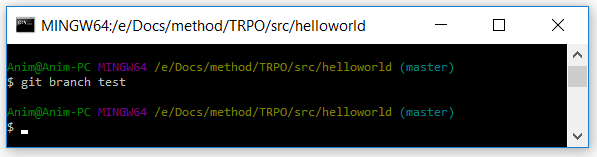
# Лабораторная работа №4:

**Основы работы с Git: ветвление**

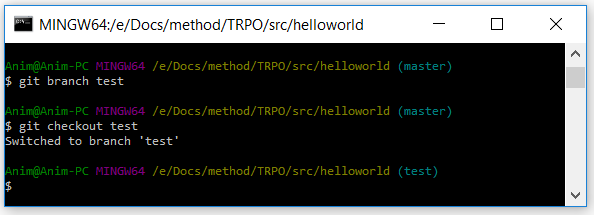
**Цель:**

Целью данной работы является получение базовых навыков работы с ветвлением в системе контроля версий git

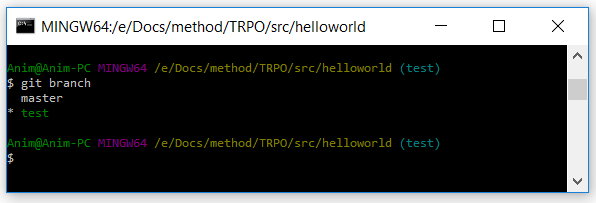
# Создание ветви:

Для **создания** ветви, используется команда: git branch <name>

Что бы **переключиться** на существующую ветвь, используется команда: git checkout

<name>

Для того, что бы посмотреть **список** ветвей, используется: git branch

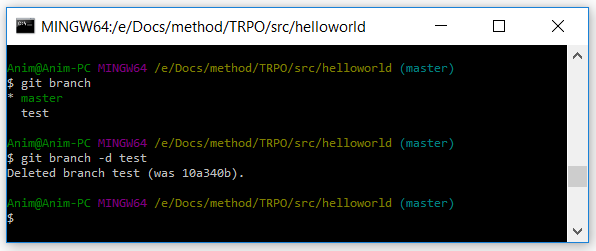


Теперь, если добавить какие-то изменения в ваш проект, а затем совершить commit, эти изменения будут доступны только в ветви test. Если переключиться на ветвь master, то в каталоге проекта будут находиться не изменённые файлы.

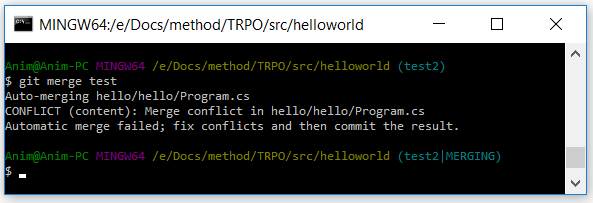
# Слияние ветвей:

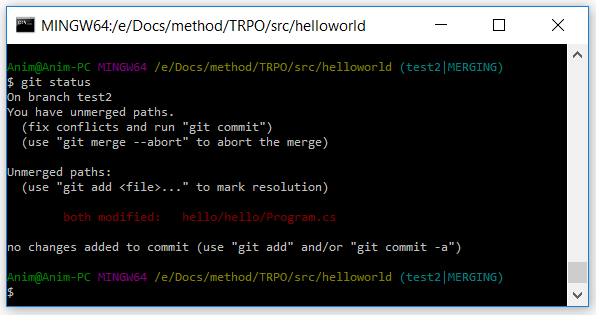
Для выполнения слияния ветвей, используется команда: git merge <name>

Поскольку изменения, внесённые в ветви тест, добавлены в ветвь master, ветвь test можно удалить, используя команду: git branch –d <name>

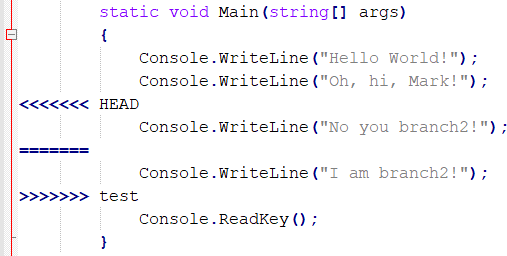


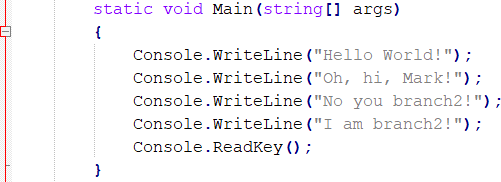
# Разрешение конфликтов:

В случае если существует две ветви, изменения в которых не могут быть объединены автоматически, команда merge выдаст следующее сообщение:

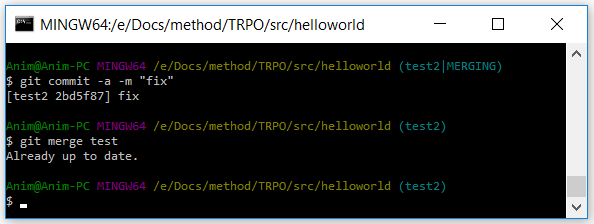
Для того, что бы определить, где именно происходит **конфликт** версий, можно использовать команду: git status

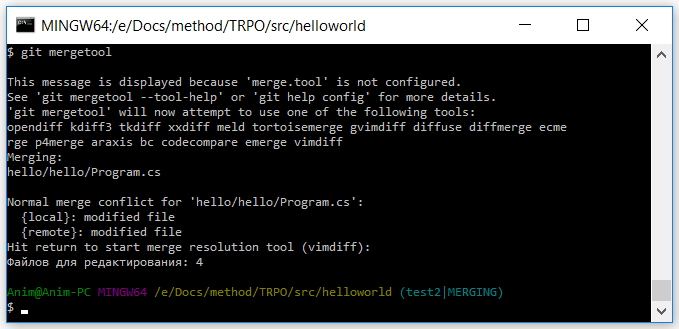
При этом в самом конфликтном файле будет содержаться **информация** о конфликте:



Можно разрешить конфликт вручную, **отредактировав** файл:

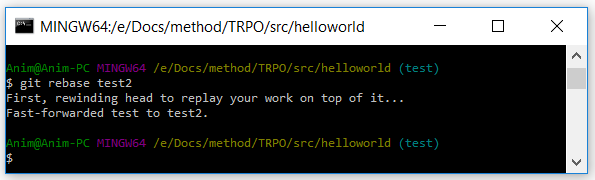
А затем, выполнить commit и merge:



Помимо этого, для слияния файлов можно использовать **графические инструменты**, для этого используется команда: git mergetool

В данном случае, в системе не установлены графические инструменты, поэтому предлагается открыть редактор vim (лучше не делайте этого, а если сделали – выйти можно набрав :qa! И нажав Enter)

# Перемещение ветвей:

Для того что бы применить изменения поверх существующих, используется команда: git rebase

Не рекомендуется использовать перемещение ветвей для тех коммитов, которые были загружены в общественный доступ.

# Задания:

1. Создать локальный репозиторий.
2. Создать в локальном репозитории интерфейс приложения согласно варианту задания.
3. Создать ветвь “Lab04” и загрузить в нее локальный репозиторий.

# Список литературы:

Более подробную информацию о git можно получить в книге:

Pro Git by Scott Chacon: <https://git-scm.com/book/ru/v2/>(на русском языке) Курс ТРПО за 2016 год: <https://github.com/evgeny-p/teaching>

Интерактивное обучение git: <https://try.github.io/levels/1/challenges/1>(на английском языке)