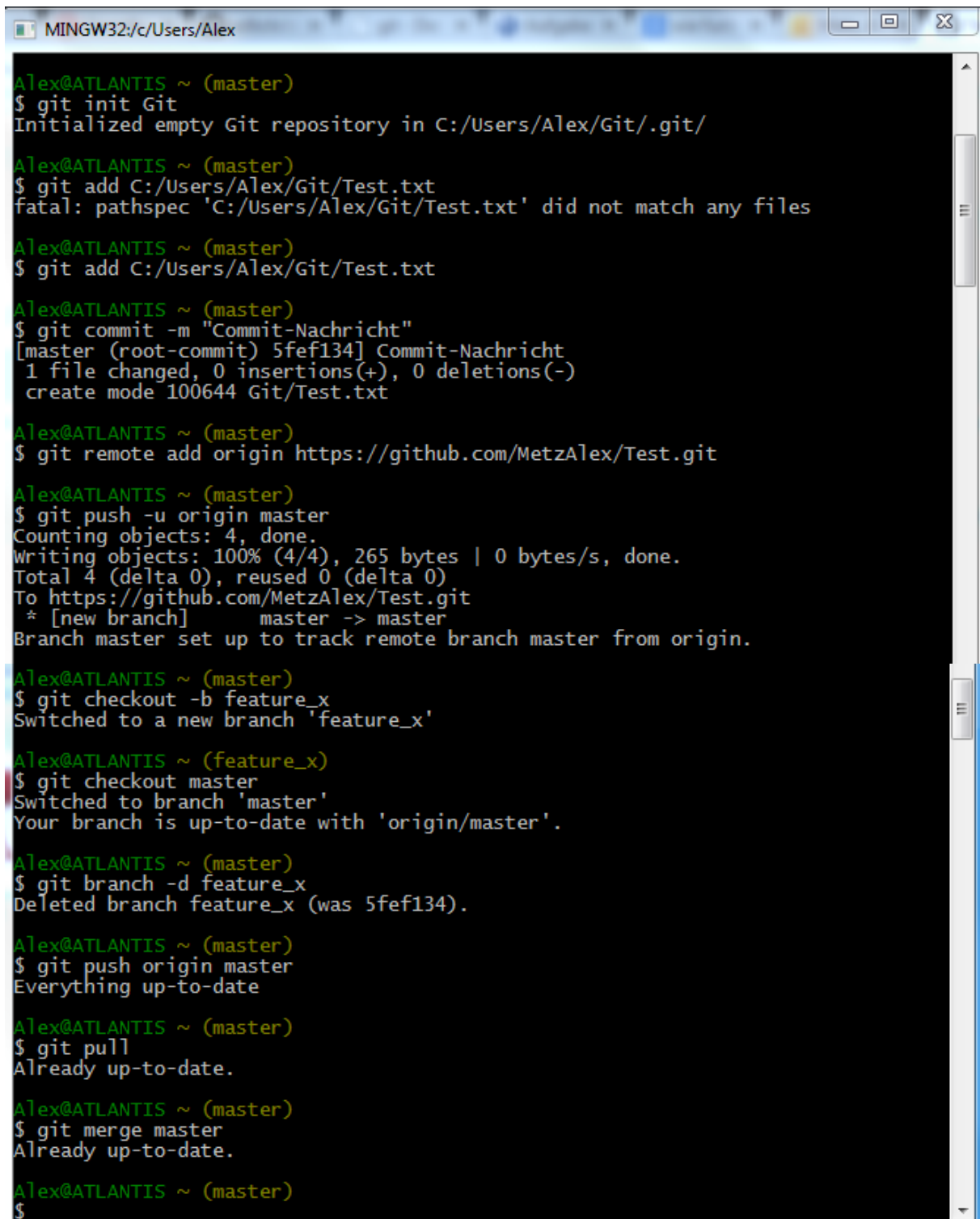


Aufgabe: DVCS mit Git

Ich habe zunächst über das Git Bash manuell einige Methoden ausprobiert, wie man in den Abbildungen 1-4 sieht. Leider habe ich nicht zu allem Screenshots gemacht und das Repository letztlich doch wieder gelöscht, sodass nicht mehr alles beweisbar ist. Ich habe aber init, clone, add, commit, push, pull, merge, checkout, branch, log, tag, diff, status und rm benutzt/ausprobiert.



```
MINGW32:/c/Users/Alex

Alex@ATLANTIS ~ (master)
$ git init Git
Initialized empty Git repository in C:/Users/Alex/Git/.git/

Alex@ATLANTIS ~ (master)
$ git add C:/Users/Alex/Git/Test.txt
fatal: pathspec 'C:/Users/Alex/Git/Test.txt' did not match any files

Alex@ATLANTIS ~ (master)
$ git add C:/Users/Alex/Git/Test.txt

Alex@ATLANTIS ~ (master)
$ git commit -m "Commit-Nachricht"
[master (root-commit) 5fef134] Commit-Nachricht
