МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Направление подготовки (шифр, название) 09.03.04 «Программная инженерия»

**Отчет по практике**

обучающегося 1 курса

Фамилия Нестеренко

Имя Пётр

Отчество Алексеевич

|  |  |  |
| --- | --- | --- |
| Обучающийся |  | Нестеренко П. А. |
|  | подпись | расшифровка подписи |

Место практики Научно-исследовательская лаборатория “Лаборатория мобильной и веб разработки” ООО НПИЦИТ “Иносфера” Кафедры математического обеспечения и применения ЭВМ ЮФУ

наименование профильной организации

Вид практики Учебная практика

Тип практики Ознакомительная практика, распределенная

Способ проведения практики Стационарная

Форма проведения практики Дискретная (по видам и периодам практик)

Сроки прохождения практики *с* 10.02.2020 *по* 21.06.2020

|  |
| --- |
| Руководитель практики  от структурного подразделения ЮФУ |
| Пирская Любовь Владимировна |

ФИО, подпись

Содержание

[1 Постановка задачи 3](#_Toc39331835)

[2 Описание выполненных работ 4](#_Toc39331836)

[2.1 Установка технологии контроля версий на базе Git 4](#_Toc39331837)

[2.2 Создание локального репозитория 4](#_Toc39331838)

[2.3 Создание и добавление удаленного репозитория 5](#_Toc39331839)

[2.4 Создание ветвлений 6](#_Toc39331840)

[2.5 Слияние ветвлений 8](#_Toc39331841)

[2.6 Создание запроса на слияние 9](#_Toc39331842)

[2.7 Разрешение конфликтов локально 11](#_Toc39331843)

[2.8 Разрешение конфликтов на GitLab 13](#_Toc39331844)

[2.9 Конфигурация сервера по умолчанию 14](#_Toc39331845)

[2.10 Конфигурация виртуального сервера 15](#_Toc39331846)

[3 Ссылки 17](#_Toc39331847)

[Заключение 18](#_Toc39331848)

[Список использованных источников 19](#_Toc39331849)

[Приложение А 20](#_Toc39331850)

1. Постановка задачи

1.1 Необходимо изучить и внедрить в работу технологию конфигурационного контроля на базе Git:

a. Установить git локально на рабочей станции.

b. Перенести в git один из предыдущих проектов

c. Создать учетную запись в GitLab

d. Связать локальный репозиторий с проектом в Gitlab.

e. Выполнить локально и отразить в gitlab: ветвление, слияние, создание запросов на слияние, принятие/отклонение запросов на слияние, разрешение конфликтов (локальные и gitlab скриншоты).

1.2 Необходимо изучить конфигурирование и подготовить платформу nginx для следующего проекта:

a. Установить nginx

b. Сконфигурировать сервер по умолчанию

c. Сконфигурировать виртуальный сервер

1.3 Факультативно: Подготовить локальный Dockerfile для сборки окружения nginx в контейнер, продемонстрировать его запуск.

1. Описание выполненных работ
   1. Установка технологии контроля версий на базе Git

Установить Git можно на современные Unix-подобные операционные систем, такие как Windows, Mac OS, Linux. Пример установки системы контроля версий на OS Linux можно видеть на Рисунке 1.

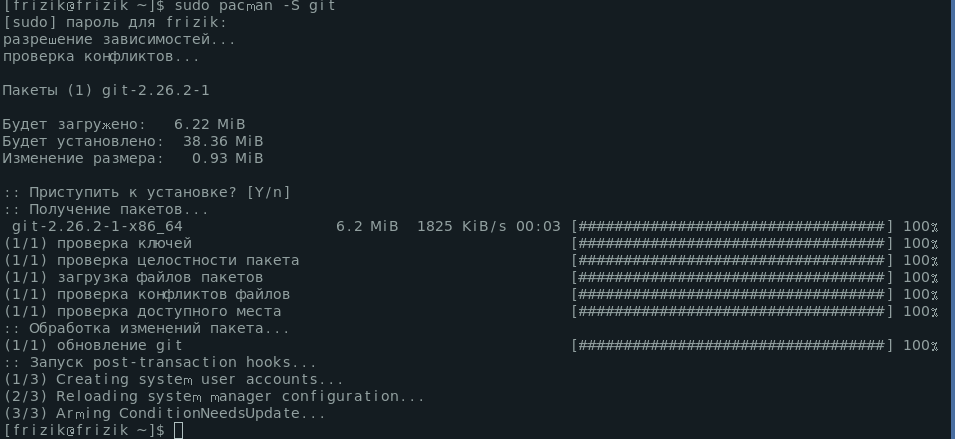


Рисунок 1 — Установка системы контроля версий

* 1. Создание локального репозитория

Создание локального репозитория Git происходит командой.

$ git init

После чего, добавляем все файлы проекта на отслеживание в Git следующей командой (в папке находятся исходники практики по разработке андроид приложения).

$ git add .

Далее, делаем самый первый commit – фиксацию последних изменений в файлах. В нашем случае происходит закрепление начального состояния файлов. Пример выполнения этой операции можем видеть на Рисунке 2.

