

Лабораторная работа 1

Кирилл Дроздов НПИбд-01-19

11 февраля, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

Целью данной работы является изучение идеологии и применения средств контроля версий.

Задачи лабораторной работы


1. Создать учетную запись на github.com
2. Настроить репозиторий
3. Изучить механизм управления версиями

Процесс выполнения лабораторной работы


Создаем учетную запись на github.com и репозиторий

Owner *

Repository name *


 kirilldrozdkov

 /


MatMod 

Great repository names are short and memorable. Need inspiration? How about [bug-free-bassoon?](#)

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 репозитория

Инициализируем локальный репозиторий

```
PS C:\Users\User\Desktop\work\MatMod> git init
Initialized empty Git repository in C:/Users/User/Desktop/work/MatMod/.git/
PS C:\Users\User\Desktop\work\MatMod> echo "# лабораторные работы" >> README.md
PS C:\Users\User\Desktop\work\MatMod> git add README.md
PS C:\Users\User\Desktop\work\MatMod>
```

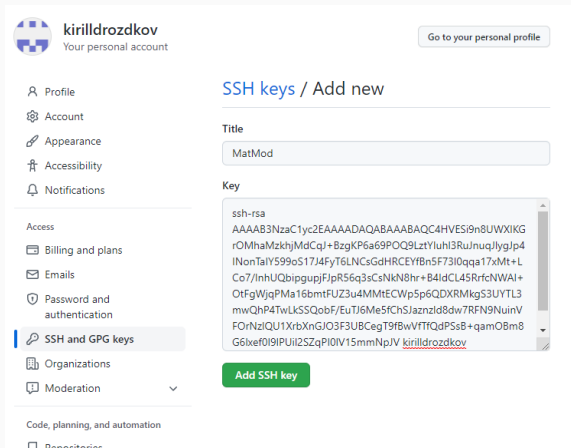
Figure 2: Инициализация репозитория

Создаем SSH-ключ

```
PS C:\Users\User\Desktop\work\MatMod> git init
Initialized empty Git repository in C:\Users\User\Desktop\work\MatMod\.git\
PS C:\Users\User\Desktop\work\MatMod> echo %notaparampue pateru >> README.md
PS C:\Users\User\Desktop\work\MatMod> git add README.md
PS C:\Users\User\Desktop\work\MatMod> git config --global user.name kirilldrozdov
PS C:\Users\User\Desktop\work\MatMod> git config --global user.email "1032160875@pfur.ru"
PS C:\Users\User\Desktop\work\MatMod> git commit -m "First commit"
[master (root-commit) 62268ab] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Users\User\Desktop\work\MatMod> ssh-keygen -C "kirilldrozdov 1032160875@pfur.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:/wJNPFpD/zMSWslxs7AbCtasKdP1x2MwbX1IwVqNU kirilldrozdov 1032160875@pfur.ru
The key's randomart image is:
+-----[RSA 2048]-----+
|      .+..O      |
|      +.E0      |
|      .+..B+     |
|      .+..++     |
|      S+.A.O     |
|      +.B.+.+    |
|      +..+..+    |
|      O+.        |
|      +.         |
+-----[SHA256]-----+
PS C:\Users\User\Desktop\work\MatMod> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCA4HVES19n8UwXIKGrOmhaMzkhJmCq14BzgpK6a69PQ09LztYtuh13RuInuqj1yqJp4INonTaiV599o5l7
J4FYtELNcsdHRCIEYfBn5F31Qqaa7XMTLCo7/InhubipgugjFJpK5Bq3ScsNK8hr+84IdCL4SRfCNMA1+0TfgwJqM16bmtfUz3u4MMTECmp5p6Q0
vWmKps3u7L3mgphfTwLSSqobz/tuTJ6w5fCS1aznz1d6dw/ut8N9uInsvFonhz1Q1XrbXncJ03F3JbCeg7F9fwvFTqdpSsb+qandBaG61xciF0191P
u1125zqP101V15mnpJV kirilldrozdov 1032160875@pfur.ru
PS C:\Users\User\Desktop\work\MatMod>
```

Figure 3: Создание SSH-ключа

Создаем SSH-ключ



The screenshot shows the GitHub account settings page for user 'kirilldrozdov'. The left sidebar contains navigation links: Profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, SSH and GPG keys (highlighted), Organizations, and Moderation. The main content area is titled 'SSH keys / Add new'. It features a 'Title' input field with the value 'MatMod' and a 'Key' text area containing a long SSH key. Below the key is a green 'Add SSH key' button. The key text area also shows the key type 'ssh-rsa' and the user's name 'kirilldrozdov' at the bottom.

