your codespaces – окно установки доп. сред, кратко – онлайн редакторы кода (прямо в GitHub), Your organizations – здесь храниться данные организаций, которые пользователь сам добавляет и настраивает, Your enterprises – для обеспечения совместной работы на GitHub, Your stars – список с избранными репозиториями.

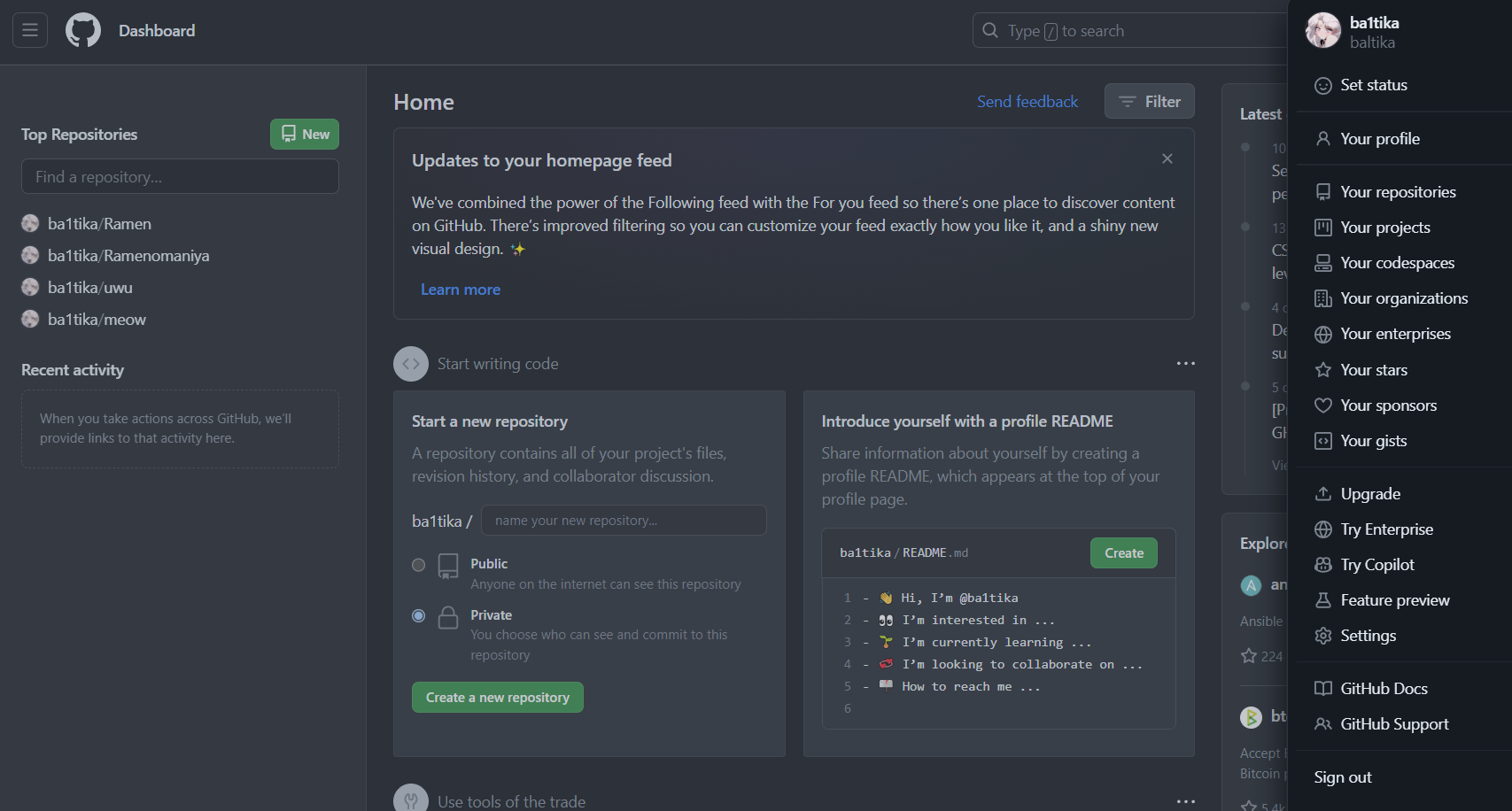


Рисунок 8. Боковое меню

На GitHub есть множество вкладок для работы с проектами. Overview – вкладка, которая показывает профиль и действие пользователя по календарю, также показаны популярные репозитории, что такое репозитории? Для них есть отдельная вкладка Repositories, она представляет с собой библиотеку проектов, которые когда-либо были добавлены пользователем. Здесь их можно просматривать, редактировать, в общем играться так как душе угодно. Идем дальше, в Projects также можно просматривать действия пользователя относительно открытия проектов и создать новый проект. Последняя вкладка – Packages это служба размещения пакетов программного обеспечения, которая позволяет размещать пакеты программного обеспечения для частного или общедоступного использования в качестве зависимостей в проектах.