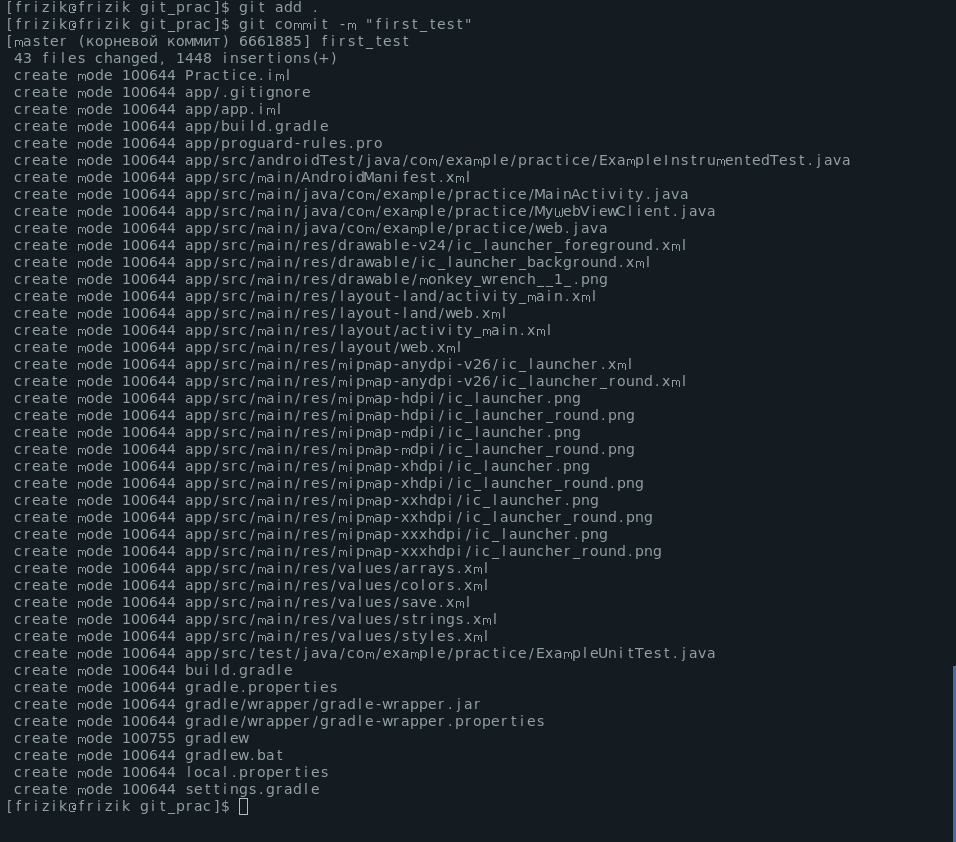


Рисунок 2 — Фиксация первоначальных файлов в репозитории

* 1. Создание и добавление удаленного репозитория

Создание удаленного репозитория происходит с помощью веб-инструмента GitLab. Процесс создания можно видеть на Рисунке 3.

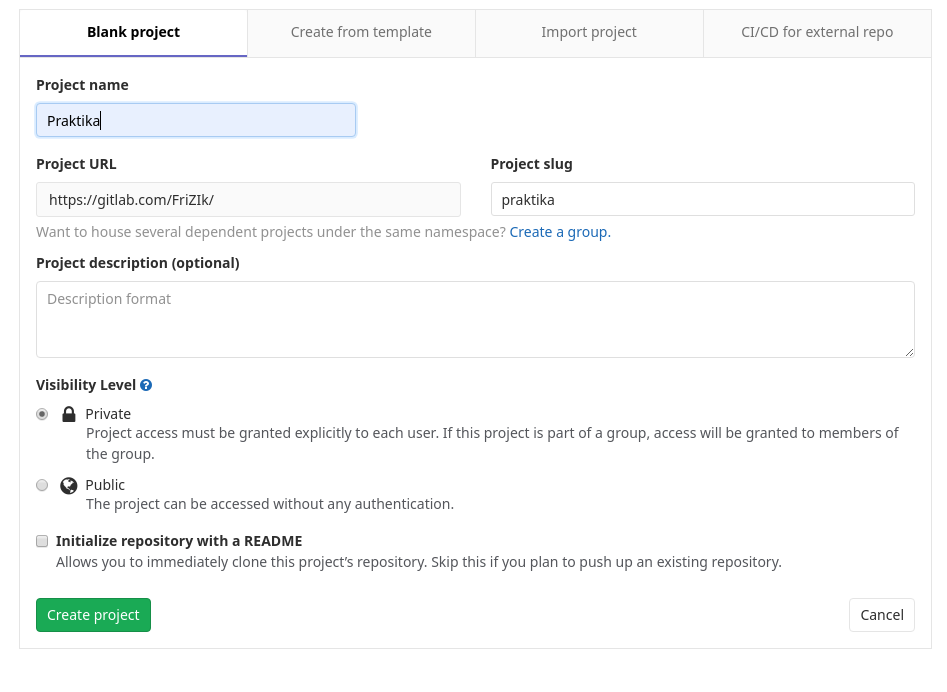


Рисунок 3 — Создание удаленного репозитория

Добавим удаленный репозиторий в наш локальный репозиторий и загрузим последние изменения на удаленный репозиторий. Сделаем это с помощью следующих команд:

$ git remote add origin https://gitlab.com/FriZIk/praktika.git

$ git push origin master

Результат выполнения данных операций можно видеть на Рисунке 4 и на Рисунке 5.

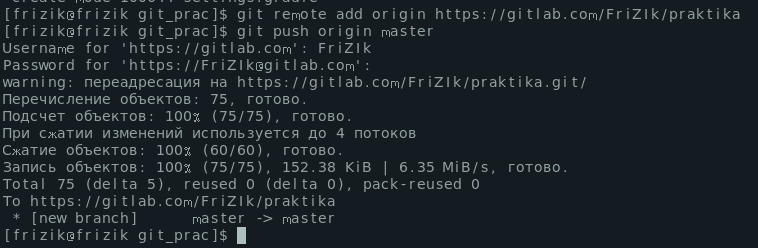


Рисунок 4 ­— Добавление удаленного репозитория с последующей загрузкой файлов на него

