МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Институт КНТ

Факультет ИСП

Кафедра ПИ им. Фельдмана

Лабораторная работа №2

по курсу: «Профессиональная практика программной инженерии»

по теме: «Работа с ветками»

Выполнил:

ст. гр. ПИ-21б

Калугин Д. В.

Проверил(а):

Ст. пр. каф. ПИ

Филипишин Д. А.

Ищенко А. П.

ДОНЕЦК – 2025

Цель работы: познакомиться с основами использования веток в системе контроля версий Git.

Задание к лабораторной работе: внести имитируемый программный код на любом из существующих языковпрограммированиядляразрабатываемогопрограммногопроектавсвойрепозиторий. Сгруппировать модули по тематике, описанные в лабораторной работе №1, добавить каждую из групп в свою ветку (не меньше 3х).

Описать в отчёте тематику группировки модулей, также отобразить созданные ветки. После добавления внести изменения в один из файлов, чтобы вызвать конфликт слияния. Разрешить конфликт одним из методов. Отобразить конфликт, метод его разрешения и итог в отчёте к лабораторной работе.

Ввиду используемых инструментов и решений, структура проекта будем иметь следующий вид:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 1 – Структура проекта

При разработке будем использовать для каждого модуля свою ветку. Так как каждый модуль может иметь как стабильные так и нестабильные подверсии. Также, как отдельную ветку выделим работу с документами.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 2 – Ветки проекта

Будем использовать методологию TDD (Test-Driver Development, разработка через тестирование), ввиду чего сначала пишем заголовочные файлы и тесты, а затем уже реализацию:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 3 – Изменения в ветке mod-ui

Однако, при добавлении новых тестов и дополнении в каждой из веток одного и того же файла CMakeLists.txt, возникает конфликт при слиянии:

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 4 – Конфликт слияния

В данном случае корректным разрешением будет слияние принятие всех изменений (комбинирование).

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 5 – История изменений после слияния всех веток в main

Отправим на сервер изменения в ветке main, а также все остальные ветки добавим на удаленный репозиторий.

Изображение выглядит как снимок экрана, текст

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 6 – Отправка веток на удаленный сервер

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 7 – Состояние главной страницы репозитория

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 8 – История изменений на удаленном репозитории

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 9 – Ветки удаленного репозитория

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 10 – История изменений одной взятой ветки mod-core

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.  
Рисунок 11 – История изменений одной взятой ветки mod-network