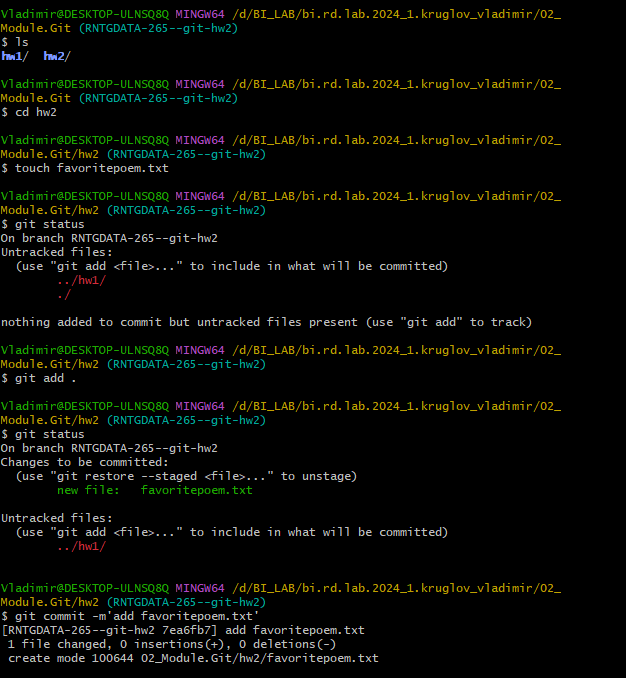
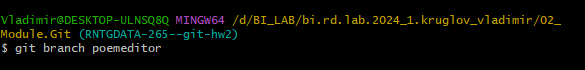
**Слияние (MERGE)**

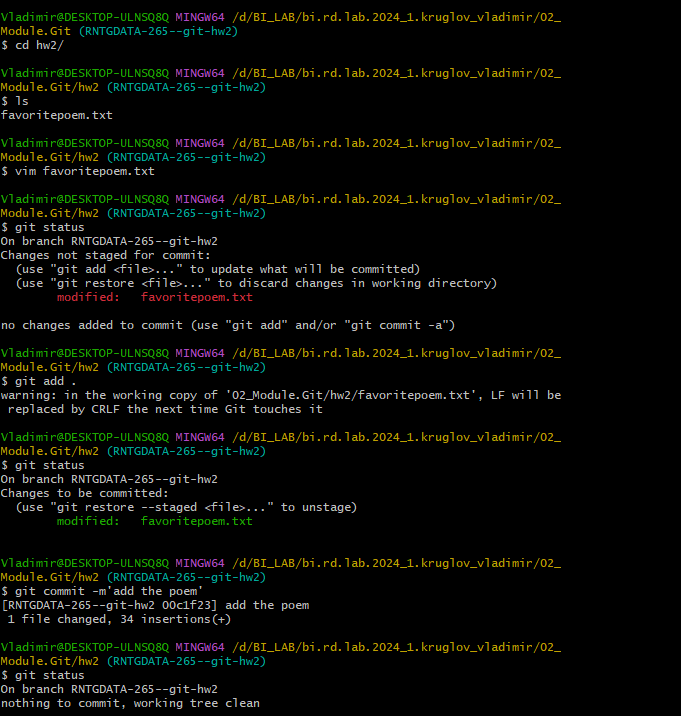
В папке «HW2» создайте файл “favoritepoem.txt”. Закоммитьте этот файл с подходящим комментарием.



Создайте ветку “poemeditor” – не переключайтесь (checkout/switch) на нее.



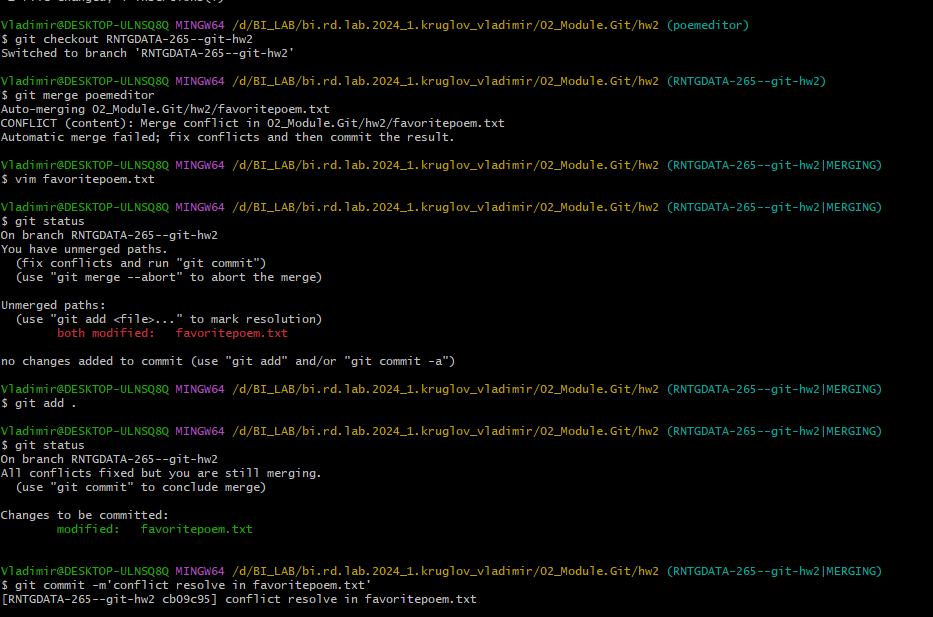
Измените файл “favoritepoem.txt” – добавьте первые две строки из своего любимого произведения. Закоммитьте этот файл с подходящим комментарием.