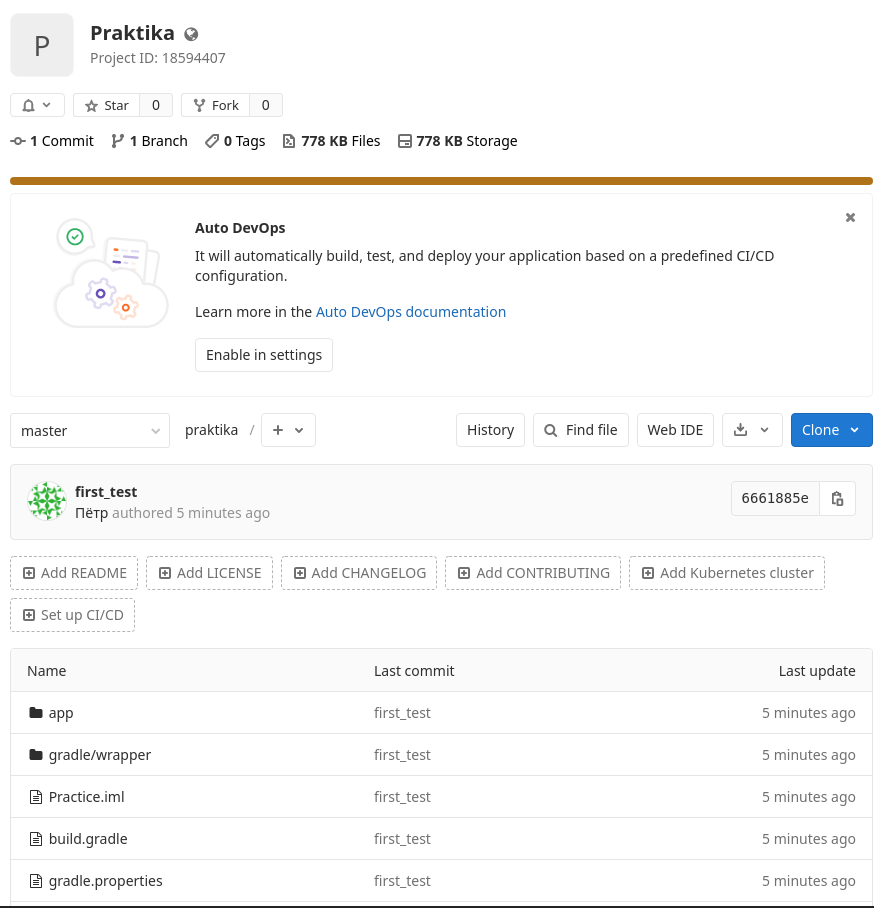


Рисунок 5 — Наш удаленный репозиторий после загрузки

* 1. Создание ветвлений

Дополнительные ветки проекта нужны для того, чтобы программисты могли вести совместную разработку проекта и при этом не мешать друг другу. При создании проекта, по умолчанию основной веткой является ветвь master.

Для создания новой ветки следует выполнить команду:

$ git branch test\_branch(название для примера)

Для вывода списка доступных ветвлений следует выполнить команду:

$ git branch

Результат выполнения данных команд можно увидеть на Рисунке 6.

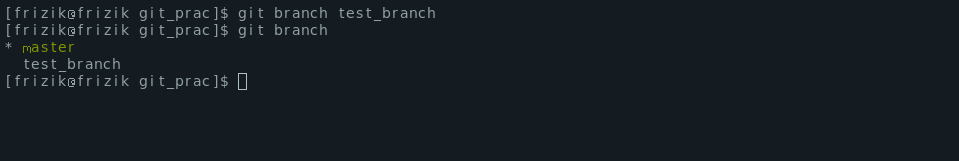


Рисунок 6 — Создание новой ветви

Для внесения изменений в новую ветвь, следует её выбрать. Это делается следующей командой.

$ git checkout test\_branch(вместо test\_branch следует вписать название созданной ветки, если она называется иначе)

Далее, после изменения содержимого файлов проекта, мы можем посмотреть, какие были внесены изменения в эти файлы, относительно последней версии проекта. Делается это следующей командой.

$ git diff

Результат выполнения данной команды можно видеть на Рисунке 7.

