

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение
высшего образования

Санкт-Петербургский национальный исследовательский университет
информационных технологий механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа №2

По дисциплине «Инструментальные средства разработки ПО»

“Изучение системы управления версиями Git”

Выполнил студент группы №М32101

Дубов Данил Денисович

Рожновский Иван Васильевич

Проверил:

Дядюшкин Александр Александрович

САНКТ-ПЕТЕРБУРГ

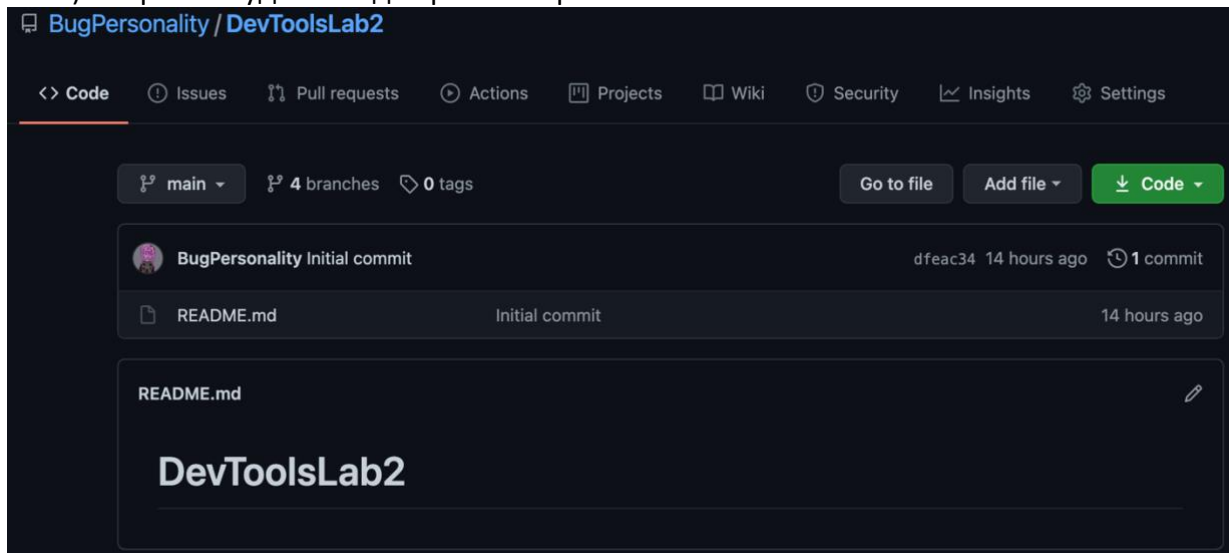
2021

Цель:

Нужно изучить и описать систему управления версиями Git в объёме не более пяти страниц, не считая титульника. Формат сдачи следующий: я задаю вопросы по основным командам Git, вам нужно дать ответы, опираясь на ваши отчеты.

Ход выполнения:

- 1) Первый студент создал репозиторий



- 2) Второй студент делает fork репозитория, создает feature, производит merge (feature -> develop) и делает Pull Request в репозиторий первого студента.

```
commit 31c488ea1a5a9891706df64f8ab0b3e6d82de607 (VanjaRo-develop)
Merge: dfeac34 02a9a0f
Author: VanjaRo <vanjaro43@gmail.com>
Date: Sun Mar 21 22:01:53 2021 +0300

    Merge branch 'feature/submodule' into develop
    Merged feature to develop.

commit 02a9a0f77fda11b00d43adddbe72f3fb4f3df939
Author: VanjaRo <vanjaro43@gmail.com>
Date: Sun Mar 21 21:59:53 2021 +0300

    Added submodule feature.
```

Первый студент принимает Pull Request и делает merge (VanjaRo-develop -> develop)

```
commit cc6a596b97aa38dcaa8d3d8b28360585823b75b4
Merge: dfeac34 31c488e
Author: BugPersonality <dubov.danil.01@mail.ru>
Date: Sun Mar 21 22:14:42 2021 +0300

    Merge branch 'VanjaRo-develop' into develop
```

- 3) Первый студент создает ветку feature/Lfs из ветки develop, добавляет добавляет большой файл с помощью lsf и делает merge (feature lfs -> develop).