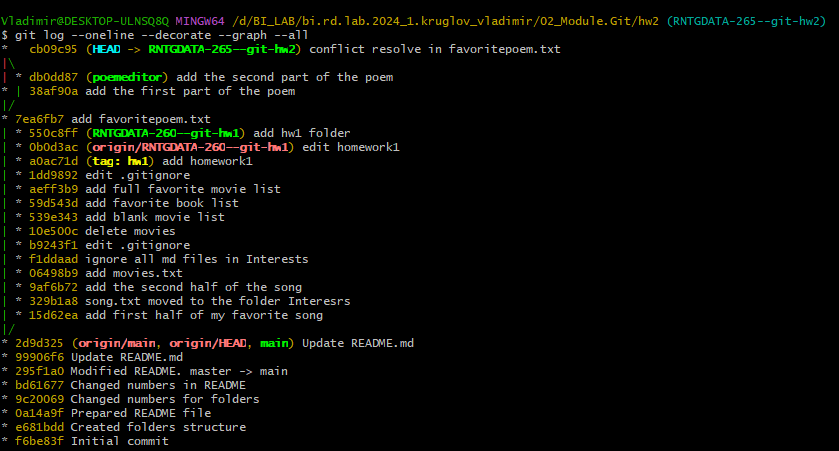


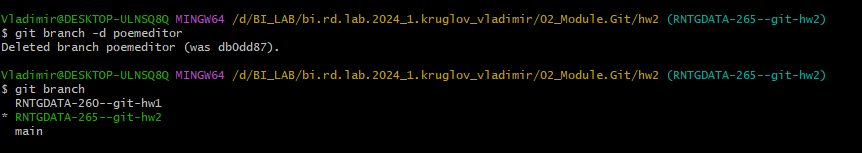
Переключитесь (checkout/switch) на ветку “poemeditor”. Измените файл “favoritepoem.txt”, добавив в него две другие строки из вашего любимого произведения и закоммитьте этот файл с подходящим комментарием.



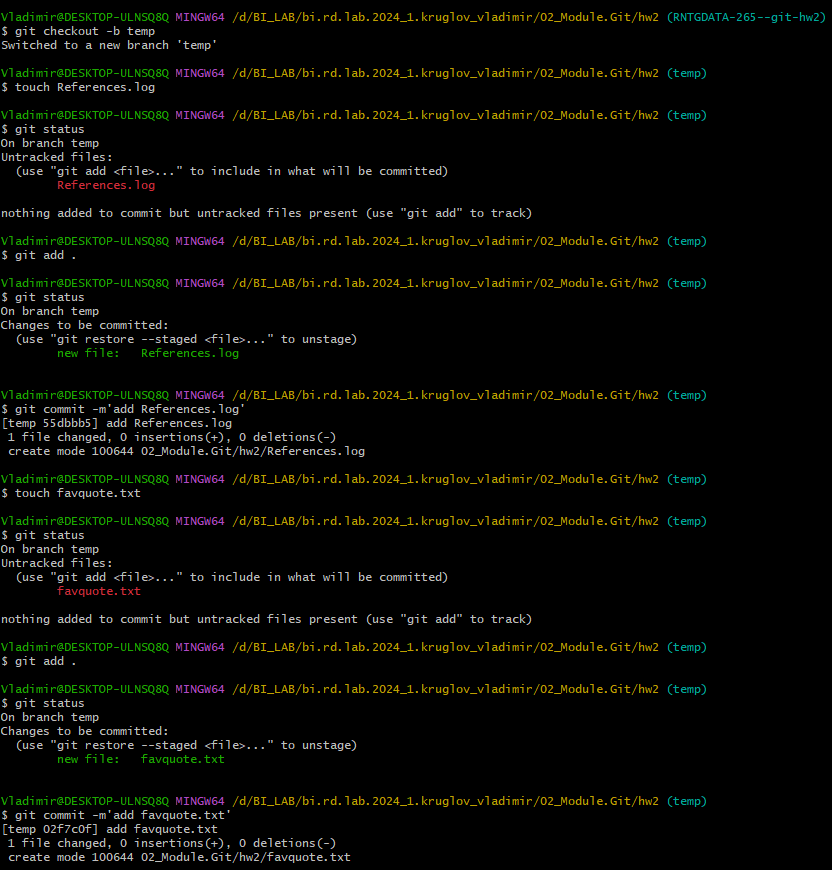
Объедините (merge) ветку "{номер задачи в JIRA}--git-hw2" с веткой “poemeditor”, разрешив merge-конфликт непосредственно в файле. Итоговый файл должен содержать все 4 строки. После того, как ветки будут успешно объединены, удалите “poemeditor”.





