ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Факультет ИСП

Кафедра ПИ им Л.П. Фельдмана

Лабораторная работа №2

по курсу: «Профессиональная практика программной инженерии»

по теме: «Работа с ветками»

Выполнила:

ст. гр. ПИ-19б

Кулагина А.

Приняла:

ассистент каф. ПИ

Дмитрюк

Ищенко

ДОНЕЦК – 2023

*Лабораторная работа 2*

**Работа с ветками**

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

**Задание к лабораторной работе**

Разработать архитектуру в репозитории для разработки выбранного по заданию из 1 лабораторной работы проекта. Реализовать каждый из модулей в отдельных ветках, сделать не менее пяти фиксаций касательно разработки программного кода и после влить в ветку master. Зафиксировать лог ревизии для каждой из веток.

Удалить все созданные ветки и предоставить вывод команды git log для trunk после вливания всех веток.

Тема: 6. Переводчик, с реализацией автоматического определения языка из доступных в системе. Учесть и описать особенности отображения клиентской части для максимального удобства пользователя.

1.модуль работы с текстом

2.модуль языка

3.модуль клиентской части

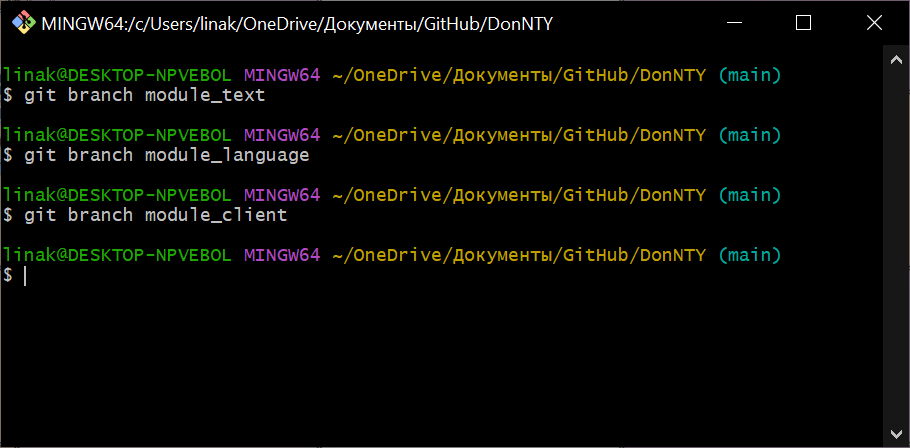
**Ход работы**

1. **Создаем ветки для каждого модуля программы с помощью команды git branch имя\_ветки**

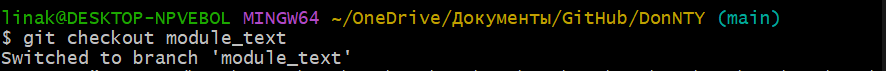
1.модуль работы с текстом – module\_text,

