

Отчёт по лабораторной работе №1

Система контроля версий Git

Андрей Васильев НПИбд-01-19

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	10
	Список литературы	11

List of Figures

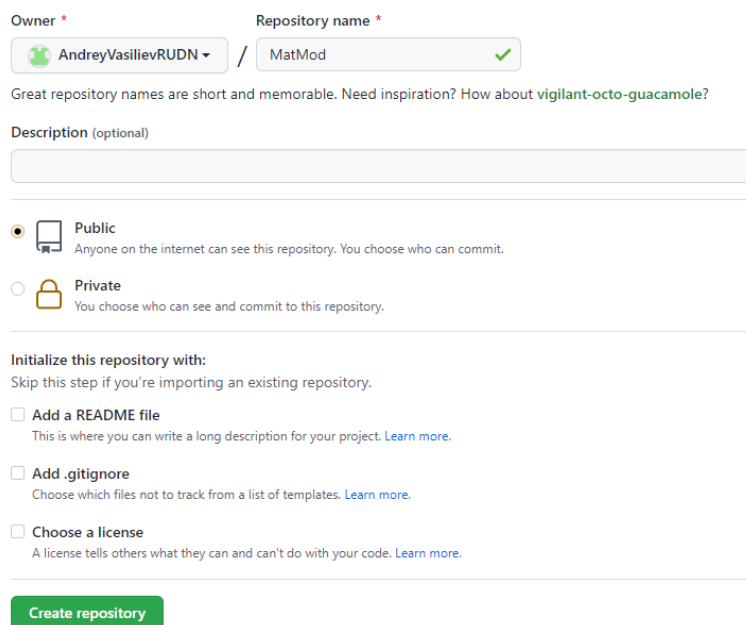
2.1	Создание репозитория	5
2.2	Инициализация репозитория	6
2.3	Создание SSH-ключа	6
2.4	Добавление ключа на github.com	7
2.5	Загрузка файлов	7
2.6	Инициализация git-flow и начало релиза	8
2.7	Завершение релиза и отправка изменений в сетевой репозиторий	8
2.8	Объединение веток в сетевом репозитории	9

1 Цель работы

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

2 Выполнение лабораторной работы

Создаем учетную запись на github.com и репозиторий



The screenshot shows the GitHub 'Create repository' form. At the top, the 'Owner' is set to 'AndreyVasilievRUDN' and the 'Repository name' is 'MatMod', which is marked as valid with a green checkmark. Below this, there is a text input field for the 'Description (optional)'. The 'Visibility' section shows 'Public' selected with a radio button, with a sub-note: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option is also visible but unselected. Under the 'Initialize this repository with:' section, there are three checkboxes: 'Add a README file' (selected), 'Add .gitignore', and 'Choose a license'. Each checkbox has a brief explanation and a 'Learn more' link. At the bottom of the form is a green 'Create repository' button.

Figure 2.1: Создание репозитория

Инициализируем локальный репозиторий и создаю в нем файл README.md

```

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

```

Figure 2.2: Инициализация репозитория

Создаем SSH-ключ и прописываем его в настройках на github.com

```

PS C:\Work\MatMod> git init
Initialized empty Git repository in C:/work/MatMod/.git/
PS C:\Work\MatMod> echo "# лабораторные работы" >> README.md
PS C:\Work\MatMod> git add README.md
PS C:\Work\MatMod> git config --global user.name AndreyVasilievRUDN
PS C:\Work\MatMod> git config --global user.email "103219294@pfur.ru"
PS C:\Work\MatMod> git commit -m "First commit"
[master (root-commit) fcc094c] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Work\MatMod> ssh-keygen -C "AndreyVasilievRUDN 103219294@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:xtL22RjUnpSqrIqdgTj/11fauc/Px492iwlMAWFPko AndreyVasilievRUDN 103219294@pfur.ru
The key's randomart image is:
+-----[RSA 2048]-----+
|      .oo.      |
| o.. .+..+..    |
|+ o ==o+o..    |
| o =E+B*+=+    |
|..oSo=o..      |
|..            |
|.. o o ..      |
|.. o.==..      |
|-----[SHA256]-----+
PS C:\Work\MatMod> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDA0B9YoxAEsh/yxn7yw6qv0J-AKCBowWcgblXZ8iixhR-BPSaxjhpZhsmdttxul3eVmj3v7DrxHgYiPLr
38x256pXfURRhp-IdFM1VnngmS-IHf6m3ovb3pvV41bs6WpTspN63MVKcpZvTD19Kcz2PQSP3PnjaxA6/AL/30z9XxLANf8mWmNkM4Hsao/zKz5seag
VjZnam326092C/oTa0fo7TTktqz7531dR04bcje/7Bwsu0Yxtbvs9epN5QbeHkqb6/6wgvTjsjimbwcqH2b9EvtvrSeb0Bzk/ZyF2gVntoaeFP7gnz806mDW
18FHPJz3jwF-PWnRUF3X AndreyVasilievRUDN 103219294@pfur.ru
PS C:\Work\MatMod>