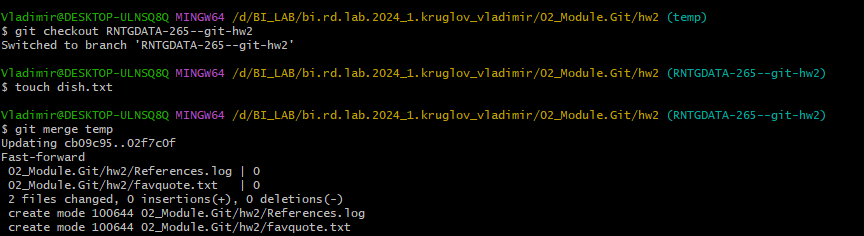


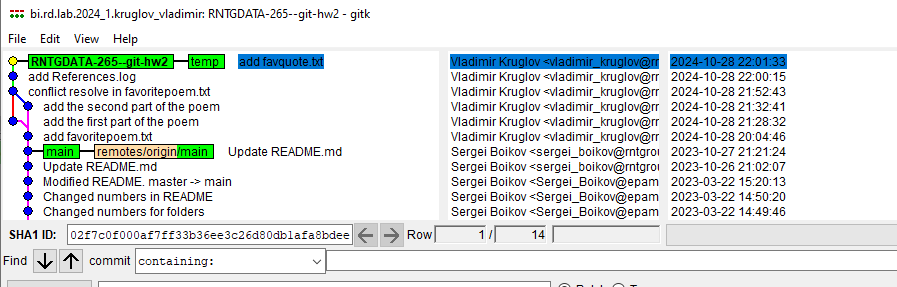
Переключитесь (checkout/switch) на новую ветку “temp” в одной команде вместе с ее созданием. Создайте новый файл “References.log” и закоммитьте его с подходящим комментарием. Создайте новый файл “favquote.txt”. Закоммитьте его с подходящим комментарием.



Переключитесь (checkout/switch) на ветку "{номер задачи в JIRA}--git-hw2" и создайте файл “dish.txt”.

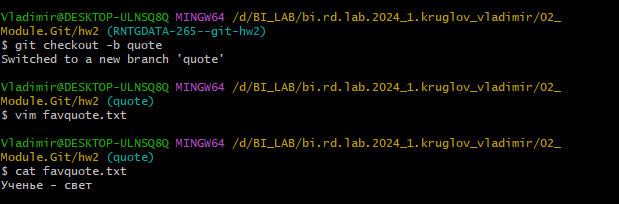
Объедините (merge) ветки "{номер задачи в JIRA}--git-hw2" и “temp”. Используйте Git GUI для показа истории всех веток и сделайте скриншот. После успешного объединения веток, удалите ветку “temp”.

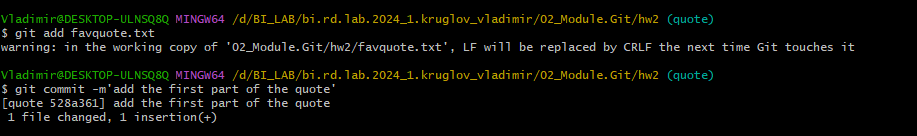




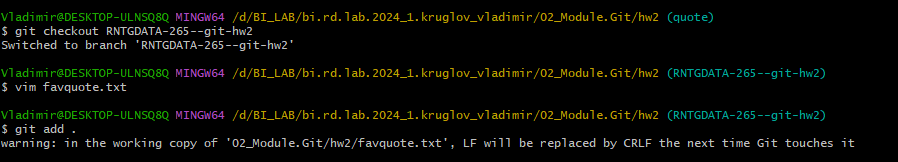


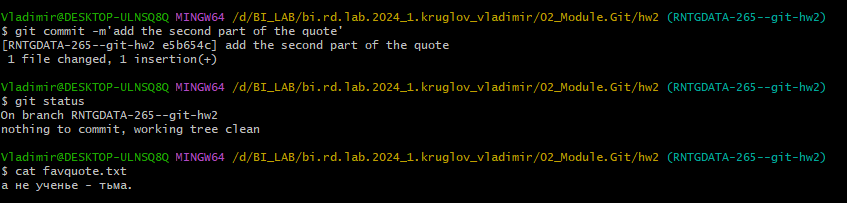
Переключитесь (checkout) на новую ветку “quote” в одной команде вместе с ее созданием. Отредактируйте файл “favquote.txt” – добавьте первую строку любимой цитаты. Закоммитьте с подходящим комментарием.