2.модуль языка – module\_language,

3.модуль клиентской части – module\_client

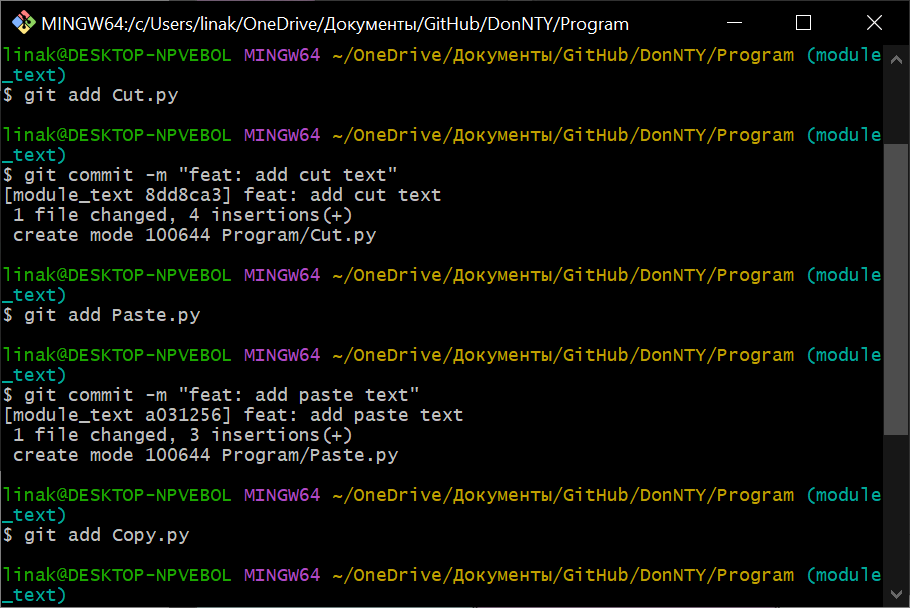


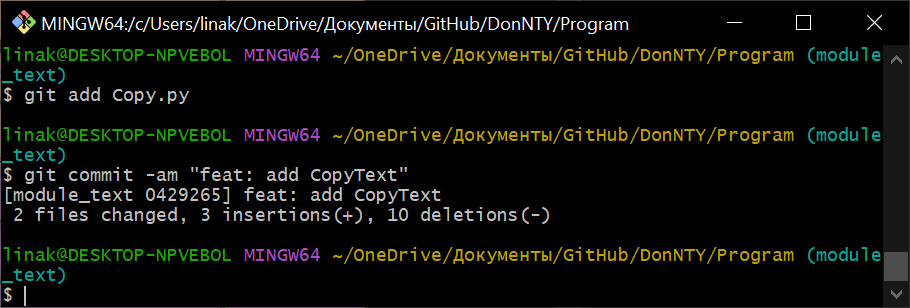
1. **Переключимся на ветку module\_text**



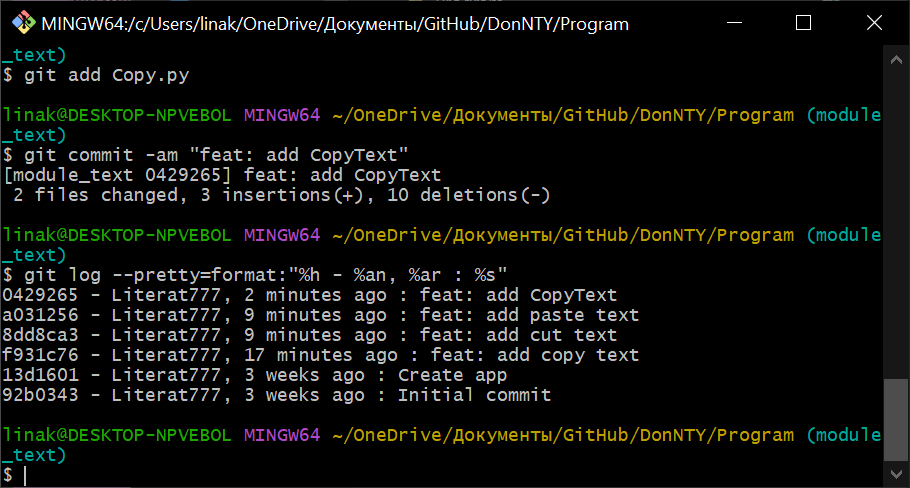
1. **Добавgitляем файлы с кодом и коммитим их**

(Копировать текст - Copy, Вставить текст - Paste, Вырезать текст -Cut)

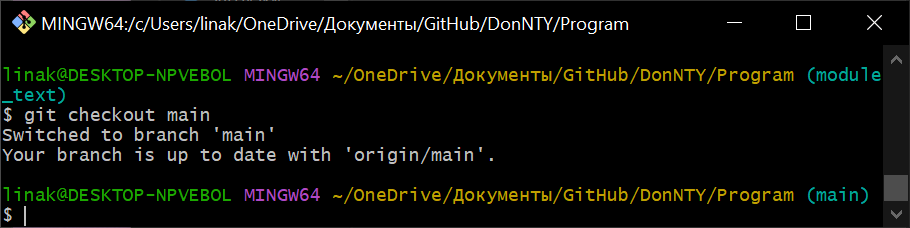




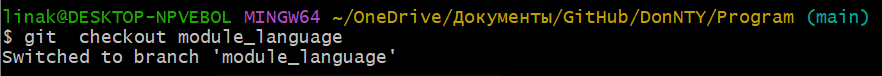
1. **Проверяем лог ревизии командой git log**