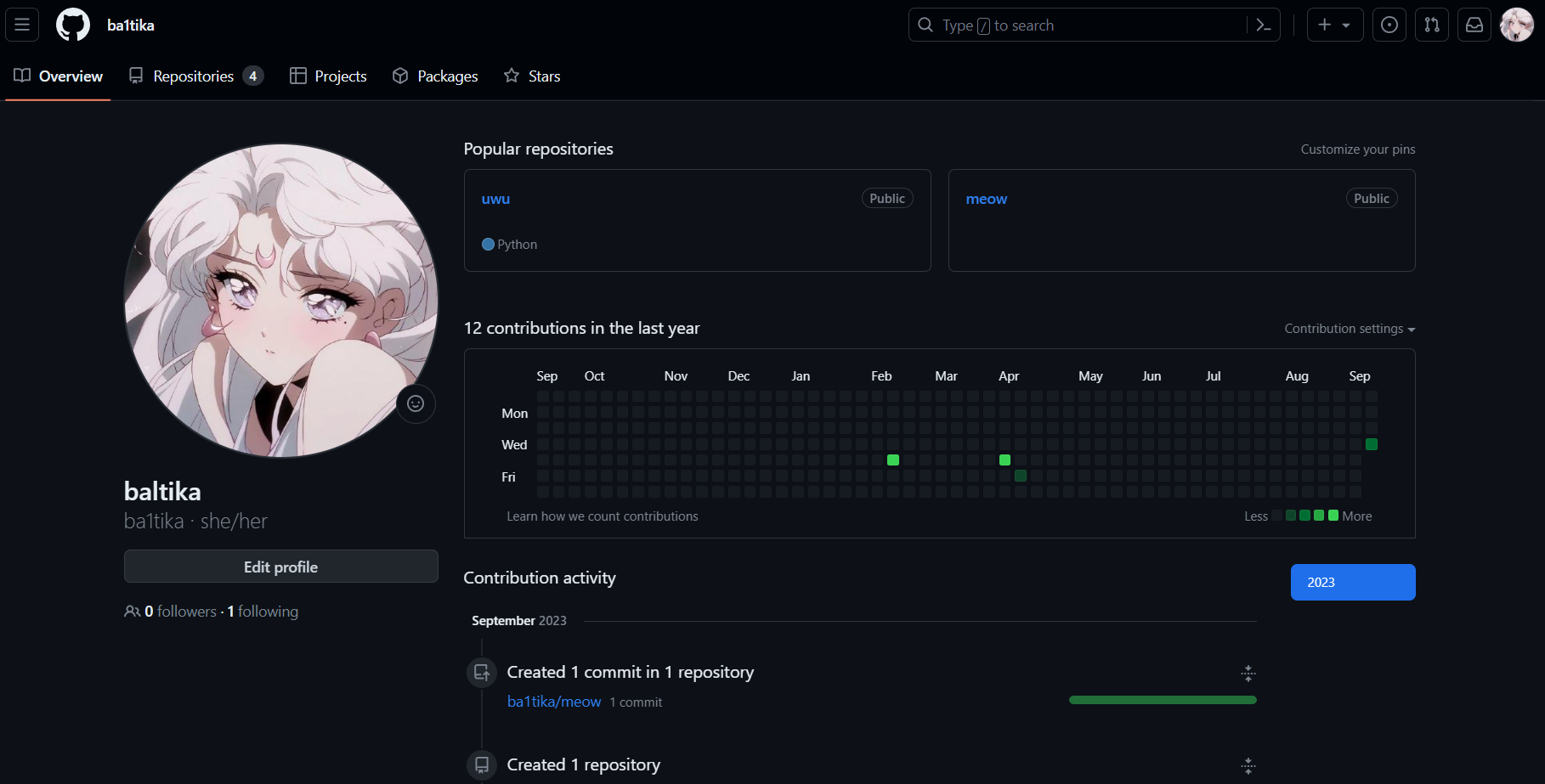


Рисунок 9. Вкладка Overview

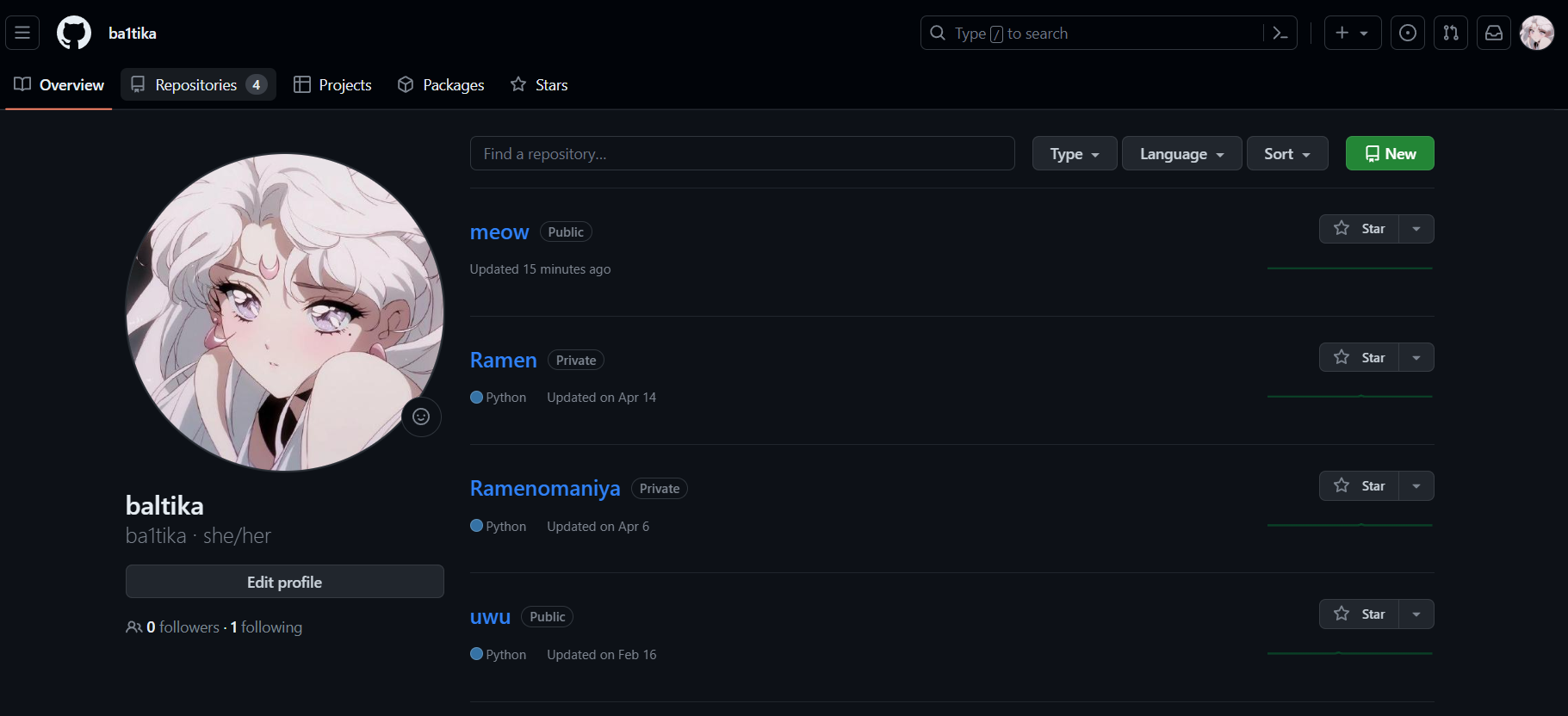


Рисунок 10. Вкладка Repositories

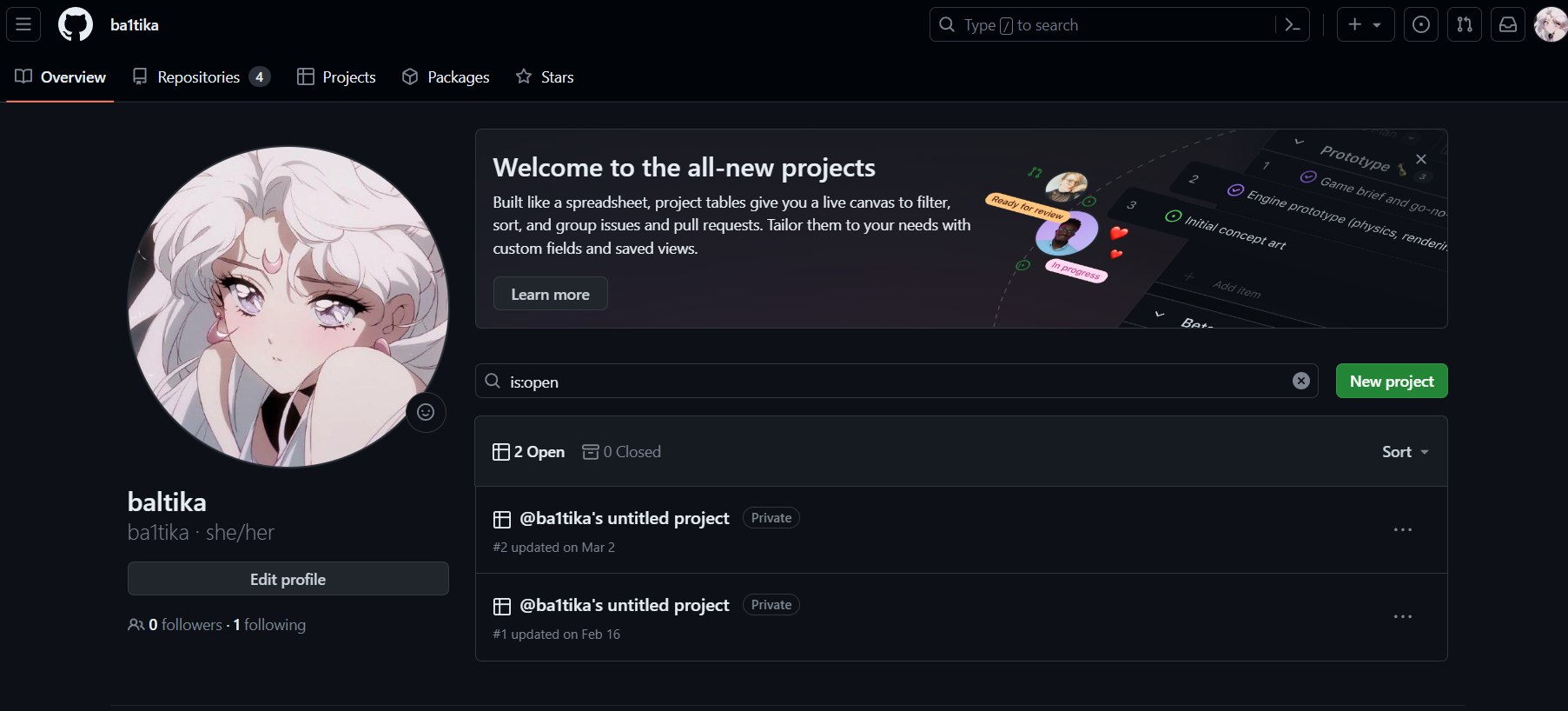