```

Figure 2.3: Создание SSH-ключа

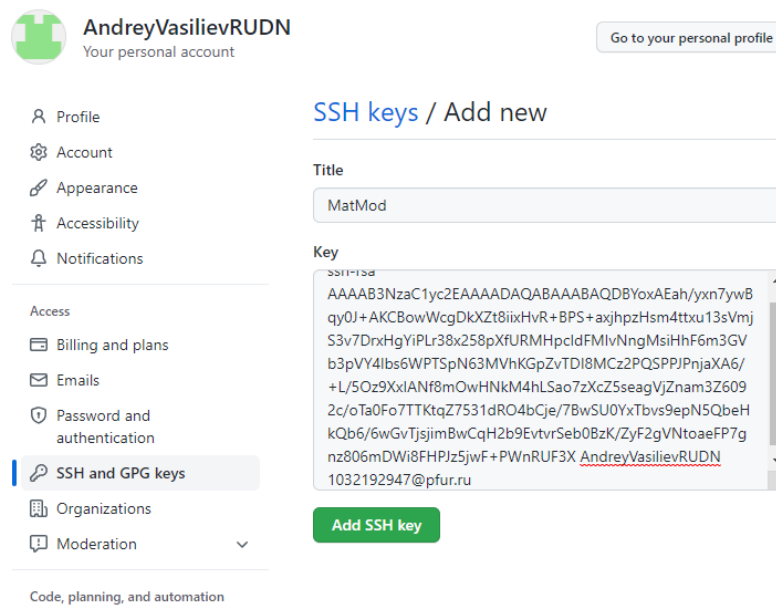


Figure 2.4: Добавление ключа на github.com

Загружаем файлы лицензионного соглашения и gitignore. Отправляем все файлы в сетевой репозиторий.

```
PS C:\Work\MatMod> git remote add origin git@github.com:AndreyVasilievRUDN/MatMod.git
PS C:\Work\MatMod> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\Work\MatMod> 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\MatMod> git commit -am "add license"
[master 02a6e58] add license
2 files changed, 355 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
PS C:\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:+DiY3wvV6TuJJhbpZisF/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.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
PS C:\Work\MatMod> git push -u origin master
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), 7.73 KiB | 1.93 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AndreyVasilievRUDN/MatMod.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Work\MatMod> git push
Everything up-to-date
PS C:\Work\MatMod>
```

Figure 2.5: Загрузка файлов

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

```

PS C:\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:/Work/MatMod/.git/hooks]
PS C:\Work\MatMod> git branch
* develop
  master
PS C:\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:\Work\MatMod> echo "1.0.0" >> version
PS C:\Work\MatMod> git add .
PS C:\Work\MatMod> git commit -am "main: add version"
[release/1.0.0 745e202] main: add version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
PS C:\Work\MatMod> git flow release finish -m "ver 1" 1.0.0

```

Figure 2.6: Инициализация git-flow и начало релиза

```

PS C:\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), 478 bytes | 239.00 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:AndreyVasilievRUDN/MatMod.git
 * [new branch]    master -> master
 * [new branch]    develop -> develop
PS C:\Work\MatMod> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 172 bytes | 86.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:AndreyVasilievRUDN/MatMod.git
 * [new tag]       v1.0.0 -> v1.0.0
PS C:\Work\MatMod>

```

Figure 2.7: Завершение релиза и отправка изменений в сетевой репозиторий

Выполним объединение веток

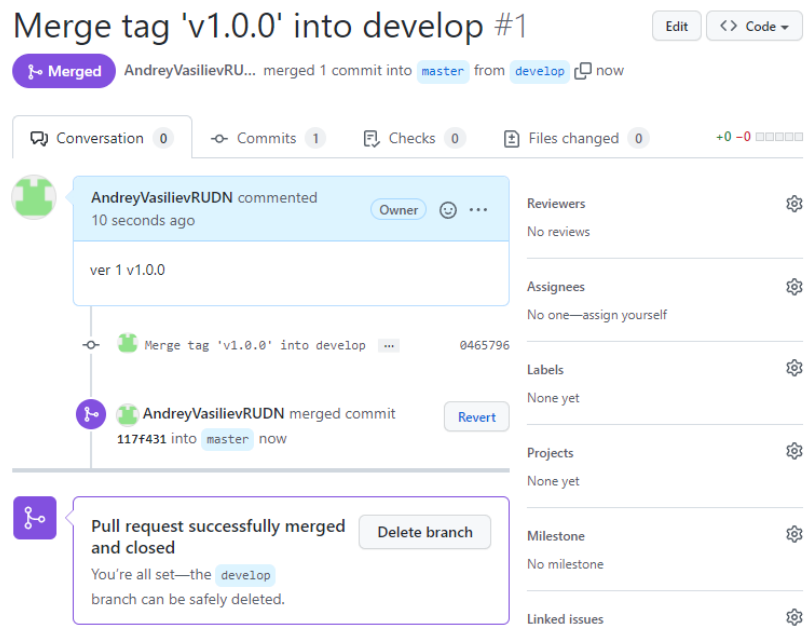


Figure 2.8: Объединение веток в сетевом репозитории

3 Вывод

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

Список литературы

1. Git для новичков
2. Основы Git
3. Руководство по оформлению Markdown файлов