

Лабораторная работа №1

Маммедгулыев Максат НФИбд-02-19

12 февраля, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

Целью данной лабораторной работы является изучение пространства git, применение различных команд и отработка элементарных действий с системой.

Задачи лабораторной работы

1. Создать учетную запись на github и репозиторий
2. Настроить репозиторий и создать файл для работы дальнейшей
3. Изучить механизм управления версиями и применить простые команды

Процесс выполнения лабораторной работы

Создаем учетную запись на github.com и репозиторий, в котором в дальнейшем будем работать

Owner * Repository name *

Macsad11 / MatMod ✓

Great repository names are short and memorable. Need inspiration? How about [laughing-train?](#)

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Figure 1: Создание учетной записи и репозитория

Инициализируем созданный нами локальный репозиторий, после создаем в нем файл README.md, с которым будем проводить все дальнейшие действия


```
PS C:\work\Labs>
PS C:\work\Labs>
PS C:\work\Labs> git init
Initialized empty Git repository in C:/work/Labs/.git/
PS C:\work\Labs> echo "Лабораторные работы" >> README.md
PS C:\work\Labs> git add README.md
PS C:\work\Labs> _
```

Figure 2: Инициализация репозитория и создание файла

Создаем SSH-ключ, который пропишем в настройках на github.com

```
PS C:\work\labs>
PS C:\work\labs> git init
Initialized empty git repository in C:\work\labs\git\
PS C:\work\labs> echo "I na600at0p0u0e p0b0r0" >> README.md
PS C:\work\labs> git add README.md
PS C:\work\labs> git config --global user.name Maccsadd1
PS C:\work\labs> git config --global user.email "10321944400pfur.ru"
PS C:\work\labs> git commit -m "first commit"
[master (root-commit) 6add9426] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
PS C:\work\labs> ssh-keygen -C "Maccsadd1 10321944400pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\User\ssh/id_rsa):
Created directory C:\Users\User\ssh\
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\User\ssh/id_rsa.
Your public key has been saved in C:\Users\User\ssh/id_rsa.pub.
The key fingerprint is:
SHA256:0k10J2wMPp0qG5YrLb80rTkdYUdE16FXJN6KPM Maccsadd1 10321944400pfur.ru
The key's randomart image is:
+--[RSA 2048]-----+
|
|+-----+
|  =  +
|   = +
|  =  +
|Eo.O X +
|   =  +
|  + . +
|   + . +
|-----[SHA256]-----+
PS C:\work\labs> cat -> ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDwppM91TdUlpTuzCXLcy7f6rTXPj9CtR1tbr9FyzBjFnmlldh0vcYGT6A13Uqn141312w/08zPAVo
S0dz4T7La0LSw0GUS2V1z632j9CkCmp0xbW0bM1tBmESq5s0R8t5B1Y7IwGspX/tcn9pJ3d08p0r7w41Z1384A05Bsdcd3fykAr91bmrQAL63m
4Yh0owwJ2fBTa5Kc5R8mr378h3Z0GfC DUTC0g4KCvL4vkuBPEcEjmwG0c1Kibqubp1vvgm0PULvp300U0w58r6v88yuaU0Pzsqb30CafG3hZfwE
PS C:\work\labs>
```

Figure 3: Создание SSH-ключа



Macsad11
Your personal account

Go to your personal profile

Profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

SSH and GPG keys

Organizations

Moderation

SSH keys / Add new

Title

MatMod

Key

AAAAB3NzaC1yc2EAAAADAQABAAQDdPmM9ITdUlePuLi
wzCXCLy7F/6rTXPj9CI8lThiR9fyz8njFnwUHbozYCGTa63JUQni4l
Ji2my/O8zPAVo5DdZ4t7tAO1sWOCGU52Vlz632j9CKCmegXBM
ObMltbMnEsQsb+R8IsB1Y7IwCGsPX/tcnQ+9PJDDo8pRo7+wi
AZ1H3R4AO5bACSDKbfykAr9lbrmBrQAL63m4YNhowoWJzfBT
ashCwSR9nmt378H3zDMgLfE0utcg64KCvLuFku6PEcejnwGoUC
IKbBquHplvgmOPULup300louU58nv68Byuya0oPTszbqQ30Caf
G3hZFew0yyyZ++FnA2rmWQsXK8n Macsad11
1032194440@pfur.ru

