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

Леонова О.С.

Приняла:

ассистент каф. ПИ

Дмитрюк Т.Г.

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: 9. Векторный редактор, с реализацией не менее 20-ти программных средств доступных в CorelDraw Graphics Suite не ниже 2021 версии.

1.модуль работы с файлами

2.модуль текста

3.модуль графики

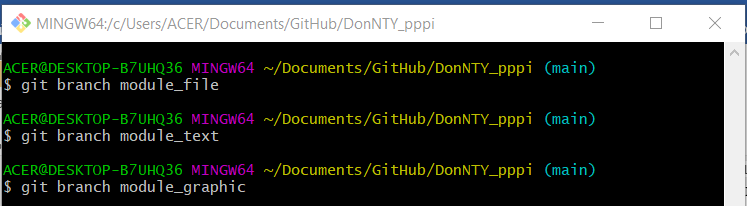
**Ход работы**

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

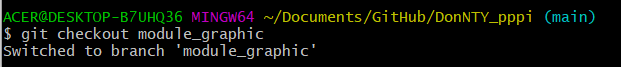
1.модуль работы с файлами

2.модуль текста

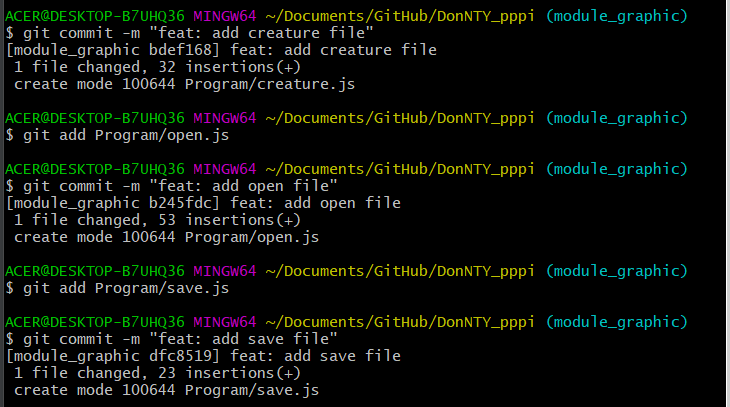
3.модуль графики

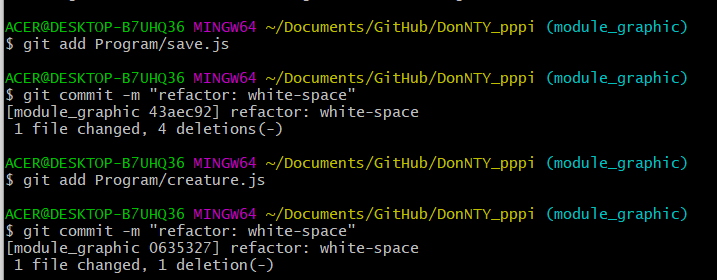
****

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

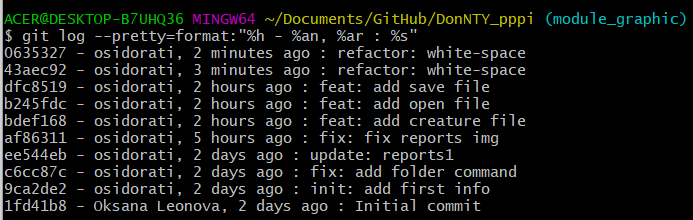
****

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

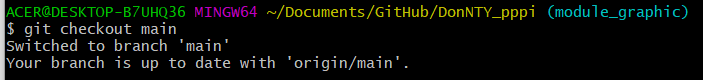
****

****

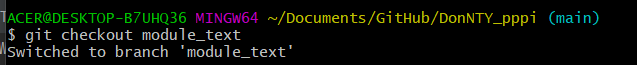
**4) Проверяем лог ревизии командой git log**