kirilldrozdov
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

Code, planning, and automation
Repositories

SSH keys / Add new

Title

MatMod

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAC4HVESi9n8UWXIKG
rOMhaMzkhjMdCqJ+BzgKP6a69POQ9LztYluh3RuJnuqJlygJp4
INonTaiY599oS17J4FyT6LNCsGdHRCeYfBn5F73lOqqa17xMt+L
Co7/lnhUQbipgupjFjpR56q3sCsNkN8hr+B4ldCL45RfcNWAi+
OtFgWjqPMa16bmtFUZ3u4MMtECWp5p6QDXRMkgS3UYTL3
mwQhP4TwLkSSQobF/EuTJ6Me5fChSJaznzd8dw7RFN9NuinV
FOrNzlQU1XrbXnGJO3F3UBCEgT9fBwVfTfQdP5sB+qamO8m8
G6lxf0l9lPUjil2SZqPI0lV15mmNpJV kirilldrozdov
```

Add SSH key

Figure 4: Добавление ключа на github.com

Загружаем служебные файлы

```
PS C:\Users\User\Desktop\work\MatMod> cat ~/ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAHk519h8WkKtK0r0w0m2zkHjMdcqj+BzqKPa69P009LztYUhl3RuJnuqJlygJp4INonTaiY599o5P7
J4FYtGNCsGdHRCFyBnSF7310qgalxmtALCo7/InhU0bipguwpJF3pr56q3c5cNkN8hr+84tdCL43RrFcNWA140tfqwgjQMa16bmtFUZ3u4dMMTECwp5p6QD
Xm4Kgs3UyLJmwhP4TWkLSQobv7EUTJ6Me3fCh5Jaznz1d8dw/RfN9Nu1nvfOrNZ1QULxrbxngJO3F3UBCeg19f8wvFTTQdPSSb-qam08m8G61xf019P
u112zop10VL5mep3Wkirillidrozkov10321608756prfu.ru
PS C:\Users\User\Desktop\work\MatMod>
PS C:\Users\User\Desktop\work\MatMod> git remote add origin git@github.com:kirillidrozkov/MatMod.git
PS C:\Users\User\Desktop\work\MatMod> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\Users\User\Desktop\work\MatMod> git add
PS C:\Users\User\Desktop\work\MatMod> git commit -m "add license"
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:\Users\User\Desktop\work\MatMod> git push -u origin master
[master 10fcec3] add license
2 files changed, 555 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
PS C:\Users\User\Desktop\work\MatMod> git push -u origin master
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:4DY3wvv6TuJJhbpZisF/zLDA0zPMSVhdkr4UvCoQu.
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% (0/3), done.
Writing objects: 100% (7/7), 7.71 KiB | 1.93 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:kirillidrozkov/MatMod.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\User\Desktop\work\MatMod> git push
Everything up-to-date
PS C:\Users\User\Desktop\work\MatMod>
```

Figure 5: Загрузка файлов лицензии и gitignore

Использование системы управления версиями

```
PS C:\Users\User\Desktop\work\MatMod> git flow init
PS C:\Users\User\Desktop\work\MatMod> 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:\Users\User\Desktop\work\MatMod\.git\hooks]
PS C:\Users\User\Desktop\work\MatMod> git branch
* develop
  master
PS C:\Users\User\Desktop\work\MatMod> 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:\Users\User\Desktop\work\MatMod> echo "1.0.0" >> version
PS C:\Users\User\Desktop\work\MatMod> git add .
PS C:\Users\User\Desktop\work\MatMod> git commit -am "main: add version"
[release/1.0.0 b0ff53b] main: add version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
PS C:\Users\User\Desktop\work\MatMod> 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.
version | Bin 0 -> 11 bytes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
Already on 'master'
Your branch is ahead of 'origin/master' by 2 commits.
```

Figure 6: Инициализация git-flow и создание релиза

Использование системы управления версиями

```
PS C:\Users\User\Desktop\work\MatMod> 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), 476 bytes | 238.00 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:kirilldrozdov/MatMod.git
   10fcec3..683293f  master -> master
* [new branch]      develop -> develop
PS C:\Users\User\Desktop\work\MatMod> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 164 bytes | 164.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:kirilldrozdov/MatMod.git
   * [new tag]       v1.0.0 -> v1.0.0
PS C:\Users\User\Desktop\work\MatMod>
```

Figure 7: Отправка изменений в сетевой репозиторий

Выполним объединение веток

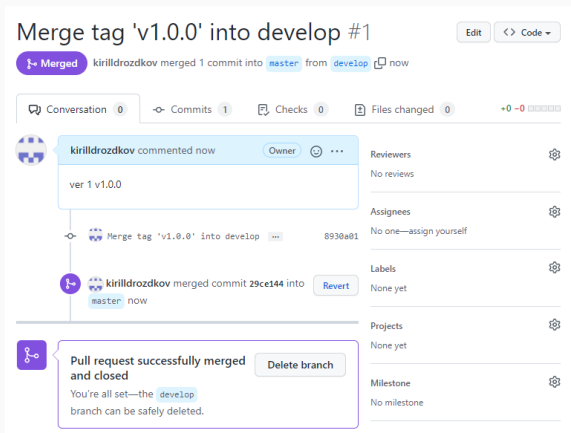


Figure 8: Объединение веток в сетевом репозитории

Выводы по проделанной работе

Мы приобрели практические навыки работы с системой контроля версий git и создали свой репозиторий