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Git/Test.txt

Alex@ATLANTIS ~ (master)
$ git remote add origin https://github.com/MetzAlex/Test.git

Alex@ATLANTIS ~ (master)
$ git push -u origin master
Counting objects: 4, done.
Writing objects: 100% (4/4), 265 bytes | 0 bytes/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/MetzAlex/Test.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.

Alex@ATLANTIS ~ (master)
$ git checkout -b feature_x
Switched to a new branch 'feature_x'

Alex@ATLANTIS ~ (feature_x)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

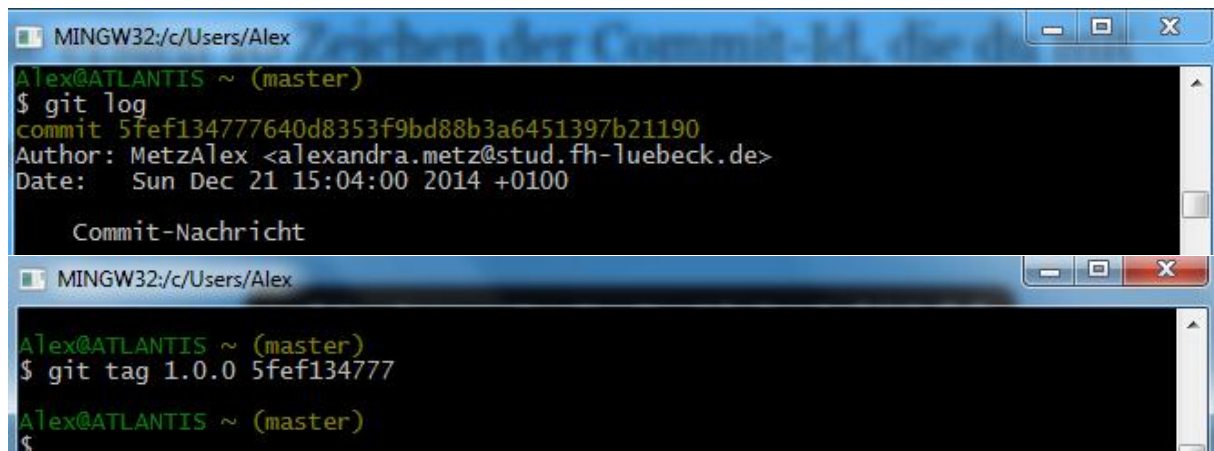
Alex@ATLANTIS ~ (master)
$ git branch -d feature_x
Deleted branch feature_x (was 5fef134).

Alex@ATLANTIS ~ (master)
$ git push origin master
Everything up-to-date

Alex@ATLANTIS ~ (master)
$ git pull
Already up-to-date.

Alex@ATLANTIS ~ (master)
$ git merge master
Already up-to-date.

Alex@ATLANTIS ~ (master)
$
```