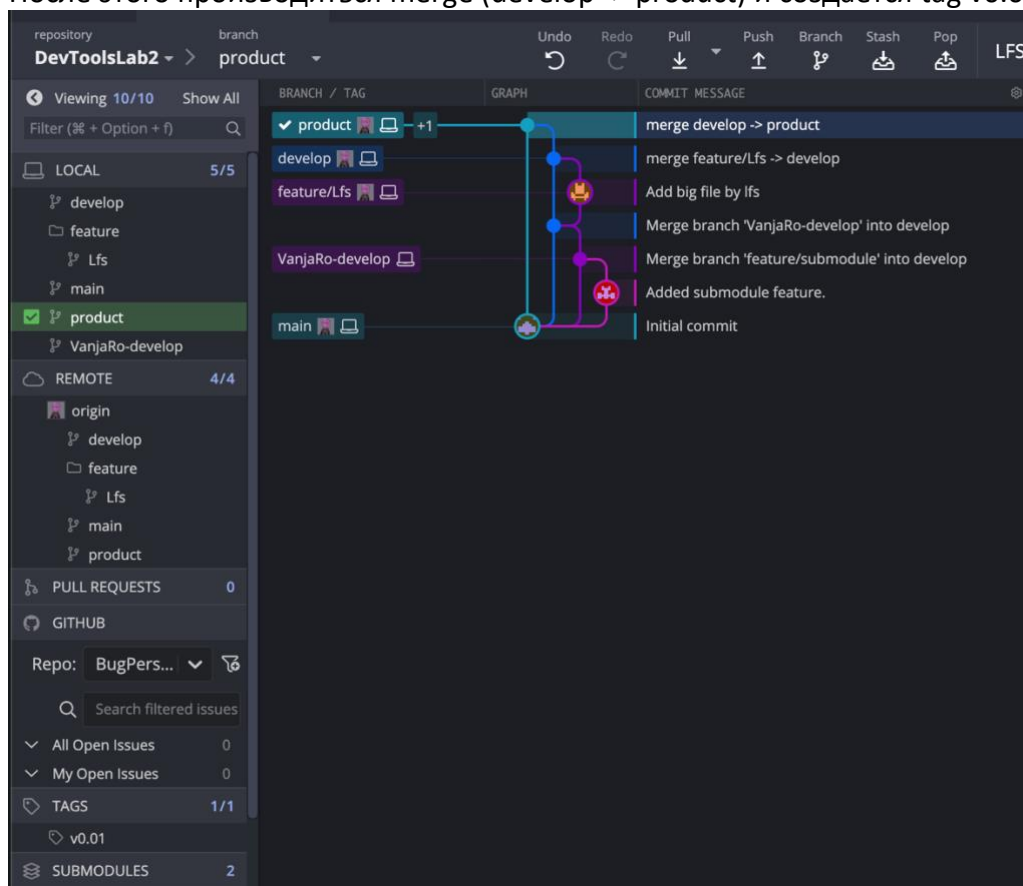
```
commit 8ecbe1d4c7833ac649e5742a7ade3f62cce5ca00 (HEAD -> develop, origin/develop)
Merge: cc6a596 7682008
Author: BugPersonality <dubov.danil.01@mail.ru>
Date: Sun Mar 21 22:36:44 2021 +0300

    merge feature/Lfs -> develop

commit 7682008a21e920a7a7fcb080d4766b51ff5321f1 (origin/feature/Lfs, feature/Lfs)
Author: BugPersonality <dubov.danil.01@mail.ru>
Date: Sun Mar 21 22:34:36 2021 +0300

    Add big file by lfs
```

- 4) После этого производится merge (develop -> product) и создается tag v0.01



Основные команды Git:



Git Cheat Sheet



Create a Repository

From scratch -- Create a new local repository

```
$ git init [project name]
```

Download from an existing repository

```
$ git clone my_url
```

Observe your Repository

List new or modified files not yet committed

```
$ git status
```

Show the changes to files not yet staged

```
$ git diff
```

Show the changes to staged files

```
$ git diff --cached
```

Show all staged and unstaged file changes

```
$ git diff HEAD
```

Show the changes between two commit ids

```
$ git diff commit1 commit2
```

List the change dates and authors for a file

```
$ git blame [file]
```

Show the file changes for a commit id and/or file

```
$ git show [commit]:[file]
```

Show full change history

```
$ git log
```

Show change history for file/directory including diffs

```
$ git log -p [file/directory]
```

Working with Branches

List all local branches

```
$ git branch
```

List all branches, local and remote

```
$ git branch -av
```

Switch to a branch, my_branch, and update working directory

```
$ git checkout my_branch
```

Create a new branch called new_branch

```
$ git branch new_branch
```

Delete the branch called my_branch

```
$ git branch -d my_branch
```

Merge branch_a into branch_b

```
$ git checkout branch_b
```

```
$ git merge branch_a
```

Tag the current commit

```
$ git tag my_tag
```

Make a change

Stages the file, ready for commit

```
$ git add [file]
```

Stage all changed files, ready for commit

```
$ git add .
```

Commit all staged files to versioned history

```
$ git commit -m "commit message"
```

Commit all your tracked files to versioned history

```
$ git commit -am "commit message"
```

Unstages file, keeping the file changes

```
$ git reset [file]
```

Revert everything to the last commit

```
$ git reset --hard
```

Synchronize

Get the latest changes from origin (no merge)

```
$ git fetch
```

Fetch the latest changes from origin and merge

```
$ git pull
```

Fetch the latest changes from origin and rebase

```
$ git pull --rebase
```

Push local changes to the origin

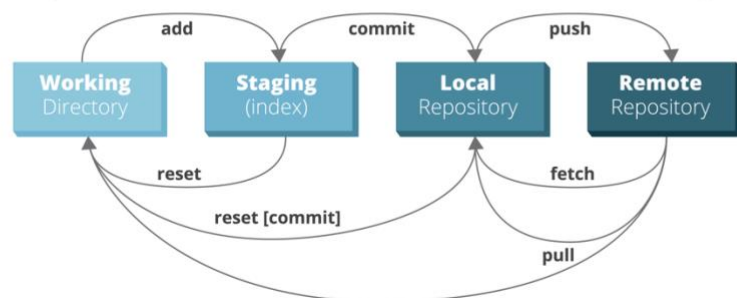
```
$ git push
```

Finally!

When in doubt, use git help

```
$ git command --help
```

Or visit <https://training.github.com/> for official GitHub training.



Вывод:

[В README.md](#)