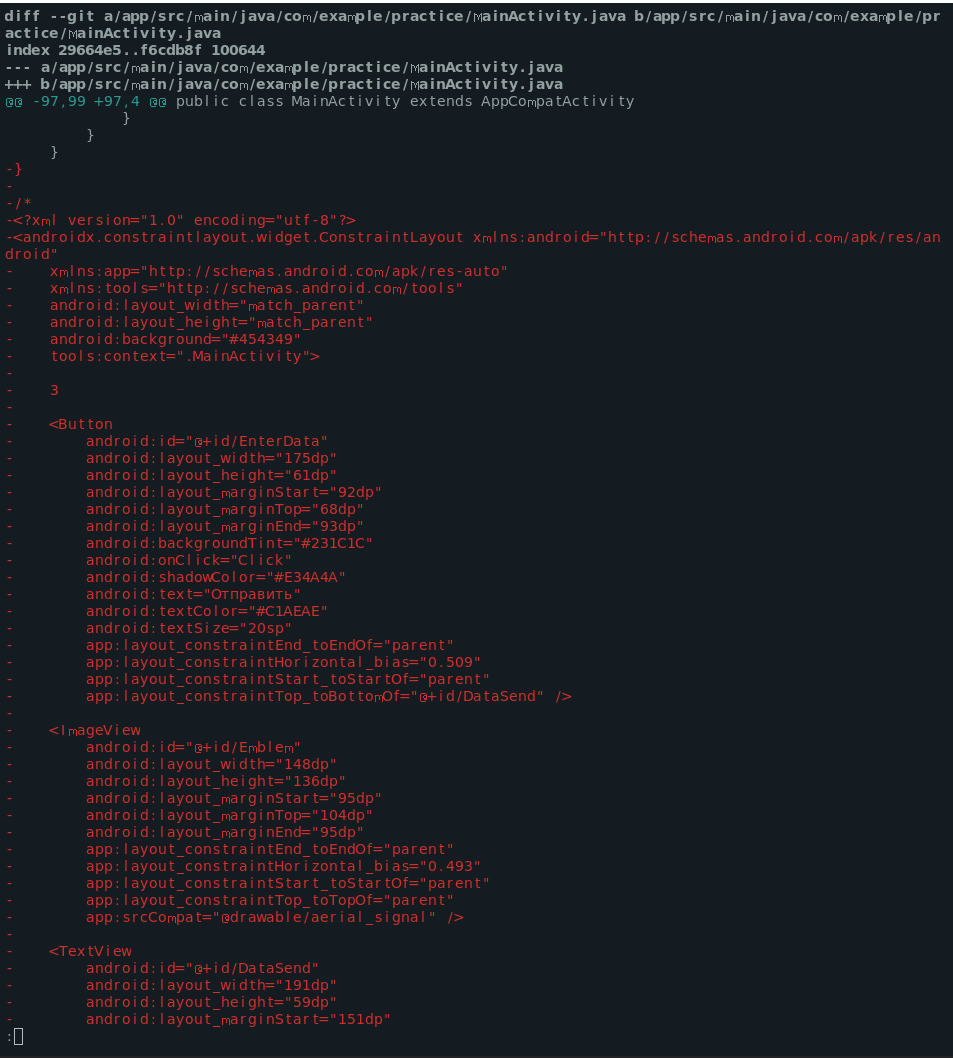
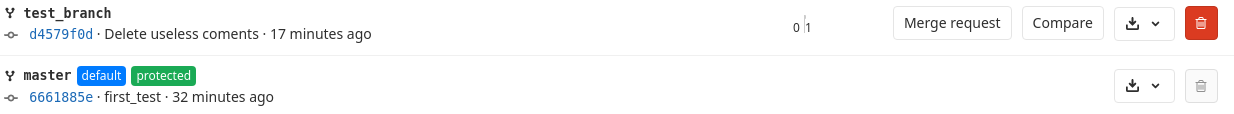


Рисунок 7 — Локальные изменения в файлах

После всех действий загружаем изменения на удаленный репозиторий. (см. Рисунок 8)

Рисунок 8 — Новое ветвление на удаленном репозитории

* 1. Слияние ветвлений

Слияние ветвлений нужно для того, чтобы наработки разных разработчиков объединить в единый проект. Слияние выполняется следующей командой.

$ git merge test\_branch

Пример выполнения данной команды можно видеть на Рисунке 9.



Рисунок 9 — Слияние ветви

В итоге можем видеть, что в ветке master появился новый commit, который ранее был в другой ветке (см. Рисунок 10, Рисунок 11).



Рисунок 10 — Результат слияния ветвления

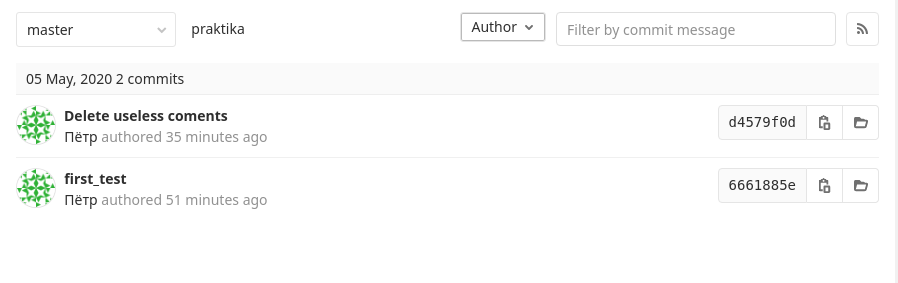


Рисунок 11 — Результат слияния, отраженный в GitLab

* 1. Создание запроса на слияние

Создание запроса на слияние происходит на странице репозитория в GitLab в разделе ветвлений (см. Рисунок 12 и Рисунок 13).