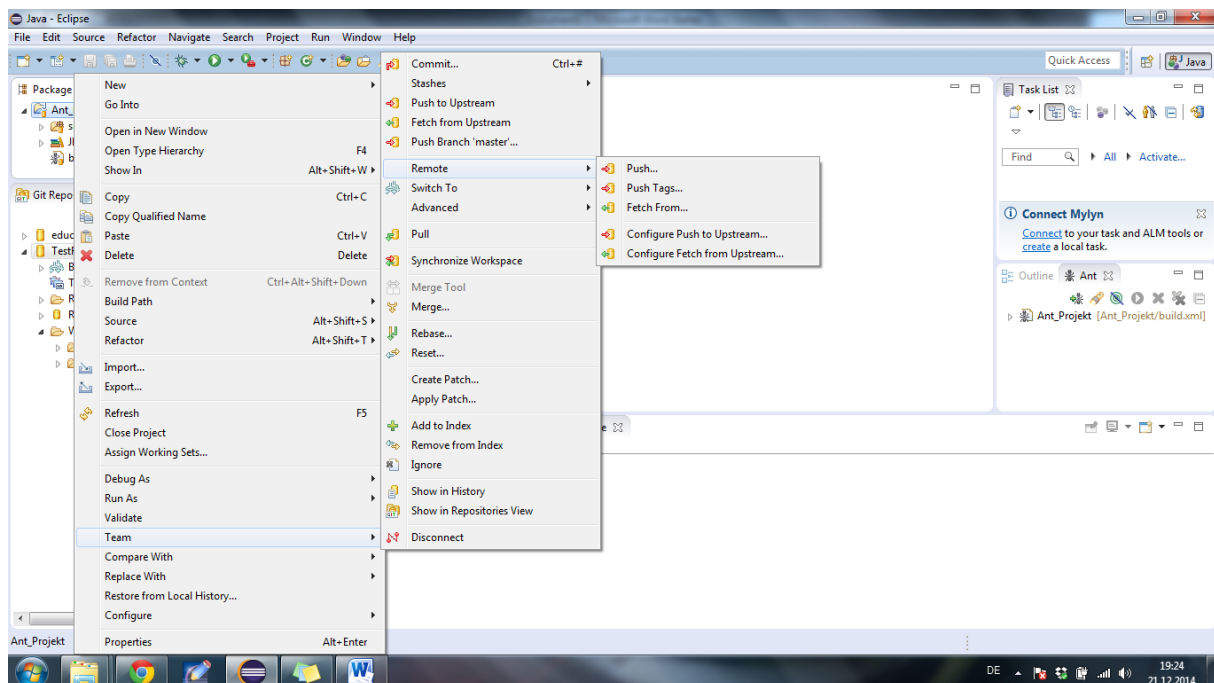
```
MINGW32/c/Users/Alex
Alex@ATLANTIS ~ (master)
$ git log
commit 5fef134777640d8353f9bd88b3a6451397b21190
Author: MetzAlex <alexandra.metz@stud.fh-luebeck.de>
Date: Sun Dec 21 15:04:00 2014 +0100

Commit-Nachricht

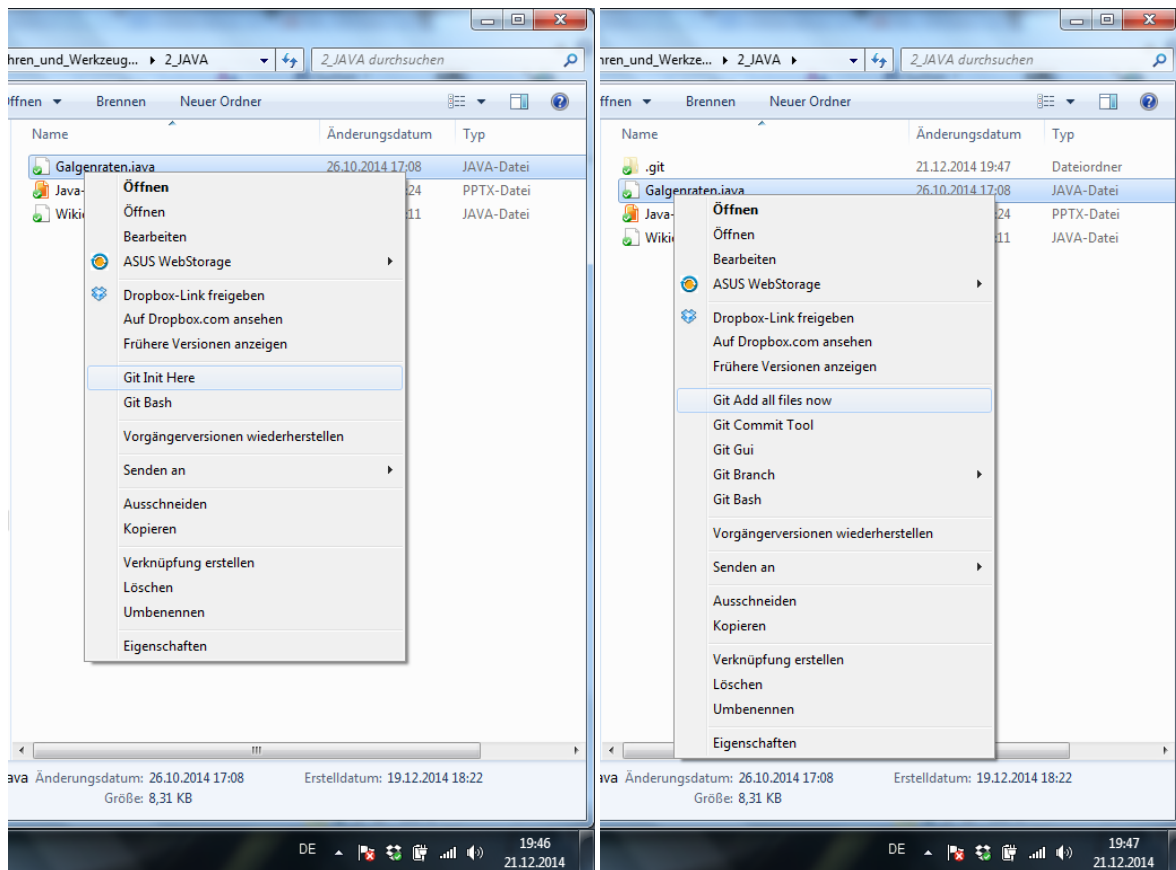
MINGW32/c/Users/Alex
Alex@ATLANTIS ~ (master)
$ git tag 1.0.0 5fef134777
Alex@ATLANTIS ~ (master)
$
```