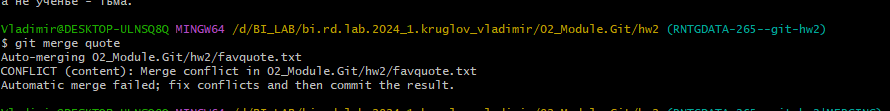
Переключитесь (checkout/switch) на ветку "{номер задачи в JIRA}--git-hw2". Измените файл “favquote.txt” – добавьте вторую строку любимой цитаты. Закоммитьте с подходящим комментарием.





Как вы можете догадаться здесь возникнет конфликт, на следующем шаге вы будете его решать используя git mergetool – на текущем же шаге вам нужно изучить существующие git mergetools и выбрать один из них. Здесь оставьте описание выбранного инструмента и настройте git для его использования (через git config).

Объедините (merge) ветки "{номер задачи в JIRA}--git-hw2" и “quote”, для разрешения merge конфликта используйте команду git mergetool. Ожидаемое решение: Цитата содержит обе строки в правильной последовательности. После успешного слияния удалите ветку “quote”.

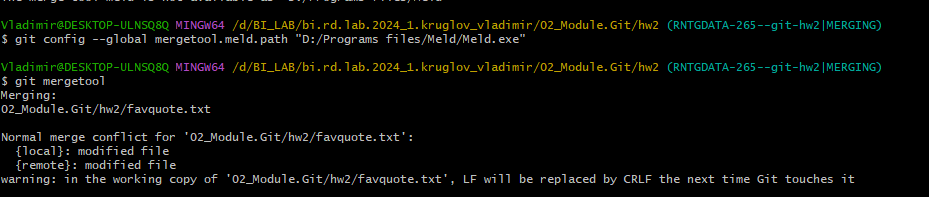


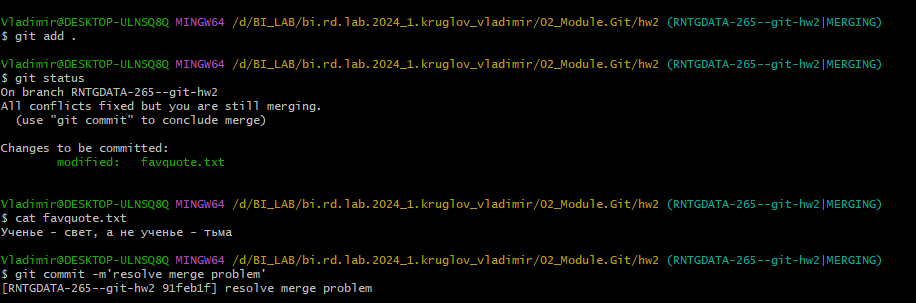
**Meld** — **компьютерная программа для сравнения содержимого текстовых файлов или каталогов**.  Это что-то вроде утилиты diff, но в графическом виде с возможностью «сливать» изменения, удалять и восстанавливать файлы.

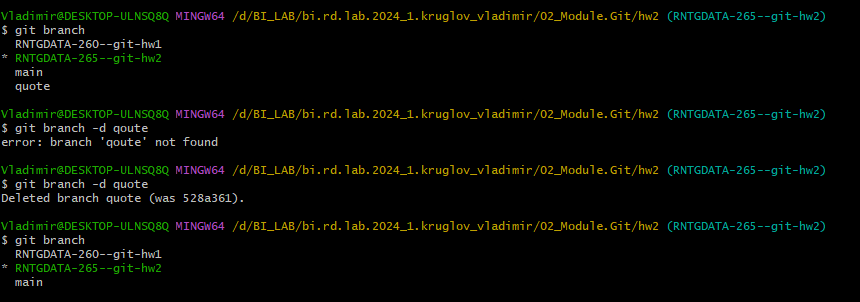
**Некоторые функции программы**:

* сравнение двух-трёх файлов или каталогов;
* создание файлов правки с описанием различий между файлами;
* работа с системами управления версиями Git, Subversion, Mercurial, Bazaar и CVS;
* вкладочный интерфейс.



