1. Слить изменения из develop-feature1 в develop

**Змінюємо поточну вітку develop-feature1 на вітку develop**

**$ git checkout develop**

Switched to branch 'develop'

Your branch is up-to-date with 'origin/develop'.

**Виконуємо злиття змінених файлів вітки develop-feature1 з файлами вітки develop**

**$ git merge develop-feature1**

Updating c9b4dcc..12e00ca

Fast-forward

.../Console\_GitTest/Console\_GitTest.csproj | 1 +

Console\_GitTest/Console\_GitTest/Program.cs | 34 ++++++++++++++++++++--

2 files changed, 33 insertions(+), 2 deletions(-)

**Виконуємо push змінених файлів вітки develop**

**$ git push origin develop**

Total 0 (delta 0), reused 0 (delta 0)

To https://github.com/vzayats/GitTest.git

c9b4dcc..12e00ca develop -> develop

1. Слить изменения develop в master и пометить полученную ревизию master тегом release1

**Змінюємо поточну вітку з develop на вітку master**

**$ git checkout master**

Switched to branch 'master'

Your branch is up-to-date with 'origin/master'.

**Виконуємо злиття файлів вітки develop з файлами вітки master**

**$ git merge develop**

Updating 7c5f321..12e00ca

Fast-forward

Console\_GitTest/Console\_GitTest.sln | 22 ++++++++

Console\_GitTest/Console\_GitTest/App.config | 6 +++

.../Console\_GitTest/Console\_GitTest.csproj | 59 ++++++++++++++++++++++

Console\_GitTest/Console\_GitTest/Program.cs | 57 +++++++++++++++++++++

.../Console\_GitTest/Properties/AssemblyInfo.cs | 36 +++++++++++++

5 files changed, 180 insertions(+)

create mode 100644 Console\_GitTest/Console\_GitTest.sln

create mode 100644 Console\_GitTest/Console\_GitTest/App.config

create mode 100644 Console\_GitTest/Console\_GitTest/Console\_GitTest.csproj

create mode 100644 Console\_GitTest/Console\_GitTest/Program.cs

create mode 100644 Console\_GitTest/Console\_GitTest/Properties/AssemblyInfo.cs

**Присвоюємо вітці тег release1**

**$ git tag release1**

**Виконуємо push файлів вітки master на GitHub**

**$ git push origin master**

Total 0 (delta 0), reused 0 (delta 0)

To https://github.com/vzayats/GitTest.git

7c5f321..12e00ca master -> master

**Виконуємо push тега release1 у вітку**

**$ git push origin release1**

Total 0 (delta 0), reused 0 (delta 0)

To https://github.com/vzayats/GitTest.git

\* [new tag] release1 -> release1

1. ***Работа с историей изменений****.* Написать команды:

1. которая выводит сообщения и автора всех коммитов из ветки develop-feature1, которые сделаны за последние 3 часа и которых нет в родительских ветках (develop, master). Сортировка по дате, первые более новые.

Виводимо в одну стрічку інформацію про коміти вітки develop-feature1 за останні 3 години, за виключенням віток master та develop, використовуючи формат стрічки виводу (автор коміту, його email та повідомлення), сортування по даті:

**$ git log --since="3 hours" develop-feature1 origin/master..develop --pretty=fo**

**rmat:"Author: %an (%ae), %s" --date-order**

1. выводит для каждого коммита из веток master, develop, в сообщении которых есть упоминание заданного номера задачи (допустим задачи 231), его сообщение, автора и дату. Сортировка по дате, первые более новые.

Виводимо в одну стрічку інформацію про коміт з маскою пошуку повідомлення, в якому міститься код “302” з віток master develop, сортування по даті, використовуючи формат стрічки виводу (повідомлення, автор коміту та дата створення):

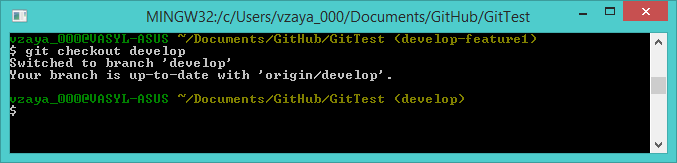
**$ git log --grep="302" --date-order master develop --pretty=format:"%s, %an, %cD"**

2. ***“Избирательное слияние”***.

Дано: После слияния ветки develop-feature1 в develop, вы продолжили работать в ветке develop-feature1, а develop отдали на тестирование, потом исправили ошибку и закоммитили исправление в develop-feature1, сделали push. Через какое-то время тим лид говорит, что это критический фикс, и надо включить его в develop. Вы делаете pull в ветке develop-feature1 и оказывается, что ваш коллега сделал еще 3 коммита после вашего, и они не стабильные, поэтому не должны быть включены в develop.   
Задание: как добавить в develop только ваш коммит, но чтобы 3 коммита после него в develop не попали?

