git exercise 2

This exercise builds on the exercises in git exercise 1 from yesterday and assumes that you have done the tutorial here: http://learngitbranching.js.org/

And should be done in pairs. If you are not sitting together fysically, then try to use git issues to communicate with comments.

creating branches, merging and rebasing

Person1: Create a new github repo, add collaborators and clone it locally. Then create a netbeans project in the repo.

Person2: Clone the repo and create a new branch: development

On github create a new label (under the Issues tab next to milestone button): 'new feature'

On github add 2 issues: feature1 and feature2. See here for more info

- Inside the issues - make a mention to other team members (HINT: a mention is done using @-sign in front of a github username inside the issue message. Add the label 'new feature' to the 2 issues.

Assign feature1 to Person1 and feature2 to Person2.

Add milestone: sprint1 to both issues. (Hint: milestones are added outside of creating a specific issue – same place as labels.)

Person1 creates and checks out feature1 and add some files + commit and push.

Person2 change the files – add – commit and push. Person 2 makes a pull request (from github).(Hint: see this tutorial) to merge feature1 with development branch.

Now merge feature1 with development (use the button on the github repo page) and add a tag to the merge commit called: v0.1(Hint: quick guide here) – delete the feature1 branch and close the issue on github.

Peson1 creates and checkout a new branch: feature2 - add some files and merge with development and add a tag: v0.2. Close the issue and delete the branch.

On github add a wiki page to the project with some text (used for documentation of the project code)

Create a new branch: feature3 and check it out. Now add file and commit. Add another commit and now instead of a merge with development – do a rebase (so that commit log will show a sequential line of commits in development branch instead of a merge commit with 2 parents).

Check out the commit log afterwards. (hint: `git log --graph --oneline --decorate`).