Exercise 6B ASDT GIT 20.12.2024

By Kateryna Kononikhina 77610

1. Change the message of the last commit to "I can change the last commit message"

   git commit --amend -m "I can change the last commit message"

2. Play with *git rebase*

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

pick d4120bc Remove new_file
reword e8cae61 I can change the last commit message

# Rebase f5ed523..e8cae61 onto f5ed523 (2 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                     commit's log message, unless -C is used, in which case
#                     keep only this commit's message; -c is same as -C but
#                     opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#         create a merge commit using the original merge commit's
#         message (or the oneline, if no original merge commit was
#         specified); use -c <commit> to reword the commit message
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#                     to this position in the new commits. The <ref> is
#                     updated at the end of the rebase
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
~
~
```