КІТ-33 Стрєльцов Віталій Вікторович 04.05.2015 11.05.2015 \_\_\_\_\_\_ група ПІБ студента Дата виконання Дата прийому Оцінка

**ЗВІТ**

Лабораторна робота №3

Тема: Основи роботи з вітками в GIT. Основи злиття.

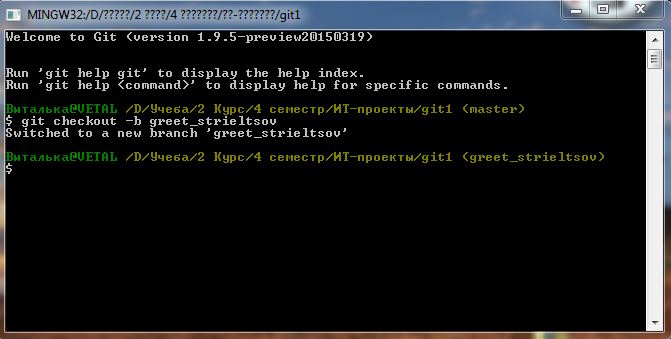
Розв’язування конфліктів.

Харків 2015

**Мета:** Познайомитись з основами роботи з вітками в Git. Освоїти процеси злиття віток та розв’язування конфліктів.

**Хід роботи**

1. **Створення гілок і їх злиття**
2. Створюємо робочий каталог на диску С з назвою «git». Зайдемо в цей каталог, клік правою кнопкою миші, натискаємо на «Git Init Here». Клікаємо павою кнопкою миші на каталозі і вибираємо «Git Bash». Відкриється консоль для роботи з Git.
3. Створюємо нову гілку greet\_strieltsov, використовуючи команду git checkout -b <branchname>.



1. Додаємо новий каталог lib\_strieltsov і файл greeter\_strieltsov.rb із вмістом:

class Greeter

  def initialize (who)

    who = who

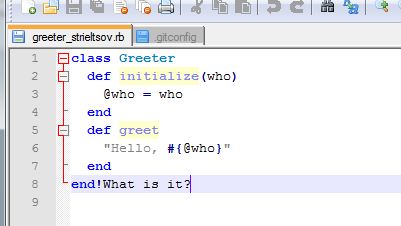
  end

  def greet

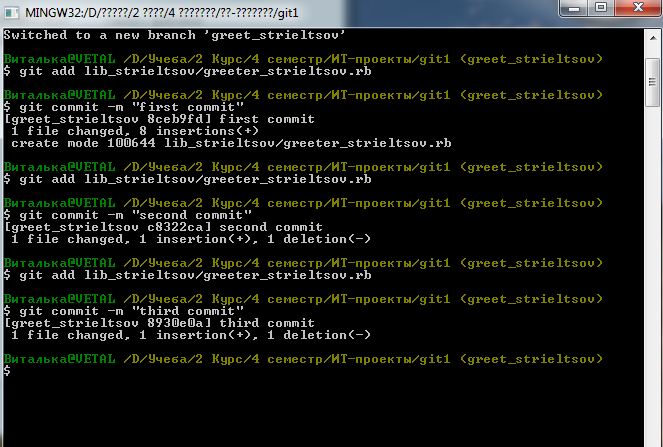
    "Hello, # {@ who}"

  end

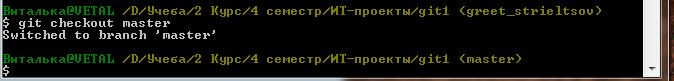
end



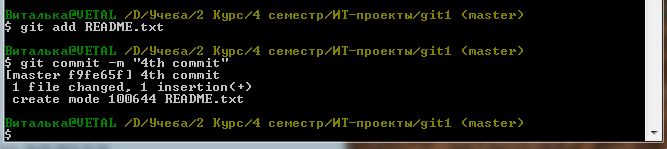
1. Зробимо 3 commits в гілку greet\_strieltsov.



