

Programming in Java (23/24)

– Exercises Day 2 –

Learning goals

Before the next day, you should achieve the following learning goals:

- Understand the concept of source code version control.
- Use the basic git commands: commit, push, pull, diff, status, log.
- Use github.com to create an account, create a repo, clone it, and push changes onto it.

1 Basic actions on GitHub

Open an account on GitHub (or reuse an existing account that you have opened in the past). Once your account is open, perform the following two tasks:

- Create a new repository names `My first repository`.
- Fork the `currency-exchanger` repository from account `carstenfuhs-bbk` so that you have your own copy. (Here “Fork” means to make a separate copy of the repository inside GitHub using the “Fork” button in the web interface.)

2 Basic actions with git

2.1 Working on your own

Clone your own copy of the `currency-exchanger` repository. This will create a local copy of it on your computer.

Add a few lines of code to the program, or modify some of them, to produce a new version. Commit it and write a short message explaining your changes.

Make additional changes and commit them. Write a descriptive message.

Change to a different folder and clone your repository again. Verify that the clone is equal to the original `currency-exchanger`.

Change to the most up-to-date copy and use `git log` to see your latest changes. Use `git diff` to see the differences between the most recent version you cloned and HEAD.

Push your changes to your copy of currency-exchanger.

Change to a different folder once again and clone your repository again. Verify that this clone contains all your recent changes to currency-exchanger.

Make sure you have completed all the former steps before moving on.

2.2 Working with others

Talk to one of your colleagues in the lab and get their account name and the name they have assigned to their fork of currency-exchanger. (if you are attending online, type this information in the chat). Look at their repository online on GitHub to see their changes.

Pull their repository into your local copy. If the likely case that this results in a conflict, resolve the conflict before committing again; you can solve the conflict in any way you choose: removing their changes, removing your changes, or doing a manual merge of both. Once the conflict is solved, commit.

(If pulling the code did not result in a conflict, pull again from other colleagues until you have a conflict. Solve the conflict and commit.)

Push your latest changes to your copy of the repository. Tell your colleague to pull from it so that you have the same version. Use your web browser to see the changes on GitHub, both yours and your colleague's.