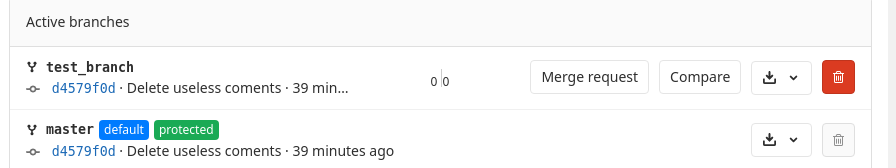


Рисунок 12 — Раздел ветвлений репозитория

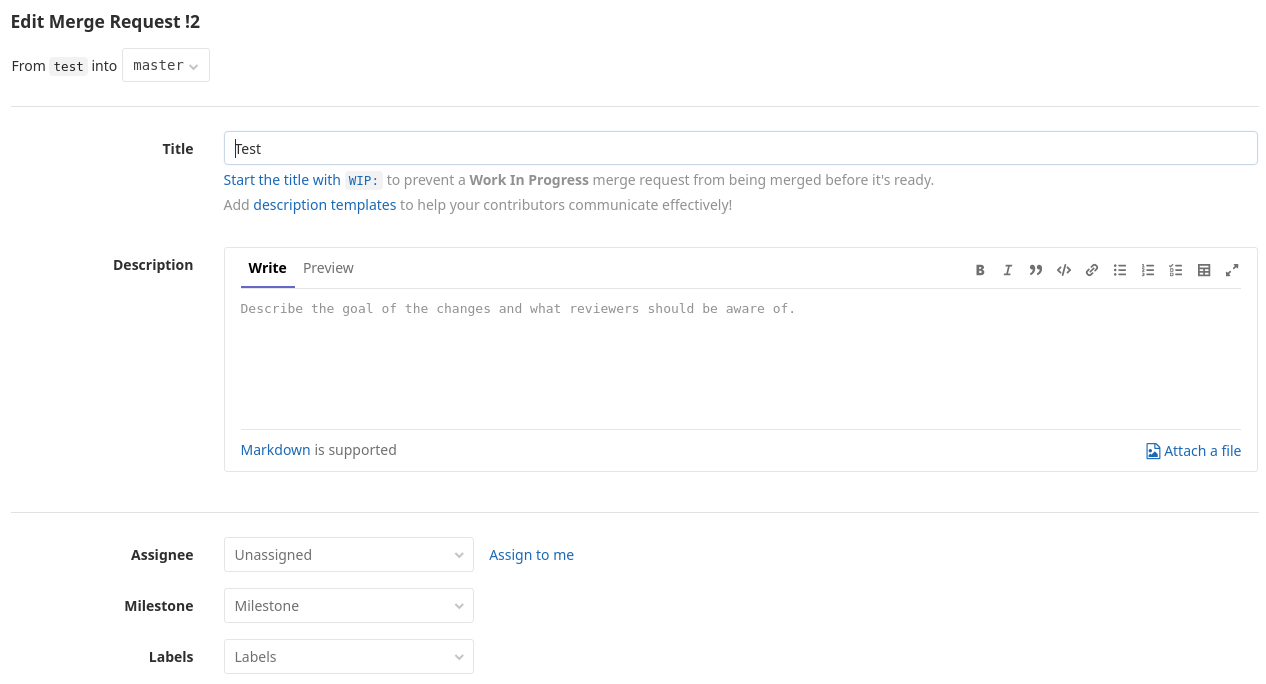


Рисунок 13 — Запрос на слияние

Принятие или отклонение запроса на слияние происходит в разделе Merge Requests (см. Рисунок 14).