Рисунок 11. Вкладка Projects

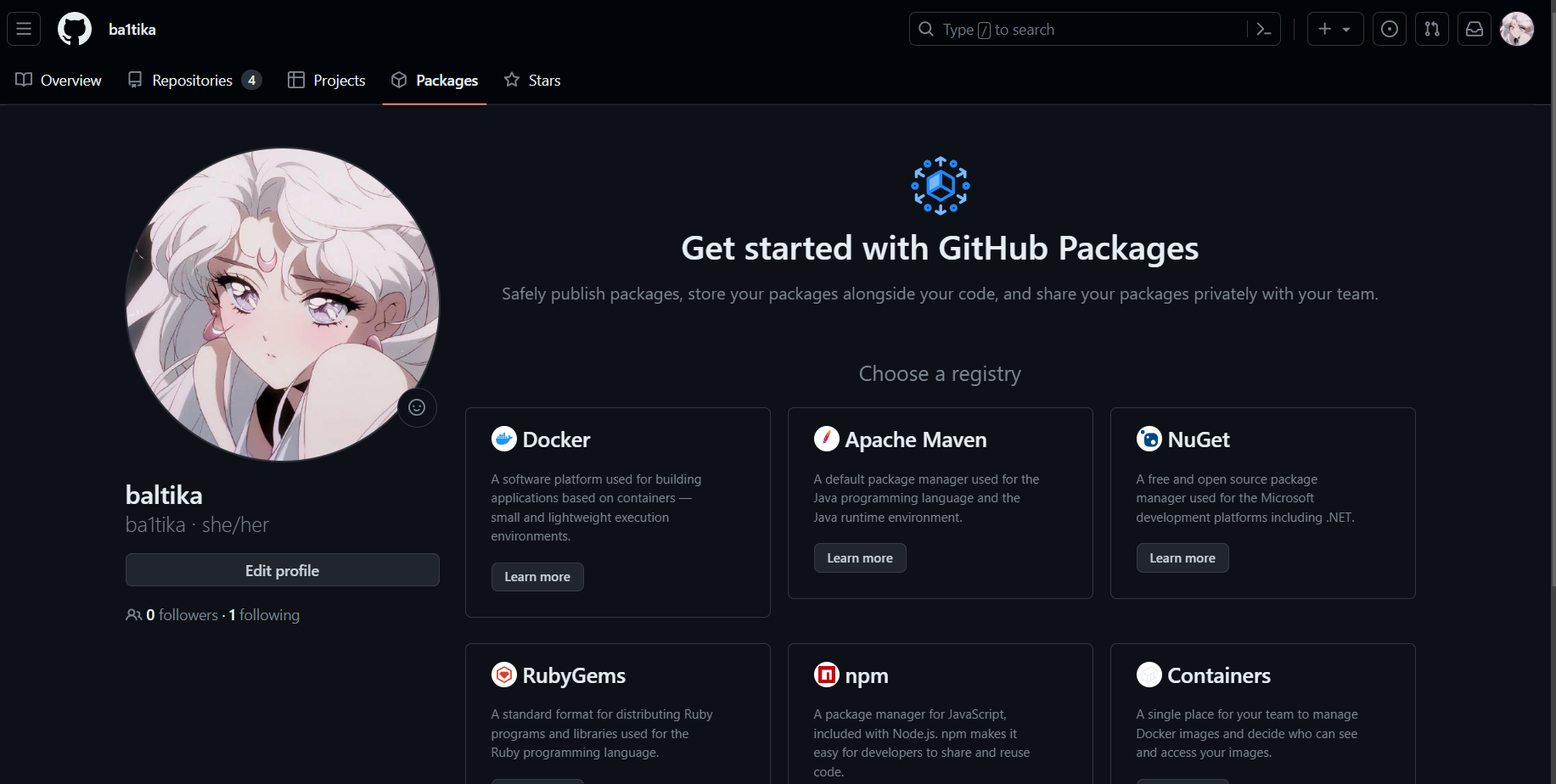


Рисунок 12. Вкладка Packages

Дальше в репозиториях есть ещё множество вкладок, у каждой есть своя функциональность. Code – в коде представлены все файлы репозитория, здесь их можно создать, просмотреть, Issues – система учета ошибок, Actions – автоматизация деплой процесса, Projects – окно работы с проектами, Wiki – создание документации к репозиторию, Security – настройка безопасности репозитория, Insights – что-то типо проверки активности репозитория, Settings – настройки проекта (далее в этой вкладке будем создавать branch).