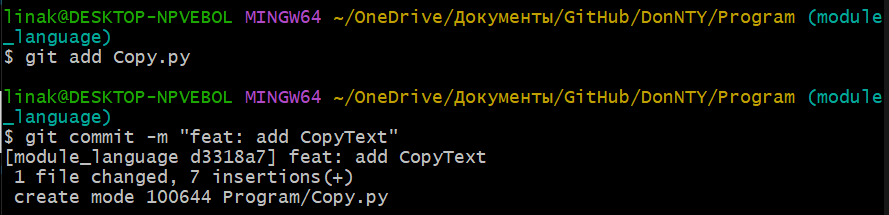


**5) Переключаемся на ветку main**

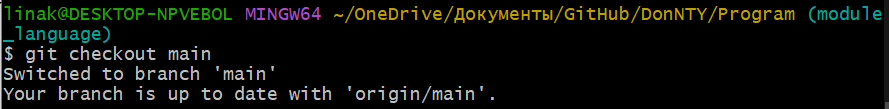


1. **Чтобы получить конфликт слияния, добавим в ветке module\_language файл с таким же названием как в ветке module\_text, но другим содержанием. Закоммитим**

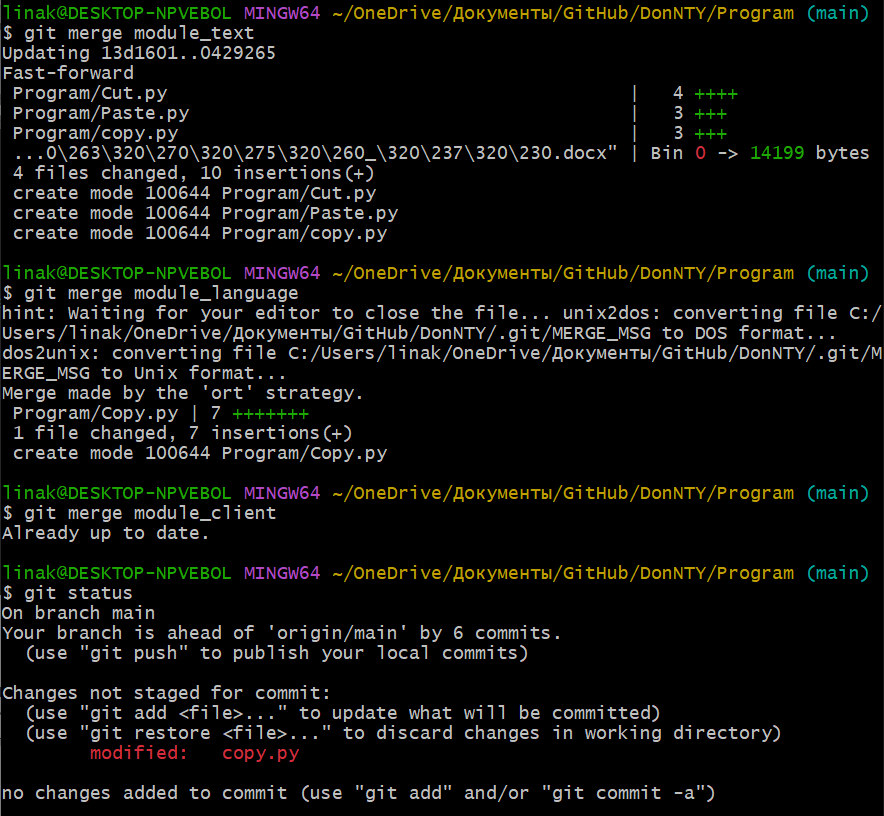




1. **Переключаемся на ветку мастера**



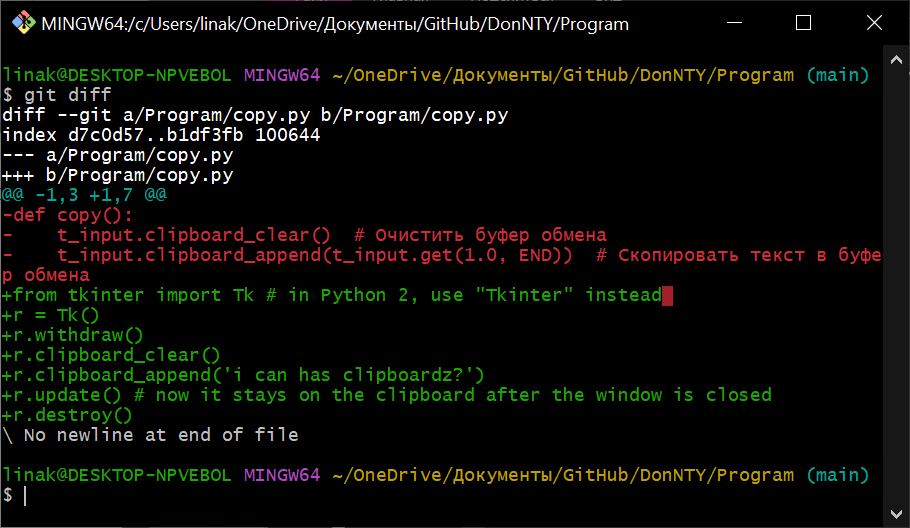
1. **Выполняем слияние всех веток и получаем конфликт слияния**