Add SSH key

Figure 4: Добавляем ключ на сайт

Загружаем файлы лицензионного соглашения и gitignore и отправляем эти файлы в сетевой репозиторий.

```
PS C:\work\labs> git remote add origin git@github.com:Macсад11/MatMod.git
PS C:\work\labs> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\work\labs> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\work\labs> git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in LICENSE.
The file will have its original line endings in your working directory
PS C:\work\labs> git commit -m "add license"
[master 21499d8] add license
2 files changed, 555 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
PS C:\work\labs> git push -- origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:4D1V3wvv6TujJhbpZ1sf/zLDAJzPMSvndkr4UvCoqu.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 7.70 KiB | 1.28 MiB/s, done.
total 7 (delta 0), reused 0 (delta 0), pack-reused 0
to github.com:Macсад11/MatMod.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
PS C:\work\labs> git push
Everything up-to-date
PS C:\work\labs>
```

Figure 5: Загрузка файлов с дальнейшей отправкой в сетевой репозиторий

Используем системы управления версиями и тегами. Создаем ветку, начинаем и завершаем в ней релиз.

```
PS C:\work\Labs> git flow init

Which branch should be used for bringing forth production releases?
- master
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] v
Hooks and filters directory? [C:/work/Labs/.git/hooks]
PS C:\work\Labs> git branch
* develop
  master
PS C:\work\Labs> git flow release start 1.0.0
Switched to a new branch 'release/1.0.0'

Summary of actions:
- A new branch 'release/1.0.0' was created, based on 'develop'
- You are now on branch 'release/1.0.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '1.0.0'

PS C:\work\Labs> echo "1.0.0" >> version
PS C:\work\Labs> git add .
PS C:\work\Labs> git commit -am "main: add version"
[release/1.0.0 60c3196] main: add version
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 version
PS C:\work\Labs> git flow release finish -m "ver 1" 1.0.0
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'ort' strategy.
```

Figure 6: Инициализируем git-flow и создание релиза

Используем систему управления версиями

```
PS C:\work\Labs> git push --all
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 486 bytes | 243.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:Macсад11/MatMod.git
 * 21499d8..8430d6d master -> master
 * [new branch] develop -> develop
PS C:\work\Labs> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 160 bytes | 160.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Macсад11/MatMod.git
 * [new tag] v1.0.0 -> v1.0.0
PS C:\work\Labs> _
```

Figure 7: Отправляем изменения в сетевой репозиторий

Выполняем объединение веток

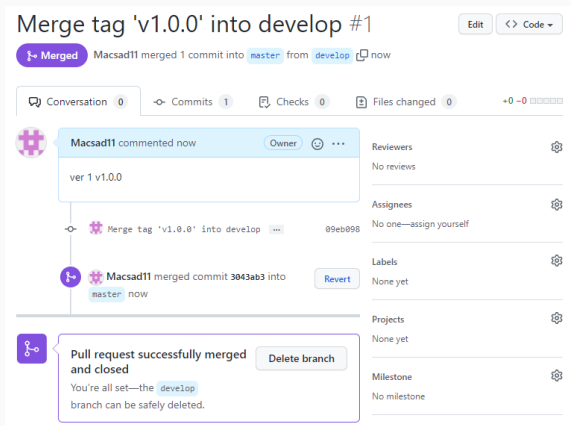


Figure 8: Объединяем ветки в сетевом репозитории

**Сделаем выводы по проделанной
работе в лабораторной №1:**

Мы приобрели практические навыки работы с системой контроля версий git и создали свой репозиторий. Изучили пространства git, применили различные команды и отработали элементарные действия с системой.