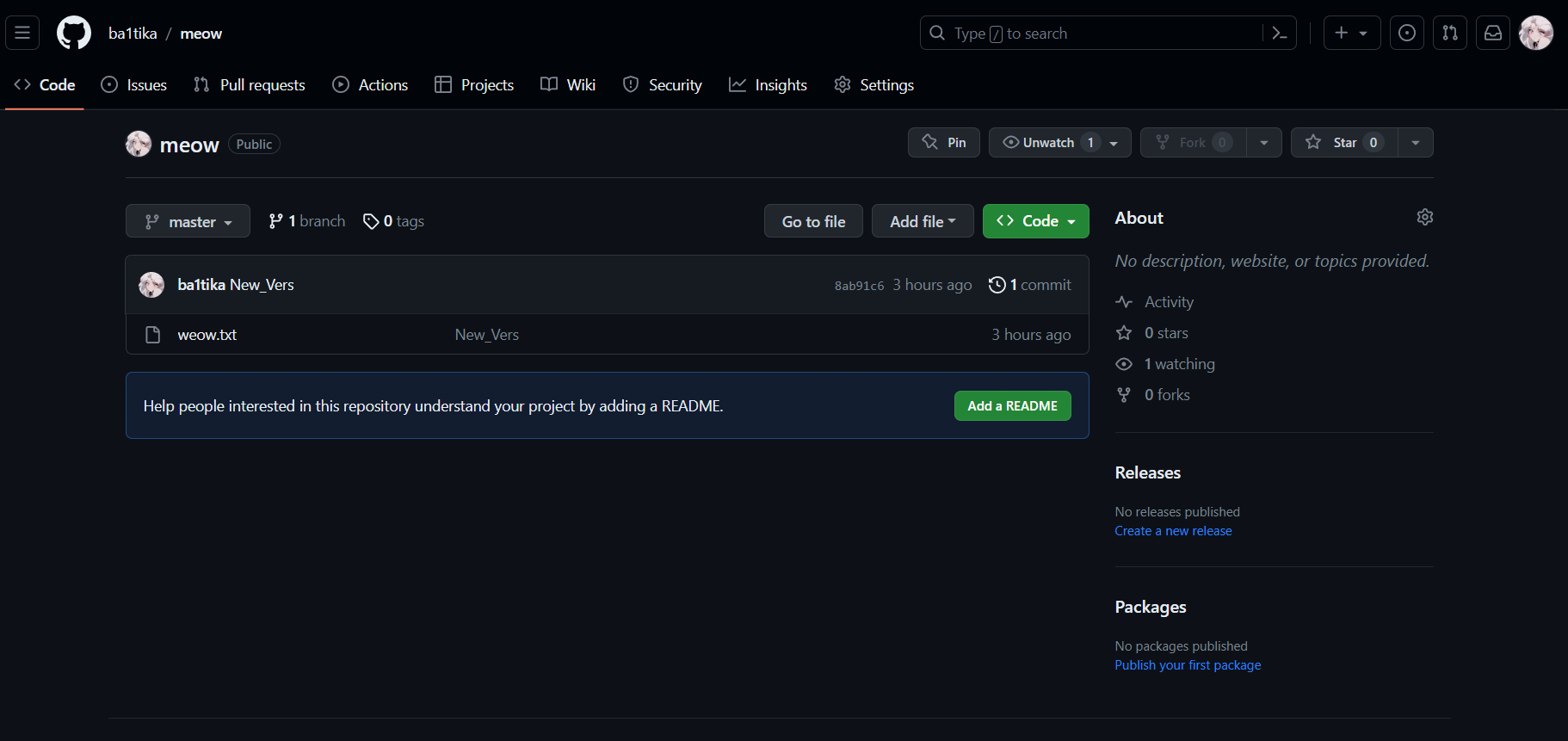


Рисунок 13. Вкладки репозитория

Для работы с ветками нужно добавить branch в репозиторий.

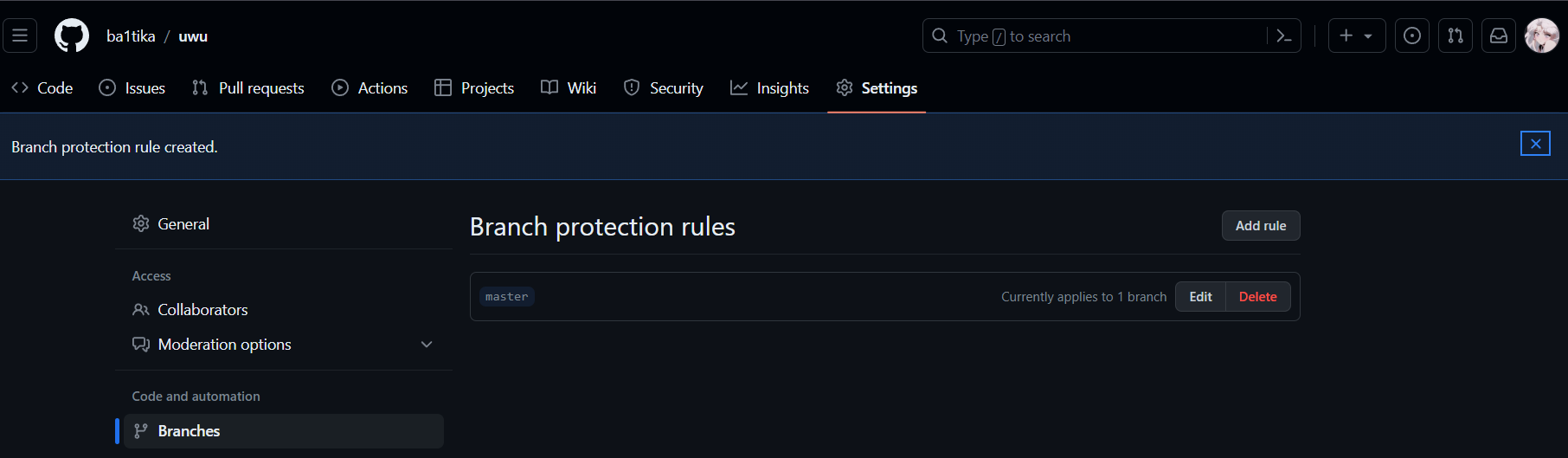


Рисунок 13. Создание branch (в его настройках нужно выбрать нужные поля с указанием ограничений ветки)



Рисунок 14. Работаем с репозиторием, в котором всего одна ветка

Вывод: во время выполнения работы закрепились навыки работы с репозиториями, GitHub и ограничениями ветвей.