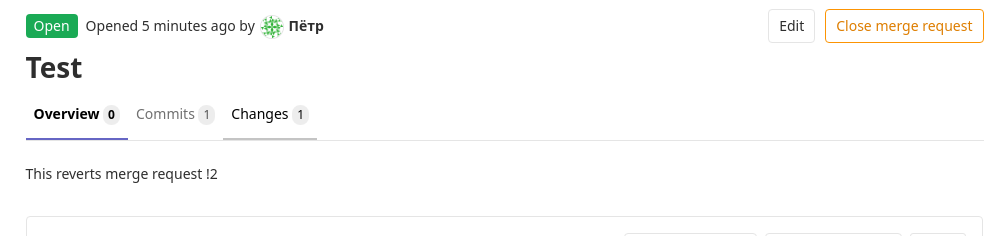


Рисунок 14 — Интерфейс принятия и отклонения запросов на слияние

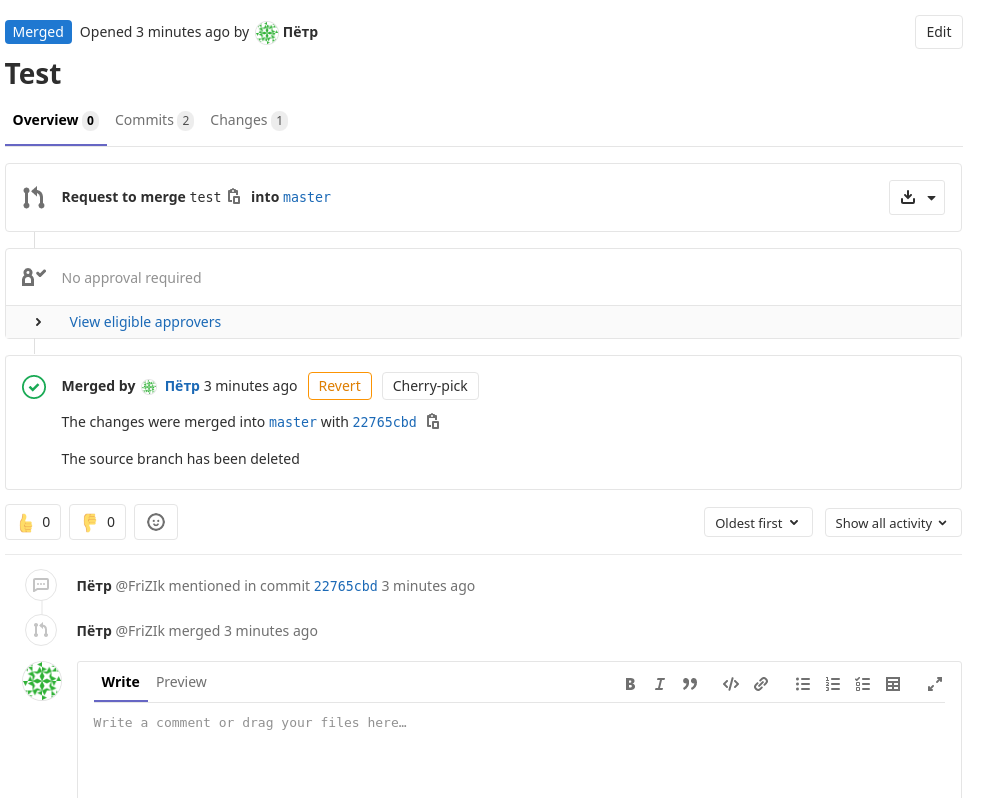


Рисунок 15 — Принятие запроса на слияние

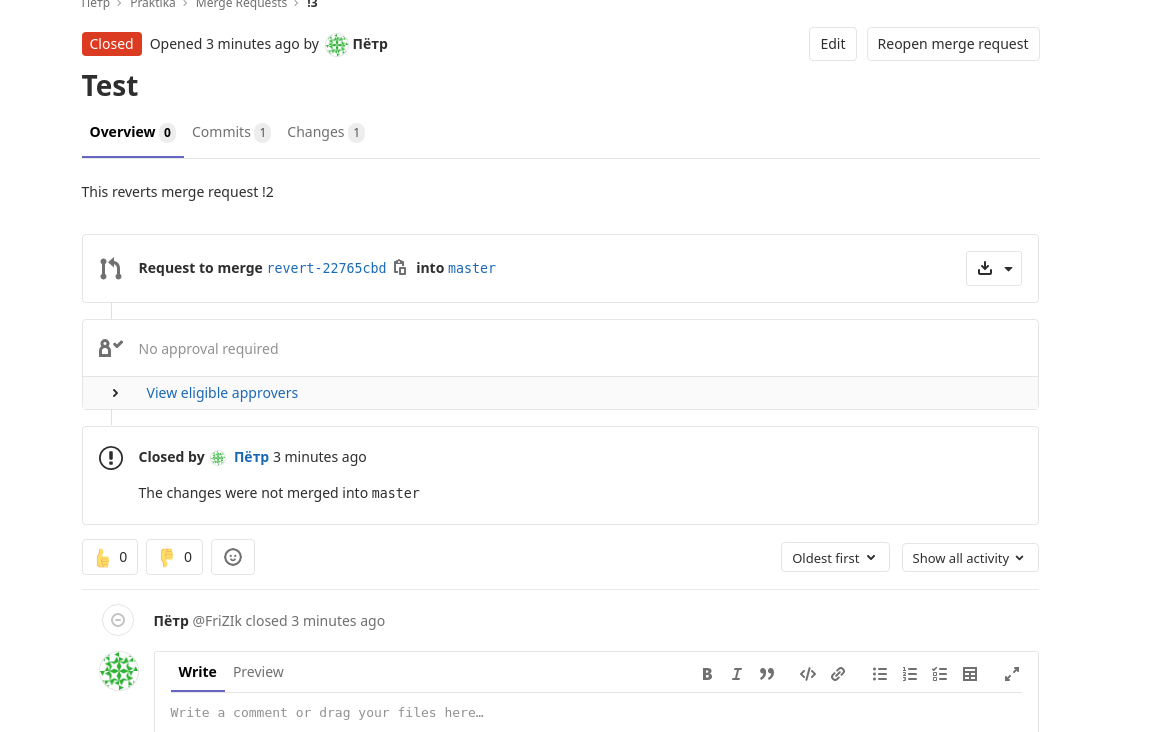


Рисунок 16 — Отклонение запроса на слияние (возврат)

* 1. Разрешение конфликтов локально

Конфликты происходят при объединении ветвлений, если в них было произведено изменения одного и того же файла.

Делаем новую ветвь, в которой вносим следующие изменения (см. Рисунок 17).



Рисунок 17 — Изменения в новой ветви

После чего вносим дополнительные изменений в основную ветвь master (см. Рисунок 18).

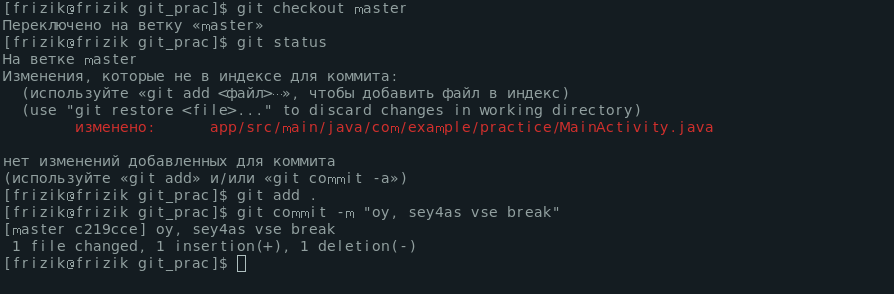


Рисунок 18 — Изменения в основной ветви master

При попытке выполнить слияние, появится сообщение о конфликте (см. Рисунок 19).

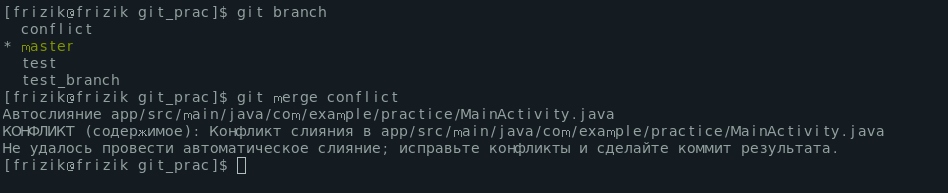


Рисунок 19 — Неудачная попытка провести слияние

В итоге выполнения данной операции, в наш изменяемый файл внесутся изменения из двух конфликтующих ветвлений следующим образом (см. Рисунок 20).

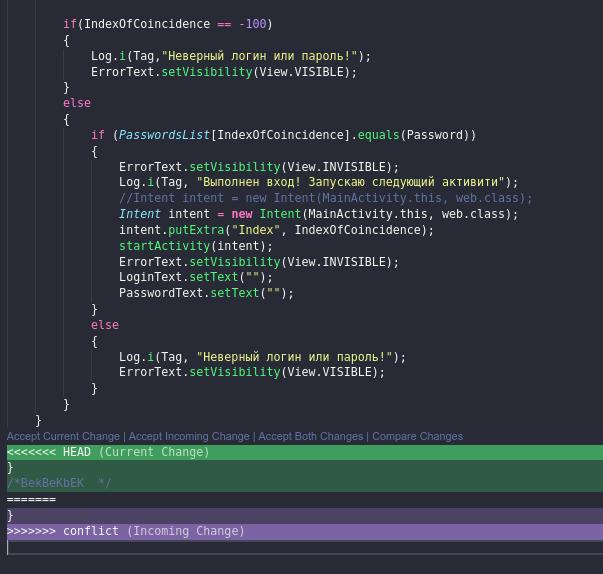


Рисунок 20 — Конфликты ветвлений в файле программы

Далее следует исправить файлы и сделать заключительный коммит, закрепляющий результат слияния ветвлений (см. Рисунок 21).