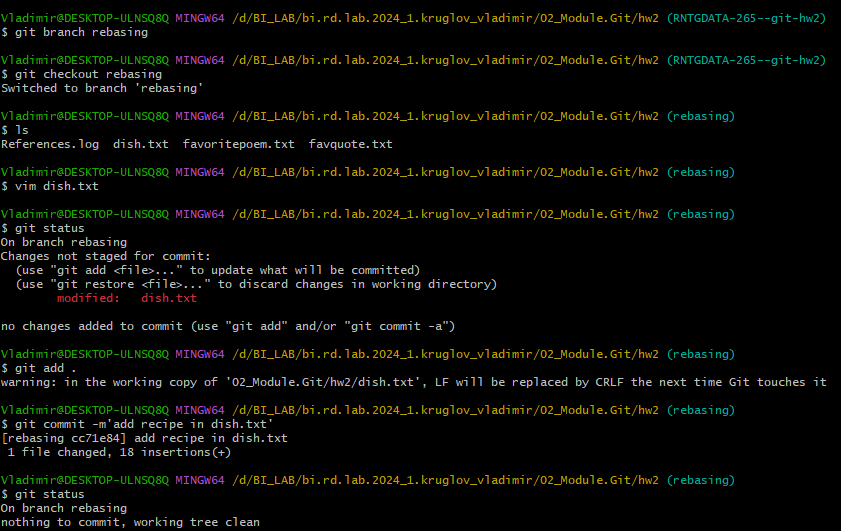


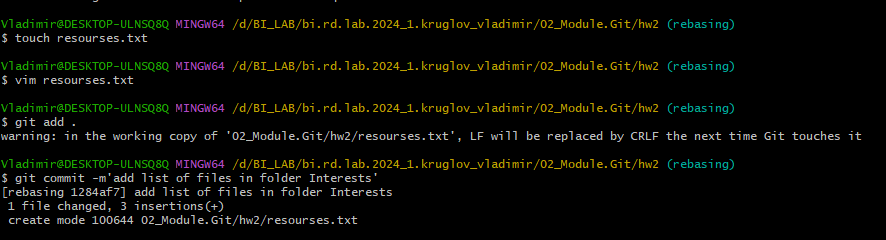




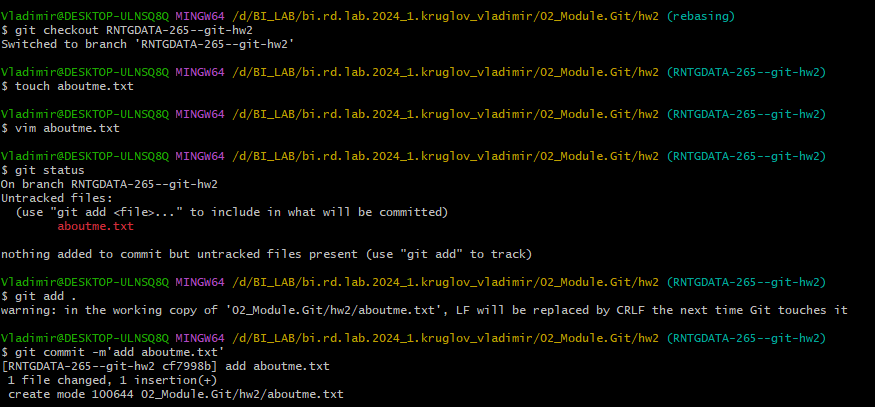
**Перебазирование (REBASE)**

Создайте новую ветку “rebasing” и переключитесь (checkout/switch) на нее. Добавьте в файл “dish.txt” название и ингредиенты вашего любимого блюда, закоммитьте изменения с подходящим комментарием. Измените файл “resources”, добавив в него информацию о вашей папке “Interests” (укажите в файле список файлов из папки “Interests”) и закоммитьте изменения с подходящим комментарием.

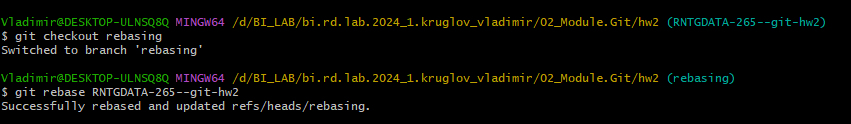


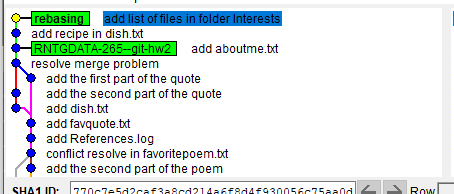


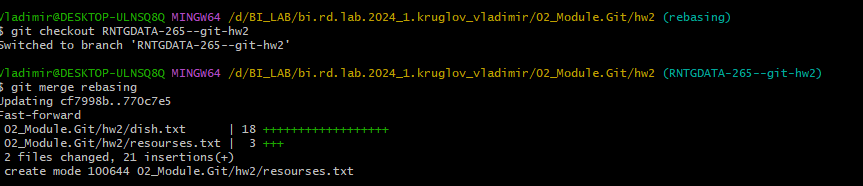
Переключитесь (checkout/switch) на ветку "{номер задачи в JIRA}--git-hw2" и создайте файл “aboutme.txt” и вставьте в него 2-3 предложения о себе. Закоммитьте файл и изменения, указав подходящий комментарий.