# ПРАКТИЧЕСКАЯ РАБОТА №3

Цель работы: Docker.

Docker - открытая платформа для разработки, доставки и запуска приложений. Он позволяет упаковать приложение и его зависимости в контейнер, который может быть запущен на любой совместимой с докер системой. Контейнеры докер предлагают независимо от операционной системы среду выполнения.

Преимущества использования в нашем программном обеспечении:

Изолированная среда. Докер обеспечивает изолированное окружение, в котором приложение может работать независимо от других приложений и зависимостей на хостовой системе. Это позволит избежать конфликтов и снижает вероятность неправильной работы приложения. Поэтому разработчики могут не задумываться в каком окружении будет работать их приложение, а инженеры по эксплуатации – единообразно запускать положение и меньше заботиться о системных зависимостях.

Управление зависимостями. Докер позволяет включать все зависимости приложения, включая операционную систему, библиотеки и другие компоненты, в контейнер. Это упрощает управление зависимостями и обеспечивает согласованность окружения между разработчиками и на разных этапах разработки и может снизить проблемы, связанные с развертыванием приложения на разных серверах.

Масштабируемость. Докер позволяет масштабировать приложение по мере необходимости, добавляя новые контейнеры или запуская их на кластере докер. Это обеспечивает гибкость и удобство в управлении нагрузкой.

Переносимость. Докер позволяет легко перемещать контейнер с приложением между различными средами разработки, тестирование и продакшн. Это упрощает процесс развертывания и масштабирования приложений.

Dockerfile – это текстовый файл, содержащий инструкции для автоматического создания докер-образа. Он описывает шаги, которые необходимо выполнить для создания докер-контейнера, включая установку и настройку приложений, копирование файлов и настройку сетей. Докер файл позволяет разработчикам создавать повторяемые и легко воспроизводимые среды разработки и выполнения приложений, где каждый шаг указан явно и может быть автоматически выполнен при создании контейнера докер.

Контейнеры – это изолированные и легковесные среды, в которых запускается приложения и их зависимости. Они представляют собой запускаемые экземпляры образов докер, содержащих все необходимые для работы приложения. Их можно создавать, запускать, останавливать и удалять. Также можно подключать к контейнеру хранилище, объединять контейнеры одной или несколькими сетями и общаться с контейнерами, используя докер апи или командную строку.

Образ – это шаблон контейнеров. В образе описывается, что должно быть установлено в контейнере (например код, среда выполнения, библиотеки, переменные окружения и другие зависимости) и какие действия нужно выполнить при старте контейнера.

Используя один образ можно создавать много контейнеров.

Кластер докер – это совокупность компьютеров или серверов, которые объединены вместе для работы с контейнерами докер. Кластер докер использует контейнеры для запуска и управления приложением на нескольких компьютерах одновременно. Кластер обеспечивает масштабированность и надежность, по скольку если один компьютер выходит из строя, другие компьютеры в кластере могут продолжать работать без проблем. Кластер также позволяет распределить нагрузку между компьютерами, чтобы обеспечить эффективное использование ресурсов.

Докер компос – инструмент, который позволяет запускать и управлять несколькими контейнерами докер одновременно. Он позволяет описывать структуру и конфигурацию вашего приложения в файле YAML, что делает его очень простым для понимания и использования. Докер компос позволяет запускать и остановить все ваши контейнеры одной командой, а также связывать их между собой, чтобы они могли взаимодействовать друг с другом.

Докер демон – основной компонент в системе докер, который управляет выполнениями контролем контейнеров. Он работает в фоновом режиме и управляет ресурсами, созданием и удалением контейнеров, а также назначением им ресурсов.

Докер клиент – инструмент командной строки, который позволяет управлять контейнерами докер (через него пользователь взаимодействует с демон) позволяет выполнять различные операции с контейнерами, такие как создание, запуск, остановка, удаление и мониторинг, используя простые команды в командной строке.