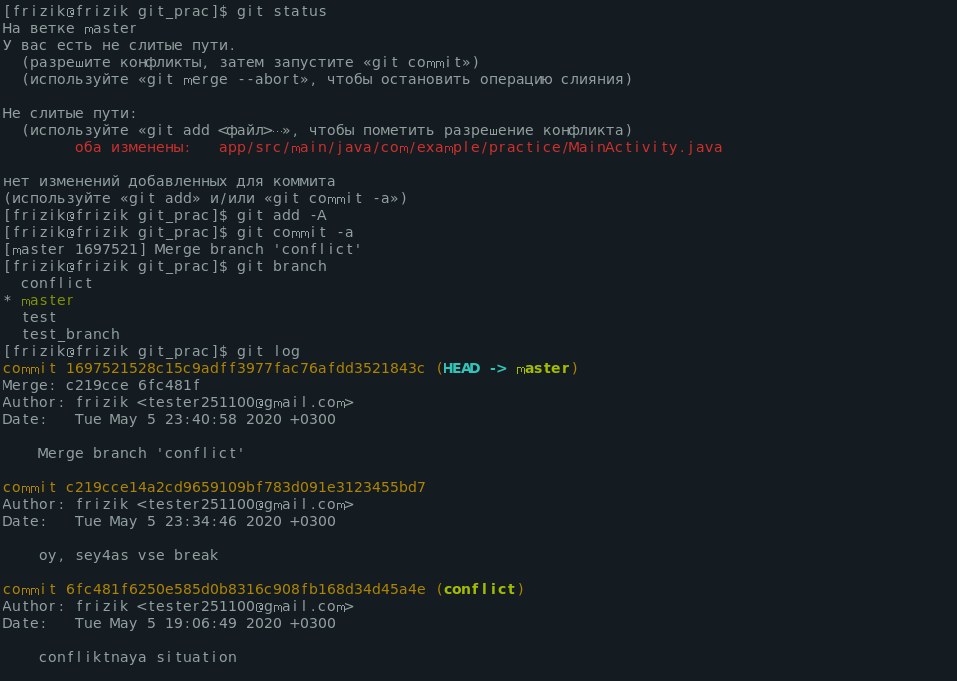


Рисунок 21 —Результат слияния ветвлений

* 1. Разрешение конфликтов на GitLab

Конфликт может произойти после принятия запроса на слияние (см. Рисунок 22).

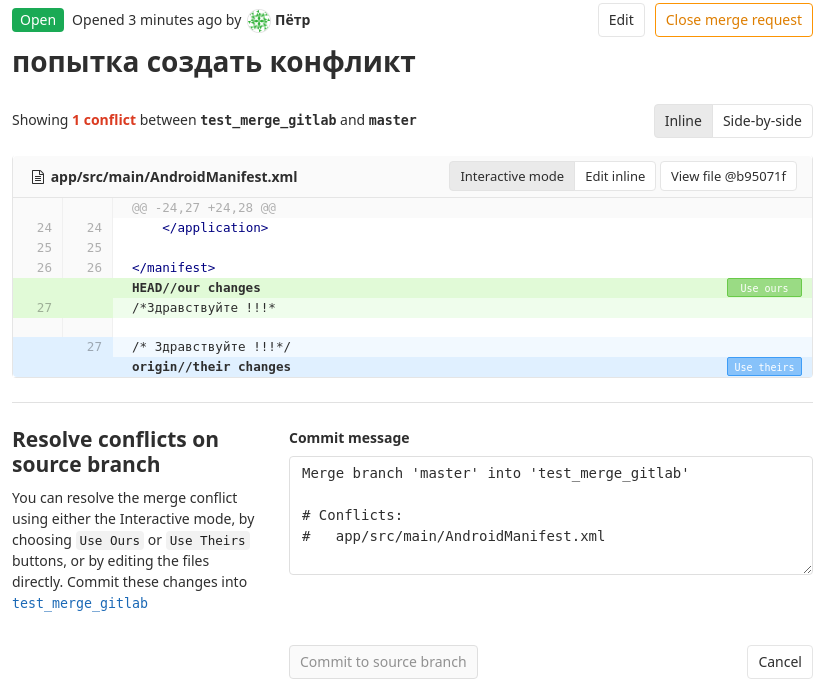


Рисунок 22 — Разрешение конфликта на GitLab

* 1. Конфигурация сервера по умолчанию

После скачивания Nginx, его можно запустить с помощью команды:

$ sudo systemctl start nginx

После этого действия запустится сервер по умолчанию, сконфигурированный в файле /etc/nginx/nginx.conf (Приложение А.1). При обращении к локальному IP адресу (127.0.0.1, localhost) будет доступна вступительная страница Nginx (см. Рисунок 23).



Рисунок 23 — Страница дефолтного сервера

* 1. Конфигурация виртуального сервера

Конфигурация виртуального сервера производится путем написания конфигурационного файла в директории /etc/nginx/nginx.conf. Прописываем в нём поля: server\_name для обозначения имени сервера, listen 80 для указания какой порт будем слушать, root для указания адреса директории с файлами сайта(корневой раздел - root) , index для определения html страницы сервера.