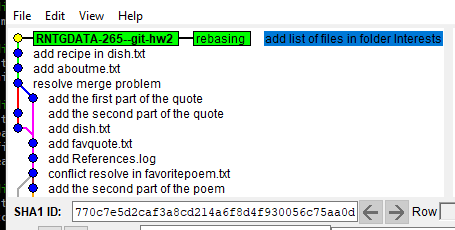


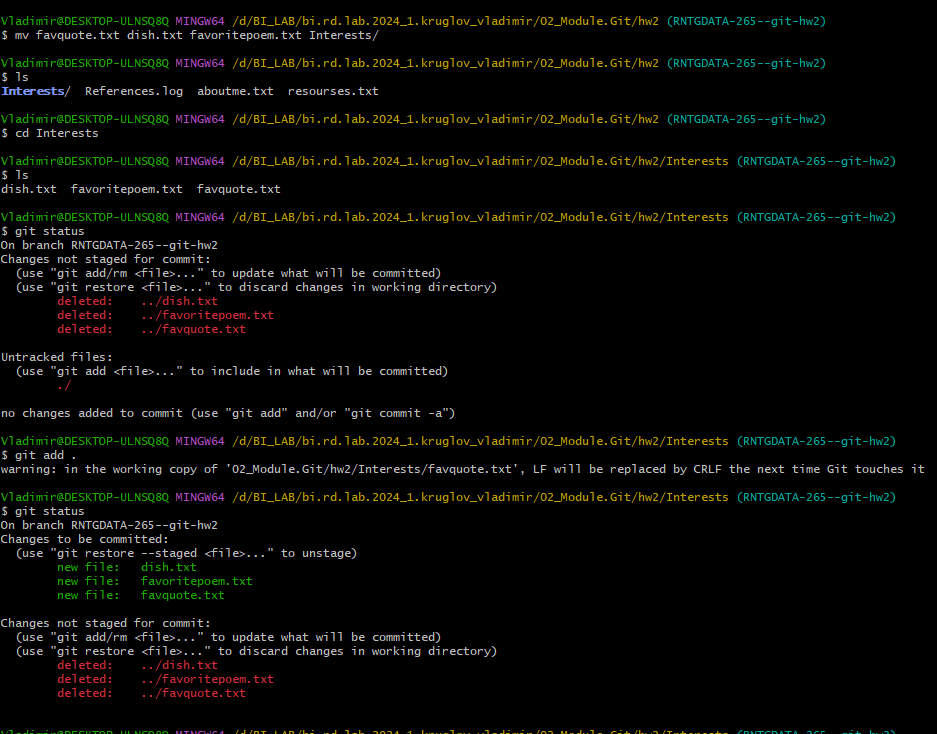
Переключитесь (checkout/switch) на ветку “rebasing” и перебазируйте (rebase) ее на ветку "{номер задачи в JIRA}--git-hw2". Используйте Git GUI для просмотра истории всех веток и сделайте скриншот лога. Объедините (merge) ветки "{номер задачи в JIRA}--git-hw2" и “rebase” после перебазирования. Используйте Git GUI для просмотра истории всех веток и сделайте скриншот лога. Переместите файлы “favquote.txt”, “dish.txt” и “favoritepoem.txt” в папку “Interests” и закоммитьте изменения с подходящим комментарием.