Anschließend habe ich das Ganze noch einmal mit Eclipse gemacht und festgestellt, dass das viel einfacher ist, wobei ich dort nicht alle Funktionen so gefunden habe. Zum Beispiel ist mit nicht klar, wie man dort direkt eine Datei adden kann, da man eine Datei nur in einem Projekt erstellen und dann das ganze Projekt hinzufügen kann. Dafür kann man aber im Explorer ein Verzeichnis zu einem Git „Init“ machen und dort Dateien adden, die man dann in Eclipse laden und von dort pushen kann. Comitten kann man die Dateien sowohl vom Explorer als auch von Eclipse.

Beim Herumprobieren mit Eclipse habe ich ein schon in Eclipse vorhandenes Projekt (das zur Aufgabe BUI) über „Team“ ins Repository „comittet“ und „gepusht“. Dort waren dann auch einige andere Funktionen, die man benutzen kann, zu finden.



Ich habe auch Ordner im Explorer zu einem Git Init gemacht und die Dateien zu Git geaddet. Diese habe ich dann in Eclipse comittet und gepusht.



Außerdem habe ich das Repository <https://github.com/edlich/education> „geclont“, eine Datei „Baileys_Kuchen.txt“ in das Verzeichnis Cake-Folder „comittet“ und „gepusht“ und eine „Pull-Request“ erstellt.