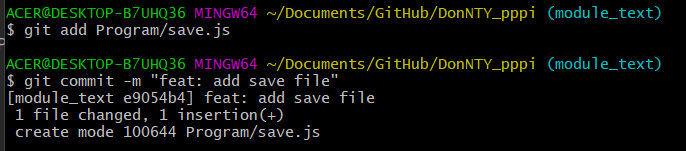
****

**5) Переключаемся на ветку main**

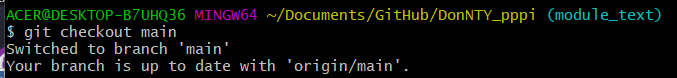
****

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

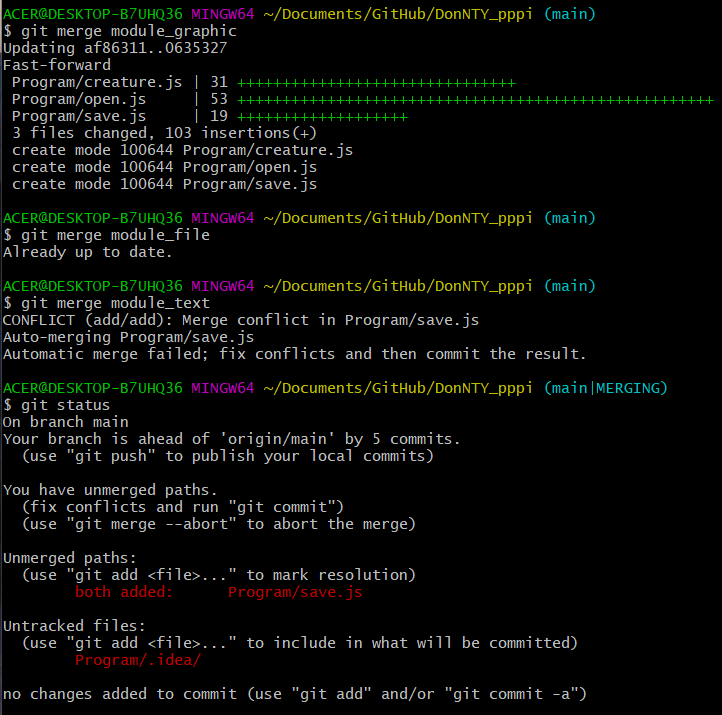
****

****

**7) Переключаемся на ветку мастера**

****

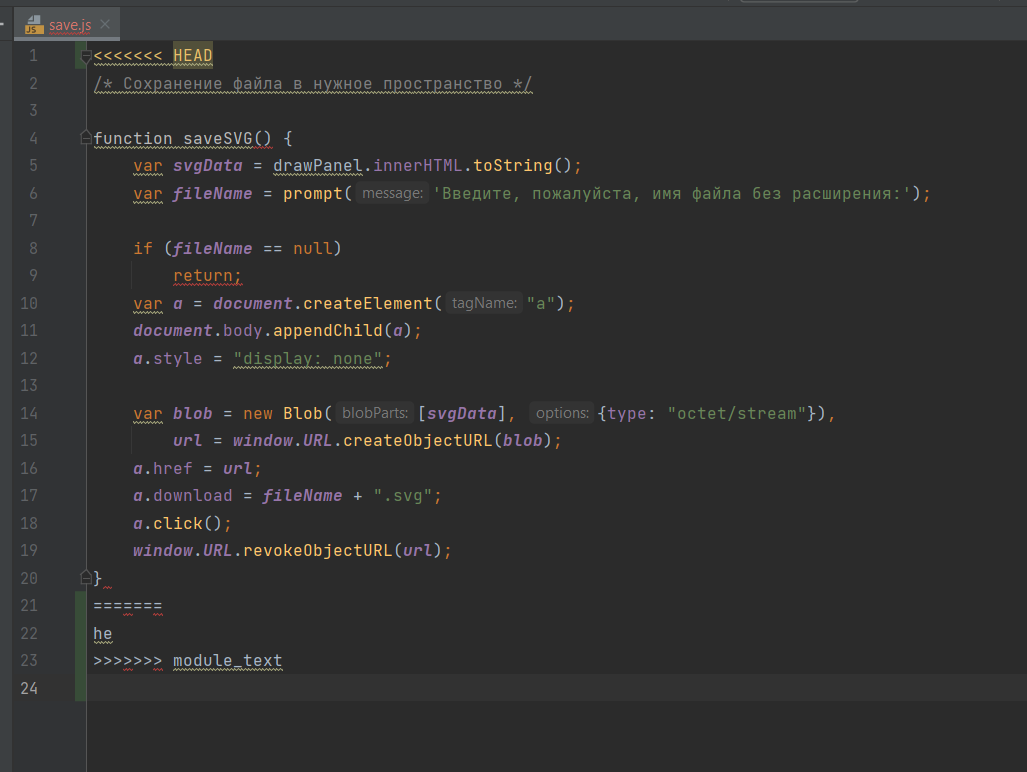
**8) Выполняем слияние всех веток и получаем конфликт слияния**