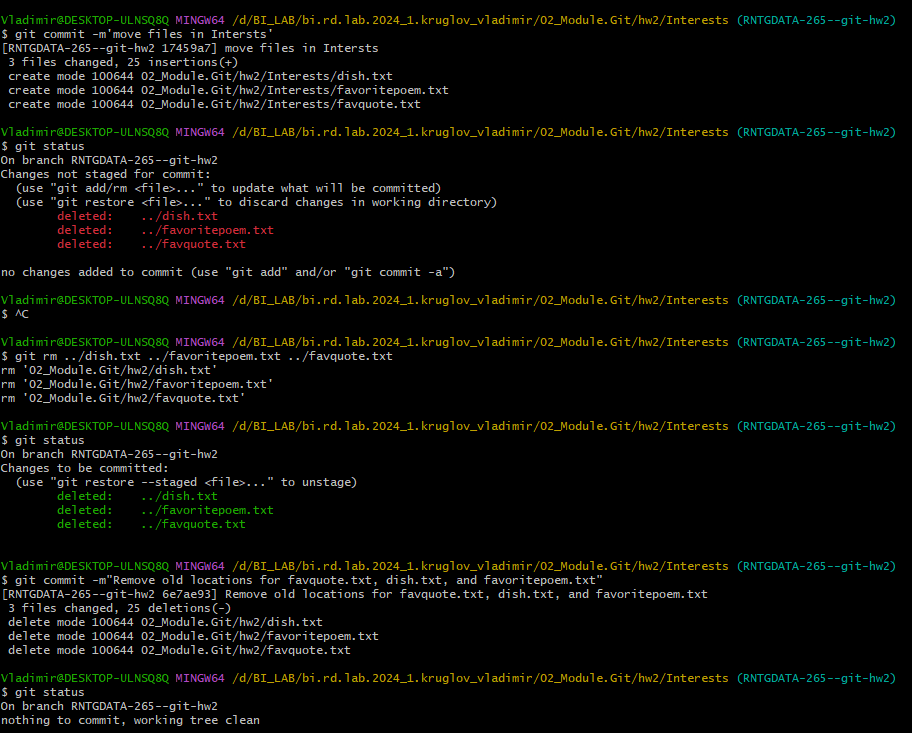






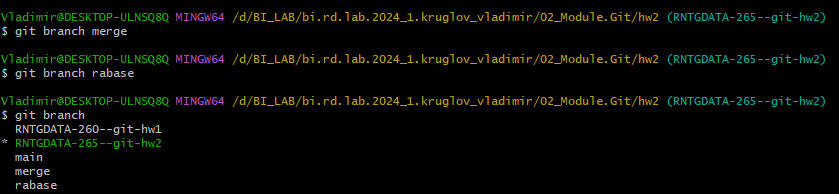




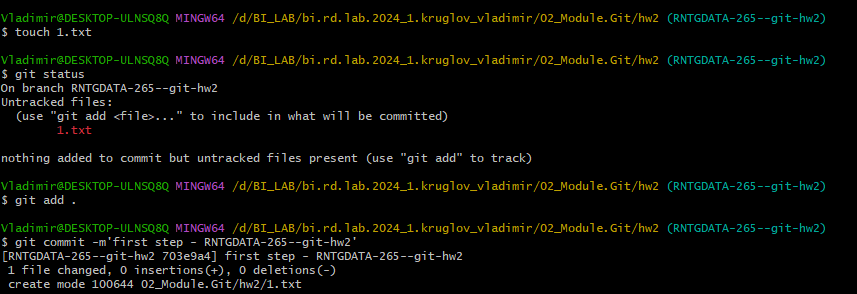


**WAY OF COMMIT**

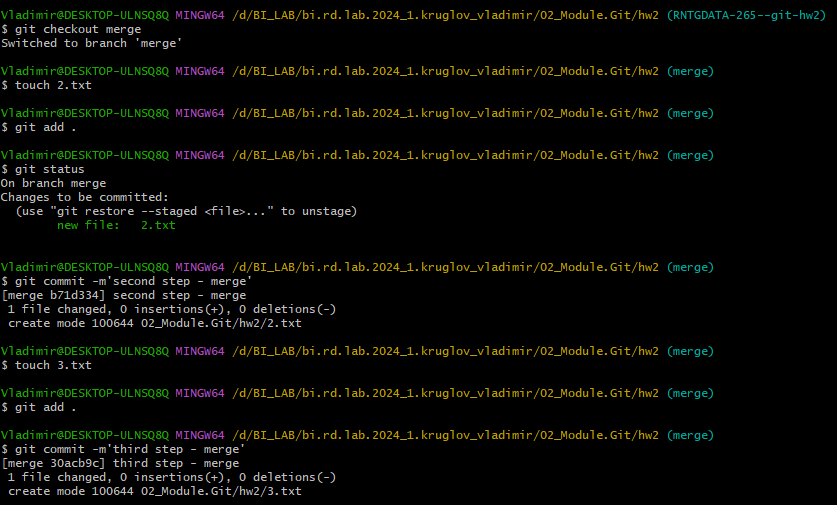
Перейдите в папку «HW2» локального гит репозитория. Создайте ветку merge, но не переключайтесь (checkout/switch) на нее. Создайте вторую ветку “rebase”, но не переключайтесь (checkout/switch) на нее.



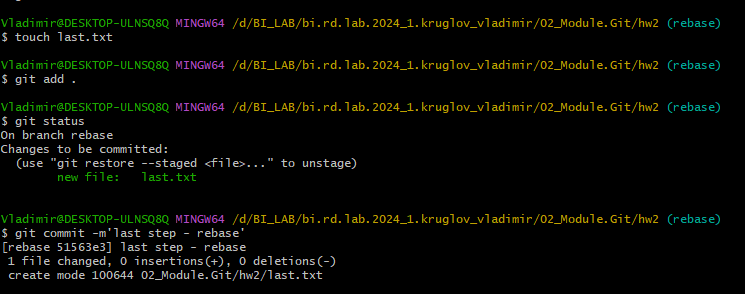
Создайте файл “1.txt” и закоммитьте его с комментарием “first step - {номер задачи в JIRA}--git-hw2” в ветку "{номер задачи в JIRA}--git-hw2".