Конфликт слияния произошел из-за того, что

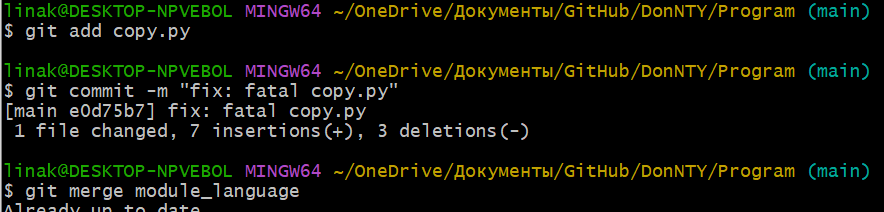
* в ветке module\_language в файле copy.py был код программы
* а в ветке module\_text в файле copy.py был код от другой программы.

После возникновения конфликта git «попытался объединить» два файла в один.

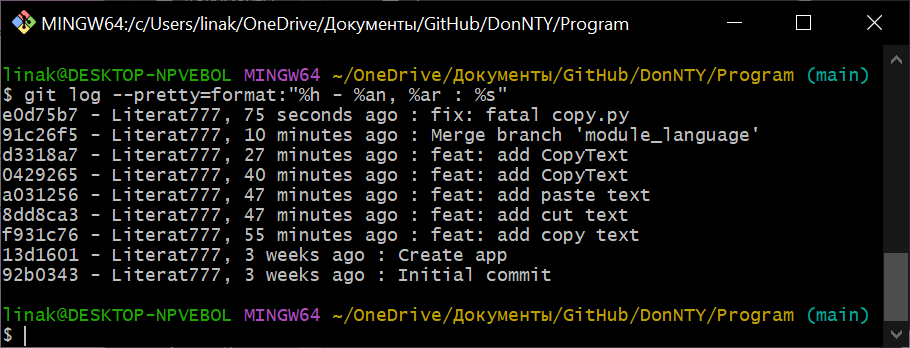


**8) Чтобы разрешить данный конфликт, достаточно отредактировать проблемный файл, выполнить команду git add file и закомитить.**

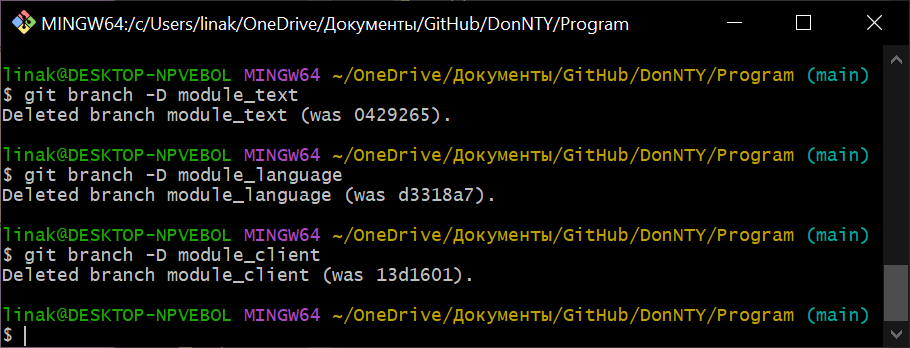
**После чего снова попробовать слить ветки**



**9) Лог ревизии**



**10) Удалим все ветки кроме основной**



**Вывод:** таким образом, в ходе лабораторной работы мы познакомились с основами использования веток в системе контроля версий Git, научились создавать и сливать их, намеренно создали конфликт слияния и разрешили его.