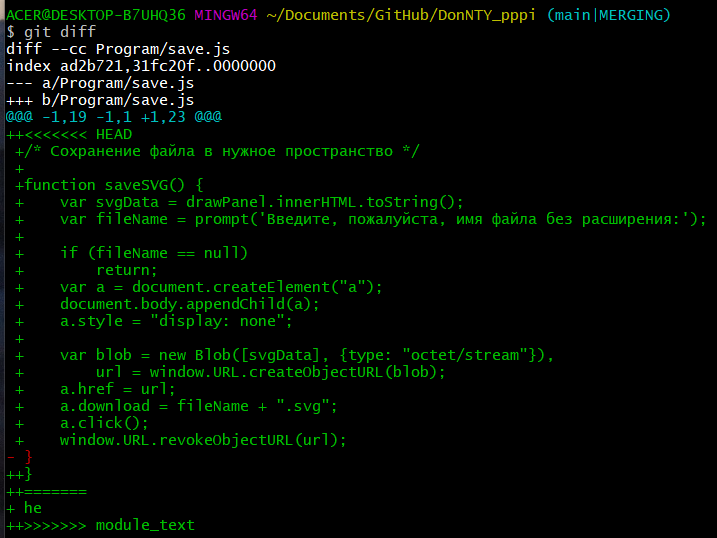
****

Конфликт слияния произошел из-за того, что

* в ветке module\_file в файле save.js был код программы
* а в ветке module\_text в файле save.js была просто строчка «he».

После возникновения конфликта git «попытался объединить» два файла в один. Строка ====== является центром или разделителем конфликта.

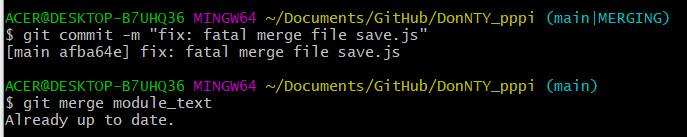
****

****

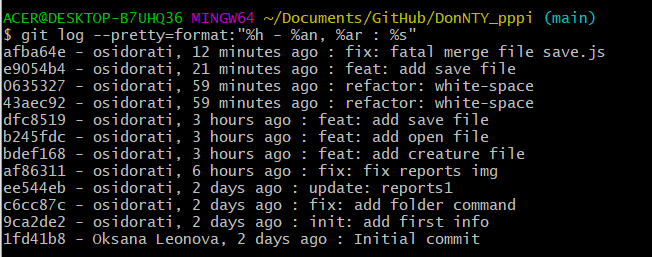
**8) Чтобы разрешить данный конфликт, достаточно отредактировать проблемный файл, выполнить команду git add file и закомитить.**