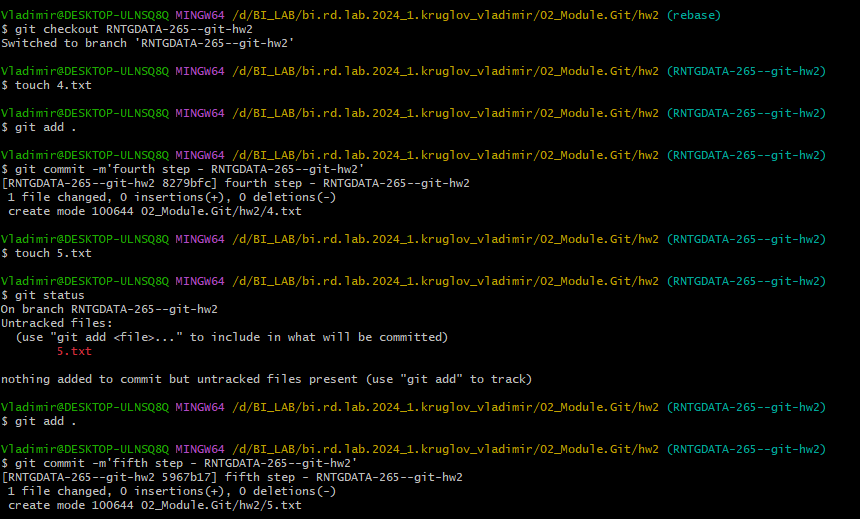
Переключитесь (checkout/switch) на ветку “merge”. Создайте файл “2.txt” и закоммитьте его с комментарием “second step – merge” в ветке “merge”. Создайте файл “3.txt” и закоммитьте его с комментарием “third step – merge” в ветку “merge”.

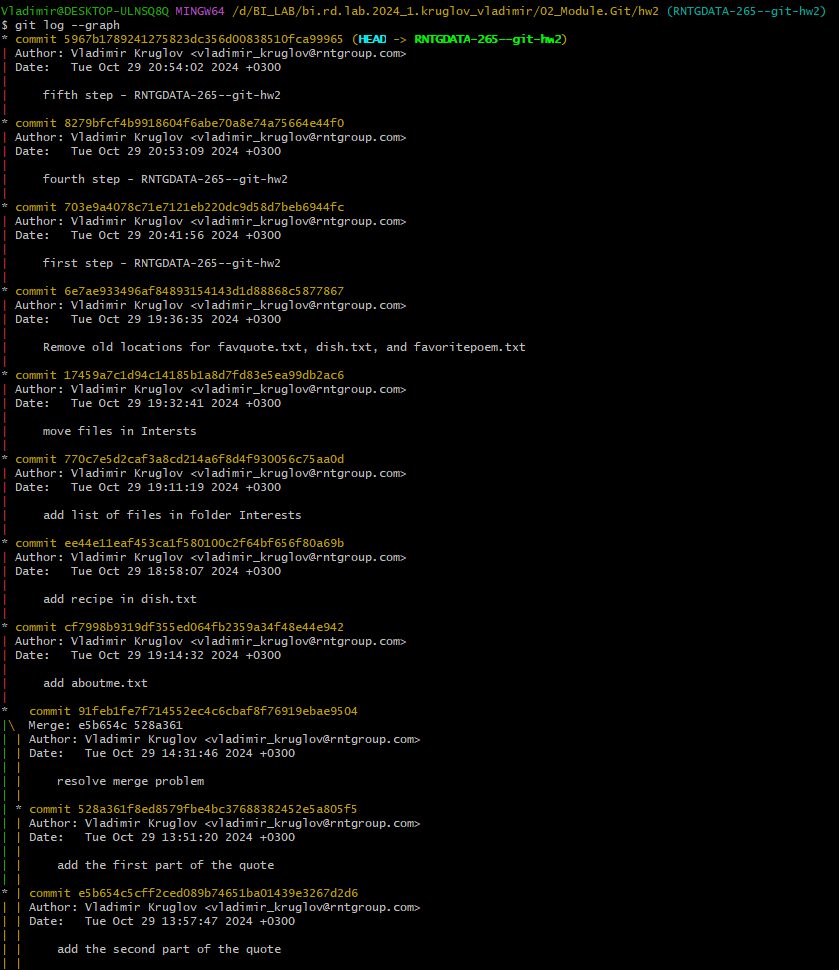


Переключитесь (checkout/switch) на ветку “rebase”. Создайте файл “last.txt” и закоммитьте его с комментарием “last step – rebase” в ветку “rebase”.

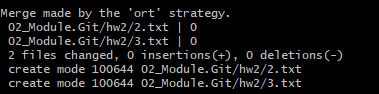


Переключитесь (checkout/switch) на ветку "{номер задачи в JIRA}--git-hw2". Создайте файл “4.txt” и закоммитьте его с комментарием “fourth step – {номер задачи в JIRA}--git-hw2” в ветку "{номер задачи в JIRA}--git-hw2". Создайте файл “5.txt” и закоммитьте его с комментарием “fifth step – {номер задачи в JIRA}--git-hw2” в ветку "{номер задачи в JIRA}--git-hw2". Используйте команду git log с параметром --graph или GIT Gui для просмотра дерева коммитов.





Объедините (merge) ветку “merge” с веткой "{номер задачи в JIRA}--git-hw2".



Перебазируйте (rebase) ветку “rebase” на верх ветки "{номер задачи в JIRA}--git-hw2". Не забудьте объединить (merge) ветку “rebase” с веткой "{номер задачи в JIRA}--git-hw2" после перебазирования. Используйте команду git log с параметром --graph или GIT Gui для просмотра дерева коммитов.