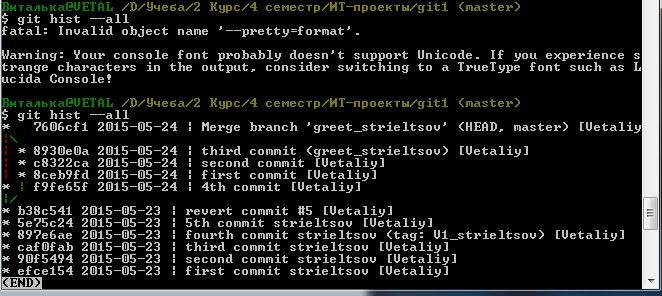
1. Продемонструємо перемикання на гілку master.



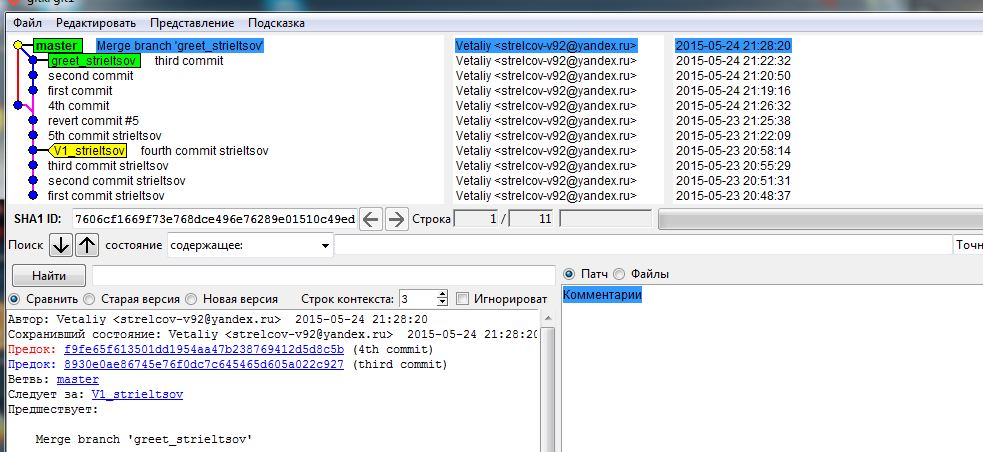
1. Створюємо файл README і зробимо commit в master.



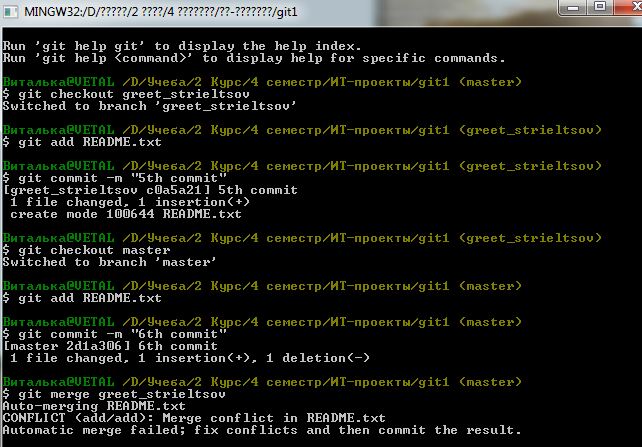
1. Проводимо злиття гілок greet і master - продемонструємо історію commits.



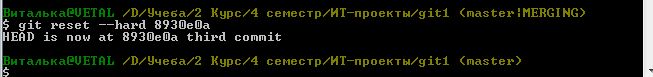
1. Зобразимо у графічному редакторі граф комітів, де вершинами графа будуть коди комітів.



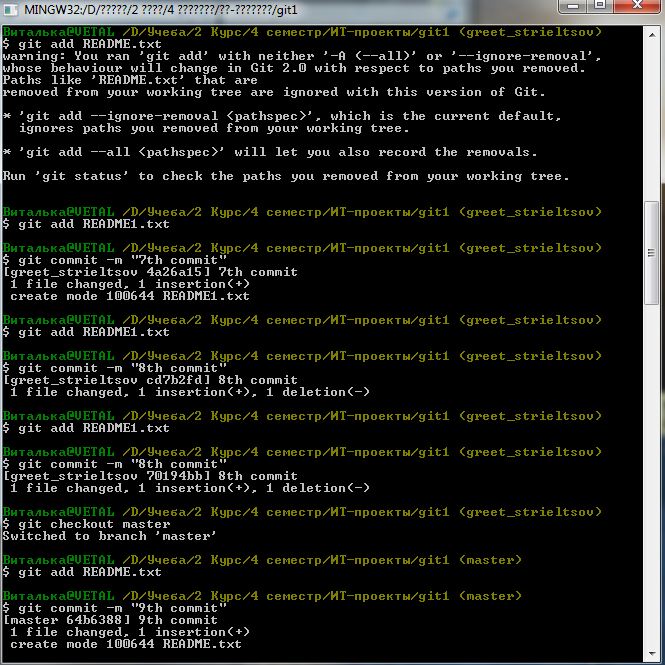
1. **Конфлікти**
2. Змоделюємо штучно конфлікт, при злитті описаному в попередній частині.