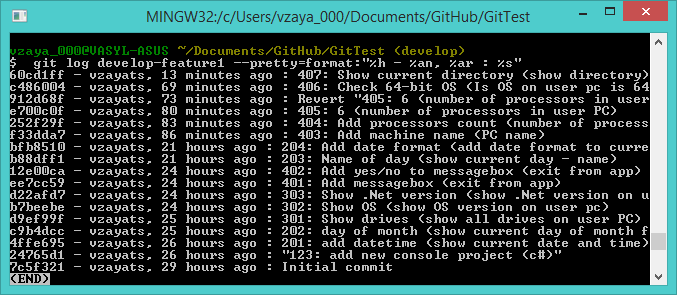
1. Переходимо у вітку develop, в яку ми будемо виконувати коміт:

**$ git checkout develop**



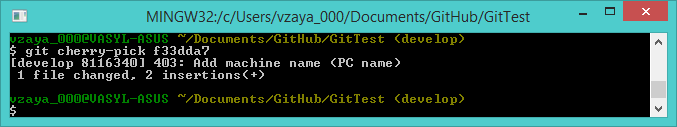
1. Переглянемо усі коміти у вітці develop-feature1, вивівши в одну стрічку хеш коміту, його автора, час зміни (відносний) та повідомлення:

**$ git log develop-feature1 --pretty=format:"%h - %an, %ar : %s"**



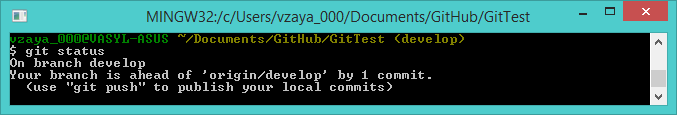
1. Оберемо коміт з повідомленням “403: Add machine name (PC name)”. Заберемо даний коміт у вітку поточну вітку develop викорставши його хеш:

**$ git cherry-pick f33dda7**



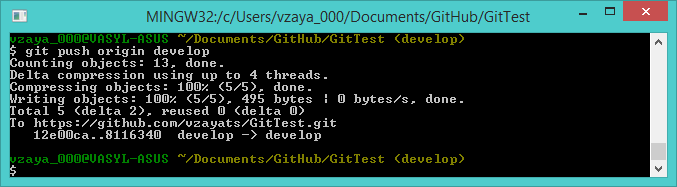
1. Перевіримо, чи ми забрали наш коміт, виконавши команду. Команда виводить поточний статус файлів:

**$ git status**



1. “Виштовхнемо” наші зміни вітки develop на віддалений сервер GitHub:

**$ git push origin develop**

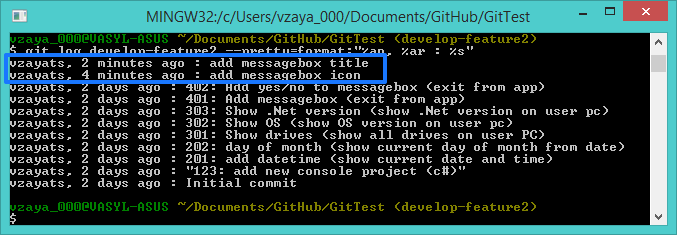


Дано: Вы начали работу над новой фичей и создали от develop дочернюю ветку develop-feature3.

Сделали там несколько коммитов, и синхронизировались с сервером - все ваши изменения теперь в develop-feature3 на сервере. И тут вы вспомнили, что тим лид строго настрого сказал, что сообщение каждого коммита должно быть форматировано определенным образом (см. примечание п.0) и обязательно содержать номер задачи. А вы об этом забыли и теперь на сервере лежит ветка develop-feature1с “плохими” сообщениями. Задание: Предложите вариант, как наиболее просто исправить ситуацию, чтобы тим лид не увидел ваш промах.

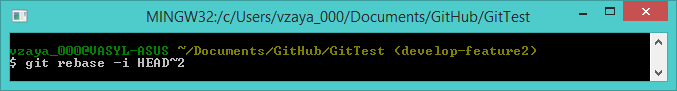
1. Для перегляду комітів, ми використовуємо команду, в якій виводяться усі коміти гілки develop-feature2, та в одному рядку показуються автор, відносну дату та повідомлення коміту. Як бачимо, ми зробили 2 коміта у вітці develop-feature2, та помітили, що повідомлення комітів не відповідають єдиному стандарту.

**$ git log develop-feature2 --pretty=format:"%an, %ar : %s"**



1. Вводимо команду, щоб відкотитися на 2 коміти назад.