**После чего снова попробовать слить ветки**

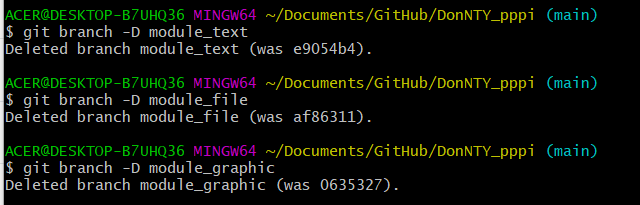
****

****

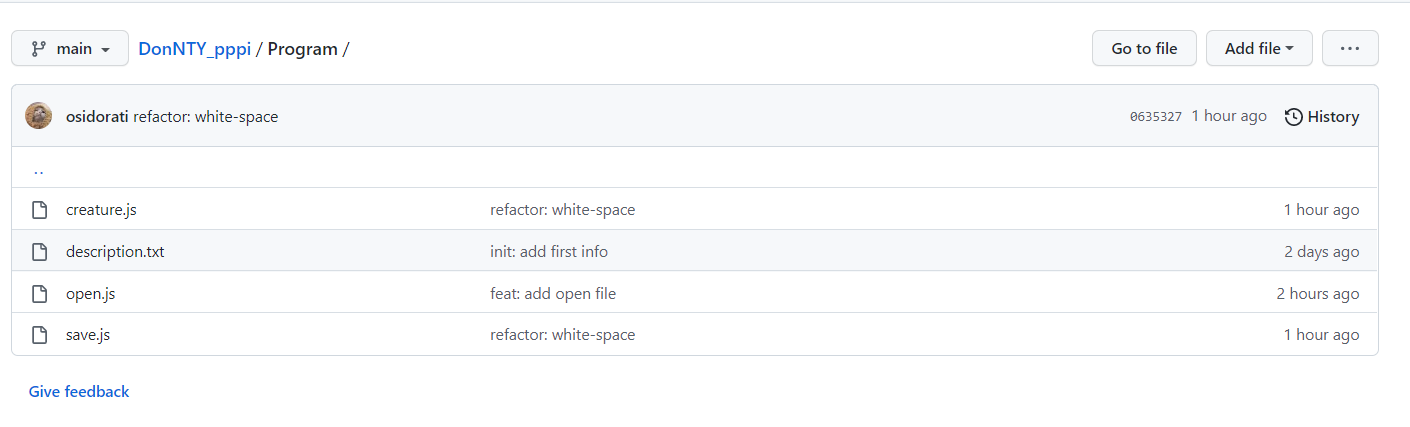
**9) Лог ревизии**

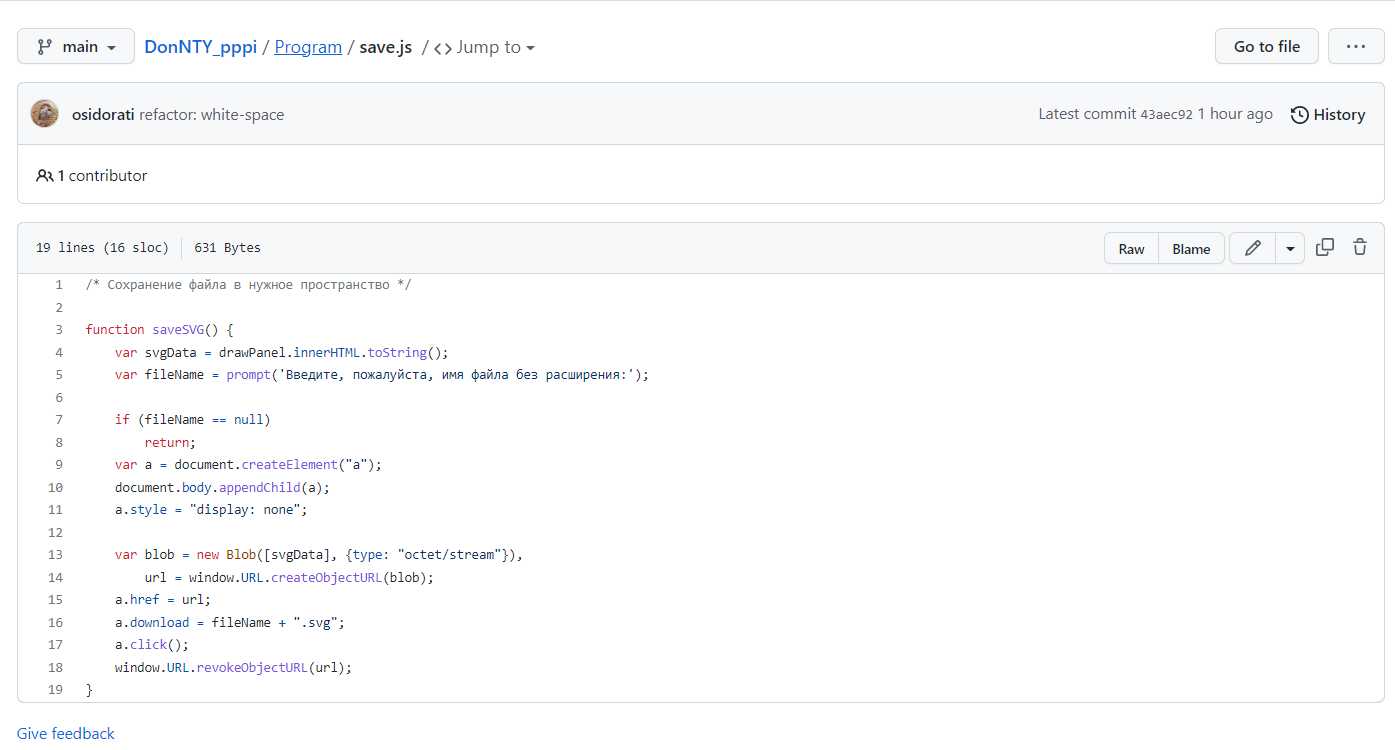
****

**10) Удалим все ветки кроме основной**

****

**В итоге, в репозитории в ветке main у нас находятся файлы, которые мы создавали в ветке module\_file**

****

****

**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.