В случае правильности синтаксиса перезапускаем nginx следующей командой:

$ sudo systemctl reload nginx (либо restart, но не рекомендуется)

После указанных действий, добавляем имя нашего сервера в список серверов, который описан в файле /etc/hosts (см. Приложение А.2). Теперь, мы можем подключаться к виртуальному серверу (я разместил туда свой старый проект из школы кода) (см. Рисунок 24).

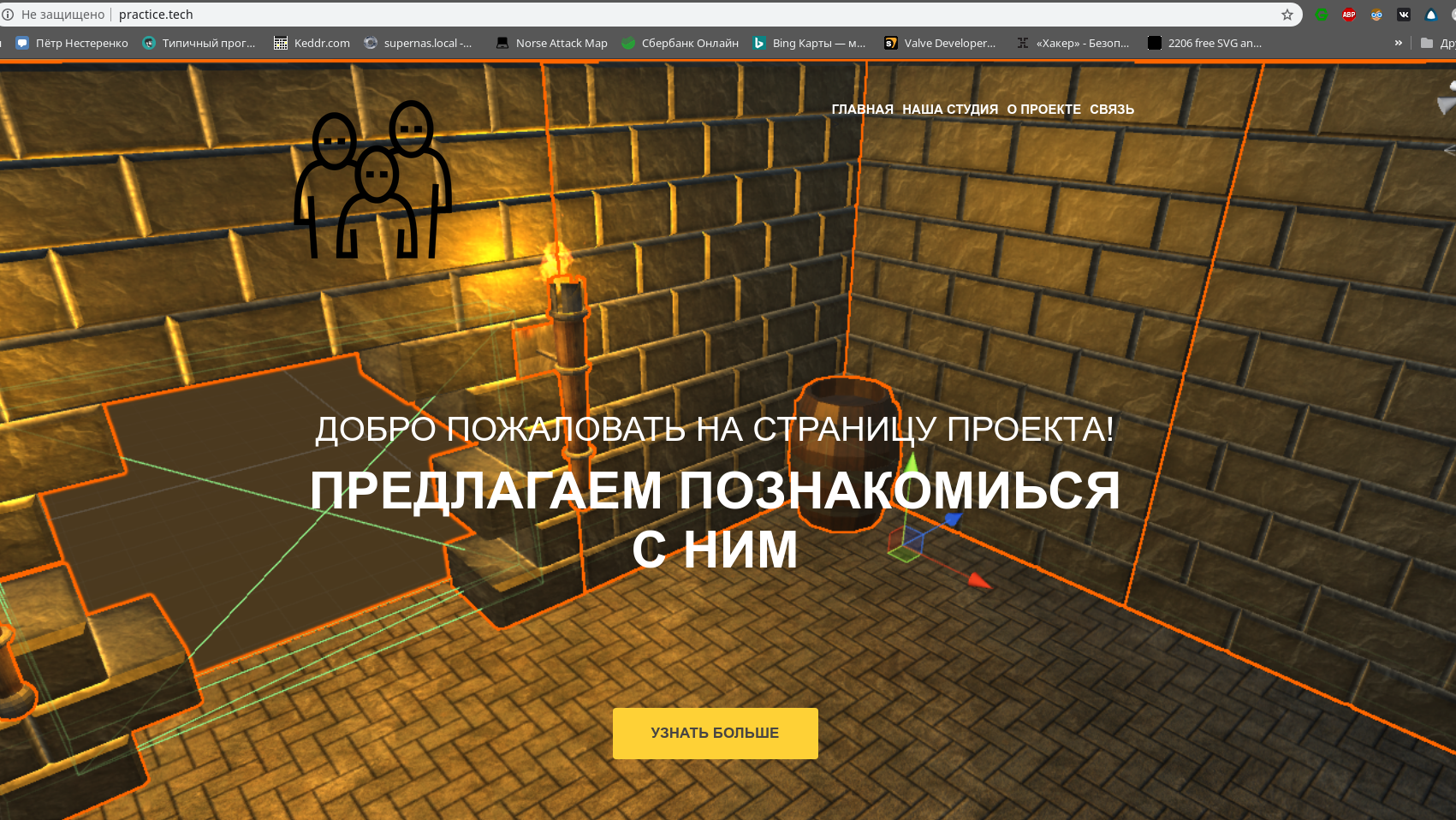


Рисунок 24 — Подключение к виртуальному серверу

1. Ссылки

Ссылка на репозиторий:

<https://gitlab.com/FriZIk/praktika>

Ссылка на видео с настройкой и тестами nginx:

<https://youtu.be/jlyDIm8cs9Q>

Заключение

В ходе практики была реализована установка системы контроля git и работа с ней, установка и настройка виртуального сервера nginx.

Для выполнения поставленной задачи использовалась операционная система GNU/Linux, система контроля версий Git, веб-сервер nginx.

В ходе выполнения задания:

- Узнал как использовать систему контроля версий Git

- Научился ветвлению проектов в Git при разработке ПО

- Познакомились со методами разрешения конфликтов, связанных со слиянием нескольких веток

- Узнал как настраивать виртуальный сервер на базе nginx

- Получил базовые навыки работы с Linux

Список использованных источников

1. Официальная документация по Nginx [Электронный Ресурс] URL: <https://nginx.org/ru/docs/>
2. Документация Git [Электронный Ресурс] URL: <https://git-scm.com/doc>

Приложение А

1. /etc/nginx/sites-abailable/default (изменённая часть)

# Default server configuration

#

user frizik http;

…

http

{

...

server

{

listen 80;

server\_name practice.tech;

Root /home/frizik/Site\_For\_Dungeon-master;

Index index.html;

}

…

* 1. **A.2 /etc/hosts**

127.0.0.1 localhost

127.0.0.1 practice.tech

127.0.1.1 frizik.localdomain frizik