**$ git rebase -i HEAD~2**

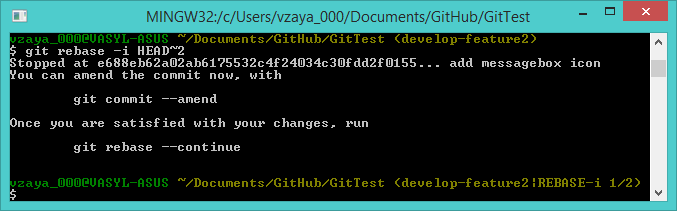


Відразу відкрився текстовий редактор з нашими двома останніми комітами, в яких необхідно поміняти початок стрічки з *pick* на *edit*



Зберігаємо зміни та закриваємо блокнот.

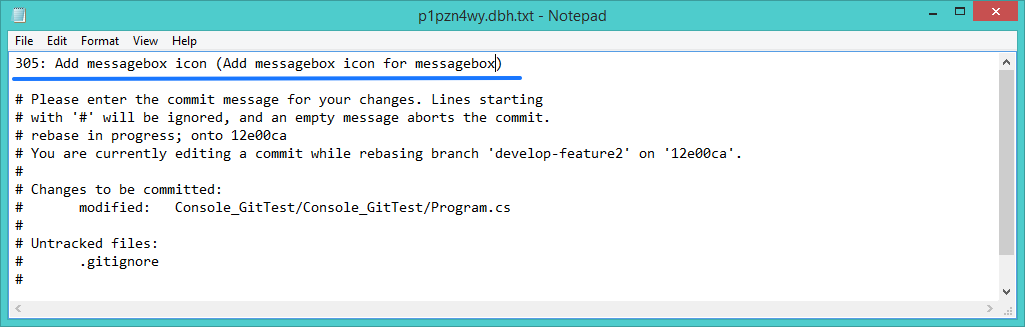
Консоль нас повідомляє, що ми можемо відредагувати повідомлення.



1. Для редагування повідомлень комітів, вводимо команду:

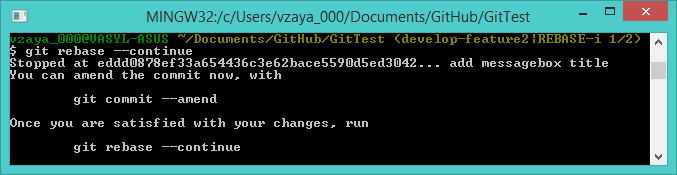
**$ git commit –amend**

Відкриється текстовий редактор, в якому ми вносимо зміни повідомлення першого коміту. Зберігаємо та закриваємо.

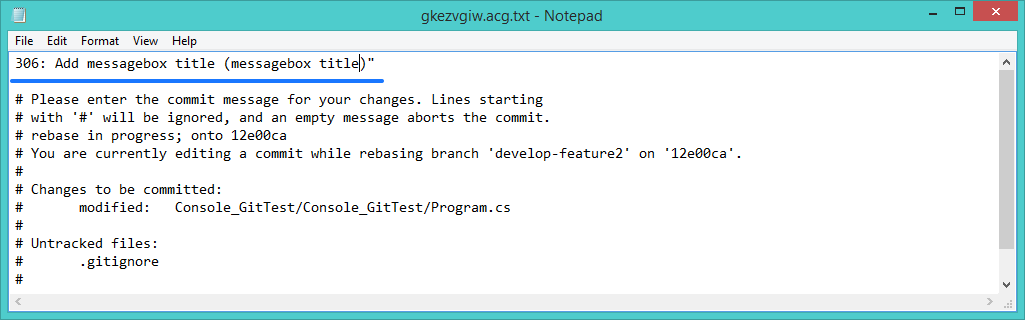


1. Для редагування наступного коміту, вводимо команду:

**$ git rebase --continue**

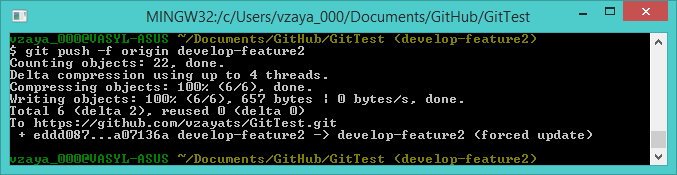


Аналогічно редагуємо повідомлення та зберігаємо.



1. Для перезапису на віддаленому сервері GitHub змін вітки develop-feature2, вводимо команду з ключом –f (force):

**$ git push -f origin develop-feature2**



1. Для перегляду, чи ми внесли зміни у вітку develop-feature2, знову вводимо команду:

**$ git log develop-feature2 --pretty=format:"%an, %ar : %s"**