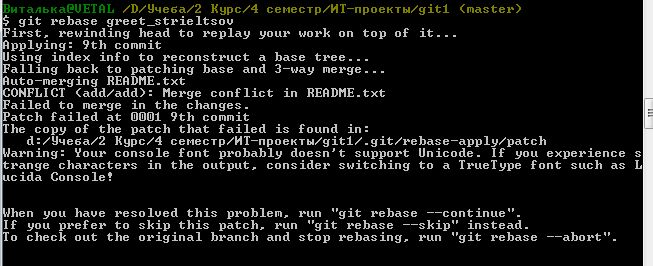


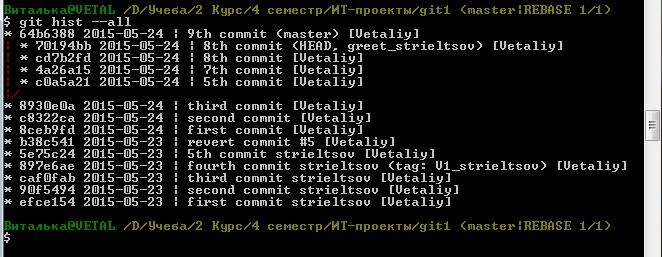
1. Використовуючи команду reset повернемо покажчик на стан гілок до їх злиття.

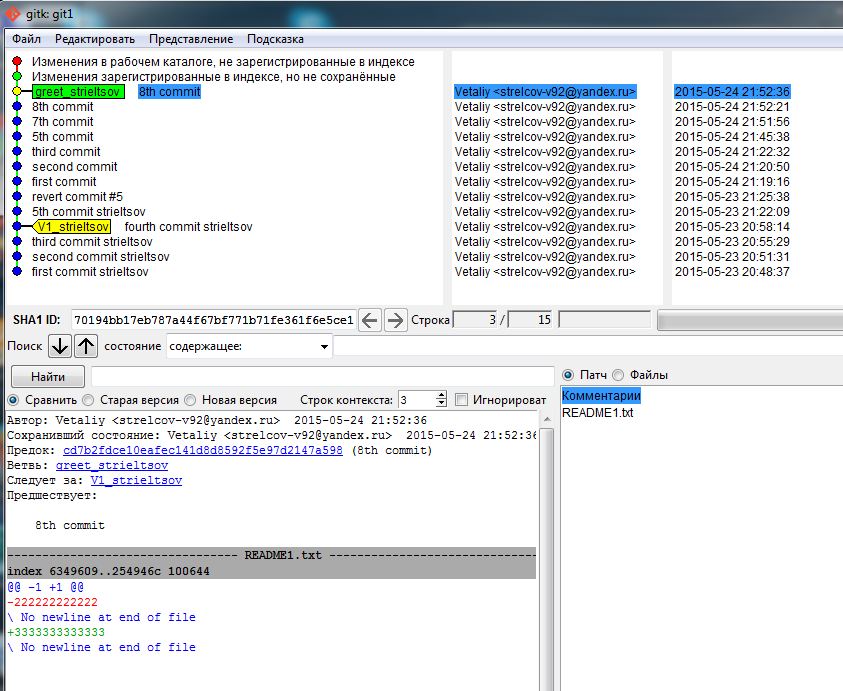


1. Виконаємо кроки 4-6 з частини 1. Виконаємо злиття змін командою rebase. Зобразимо у графічному редакторі граф комітів, де вершинами графа будуть коди комітів.



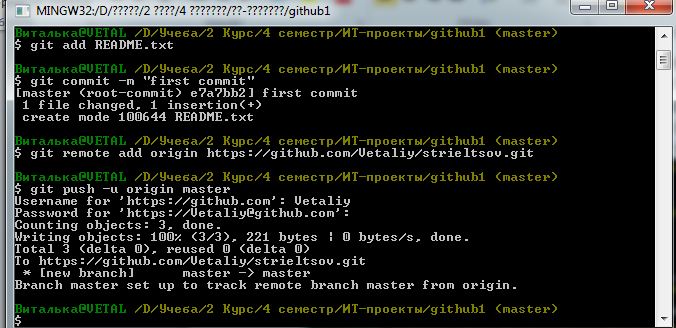




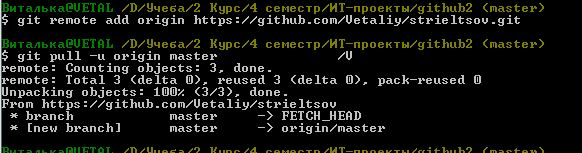


Як бачимо граф rebase має лінійну структуру, а граф merge – розгалужену.

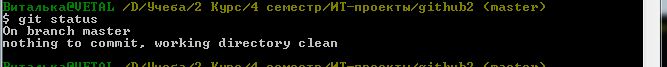
1. Для початку роботи зареєструємось на віддаленому сервісі github та ініціалізуємо віддалений репозиторій.
2. Створюємо перший робочий каталог на диску С з назвою «github». Зайдемо в цей каталог, клік правою кнопкою миші, натискаємо на «Git Init Here». Клікаємо павою кнопкою миші на каталозі і вибираємо «Git Bash». Відкриється консоль для роботи з Git. Створюємо там декілька файлів та заливаємо весь вміст папки у віддалений репозиторій.



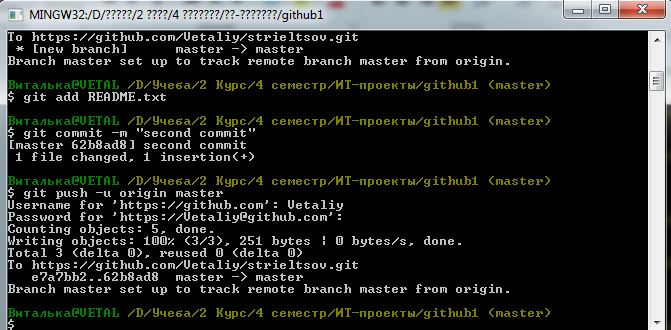
1. Створюємо другий робочий каталог на диску С з назвою «github2». Зайдемо в цей каталог, клік правою кнопкою миші, натискаємо на «Git Init Here». Клікаємо павою кнопкою миші на каталозі і вибираємо «Git Bash». Відкриється консоль для роботи з Git. Цей репозиторій «забирає» проект із віддаленого репозиторія до себе в локальний репозиторій.



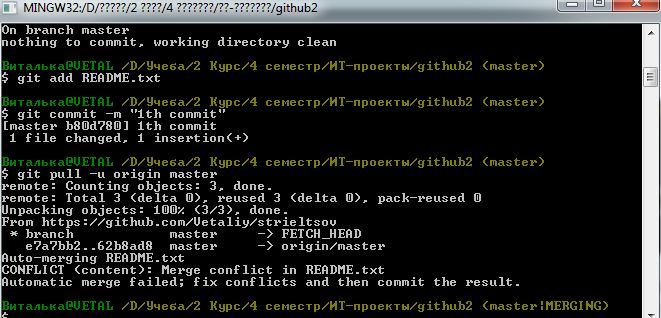
1. Виконаємо команду git status.



1. Проведемо у першому локальному репозиторії три коміти і «зальємо» зміни у віддалений репозиторій.



1. Проведемо у другому локальному репозиторії три коміти і «зальємо» зміни у віддалений репозиторій.



Як бачимо нам виводиться повідомлення про конфлікт.

Якщо ми зайдемо на віддалений сервіс github, то побачимо, що «залились» лише зміни проведені у першому локальному репозиторії. Отже, можна зробити висновок, що після виникненні конфлікту зміни не зберігаються.

**Висновки:** На даній лабораторній роботі я ознайомився з основами роботи з вітками в Git, а також освоїв процеси злиття віток та розв’язування конфліктів.