

# **Отчёт по лабораторной работе №1**

**Система контроля версий Git**

Искандер Абдуллаев НФИбд-01-19

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	9
	Список литературы	10

# List of Figures

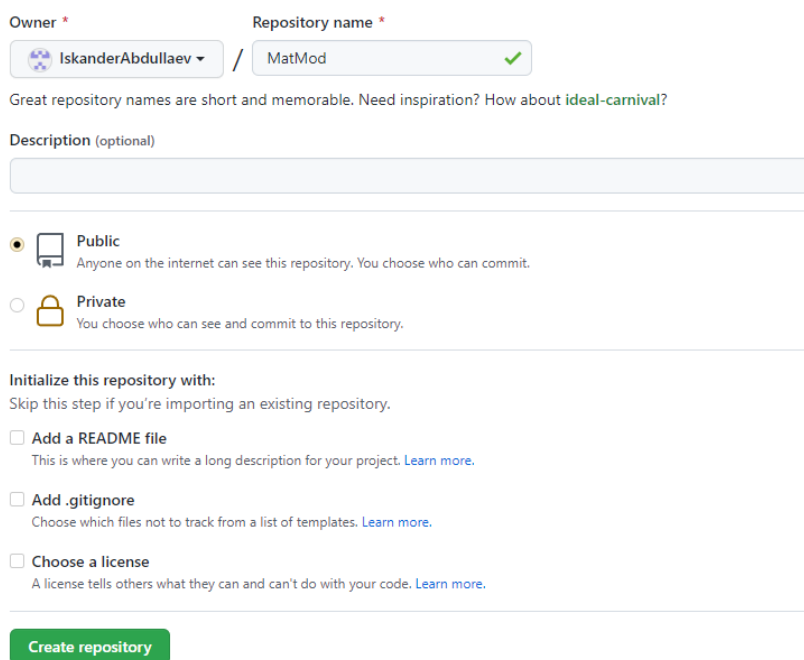
2.1	Создание репозитория . . . . .	5
2.2	Инициализация репозитория . . . . .	5
2.3	Создание SSH-ключа . . . . .	6
2.4	Добавление ключа на github.com . . . . .	6
2.5	Загрузка файлов . . . . .	7
2.6	Инициализация git-flow и начало релиза . . . . .	7
2.7	Завершение релиза и отправка изменений в сетевой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

# **1 Цель работы**

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

## 2 Выполнение лабораторной работы

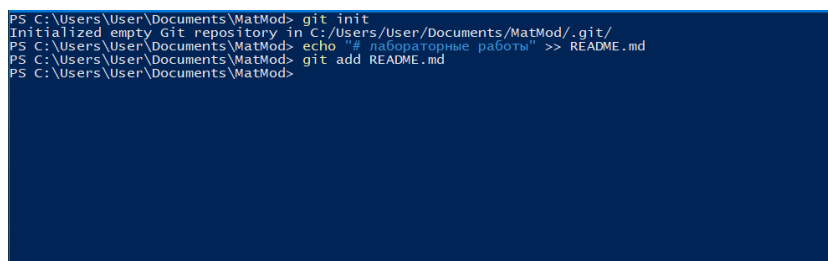
Создаем учетную запись на github.com и репозиторий



The screenshot shows the GitHub 'Create repository' form. At the top, there are two input fields: 'Owner' with a dropdown menu showing 'IskanderAbdullaev' and 'Repository name' with a text input 'MatMod' and a green checkmark. Below these is a hint: 'Great repository names are short and memorable. Need inspiration? How about [ideal-carnival](#)?'. Then is a 'Description (optional)' text area. Below that are two radio buttons for visibility: 'Public' (selected) and 'Private'. Underneath is a section 'Initialize this repository with:' with three checkboxes: 'Add a README file', 'Add .gitignore', and 'Choose a license'. Each checkbox has a brief description and a 'Learn more' link. At the bottom is a green 'Create repository' button.

Figure 2.1: Создание репозитория

Инициализируем локальный репозиторий и создаю в нем файл README.md



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

Figure 2.2: Инициализация репозитория

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

```
PS C:\Users\User\Documents\MatMod> git init
Initialized empty Git repository in C:/Users/User/Documents/MatMod/.git/
PS C:\Users\User\Documents\MatMod> echo "Лабораторные работы" >> README.md
PS C:\Users\User\Documents\MatMod> git add README.md
PS C:\Users\User\Documents\MatMod> git config --global user.name IskanderAbdullaev
PS C:\Users\User\Documents\MatMod> git config --global user.email "1032194840@pfur.ru"
PS C:\Users\User\Documents\MatMod> git commit -m "First commit"
[master (root-commit) 7bb3c43] First commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 README.md
PS C:\Users\User\Documents\MatMod> ssh-keygen -C "IskanderAbdullaev 1032194840@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:Awpu3qMfE3EipCFHaozdJVOX0Zdu7ZWJj/v6Yo8AE IskanderAbdullaev 1032194840@pfur.ru
The key's randomart image is:
+-----[RSA 2048]-----+
|BX0+...o|
|Ox+...+|
|..+...+|
|..+...+|
|..+...+|
|..+...+|
|..+...+|
|..+...+|
|..+...+|
|..+...+|
+-----[SHA256]-----+
PS C:\Users\User\Documents\MatMod> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDPA4K4A9rQxiSoA4qEIB4og3h/ovW1b8frYyX6o4ePG/LGyqRXH9EQR06BT6k1UB3oRweoGwkQD9H8p1
Q05mM1atZa0WvXcyk599Hf1B7qnST64NCqcxEcWd4jKn27M5CdV+Dt4jrOJc35xGIdaRC7bxXMsmSlnO6
ur6xylkWdGfb/Uj4afbOR8d4BENCirJqERh9TGAOzyRNQIRaosIR
ErinXcU902kVsG+6RFkimeApWrauxxzw1zalRnJSuArqyoD7vUK
yH9Q2jYGZShsF82ZHOKwYwNQJEjJbzSXq/MqGNvaEUQicrbTpz
6Lqd/Y1adHWerPfmXUa96HY/1h3 IskanderAbdullaev
1032194840@pfur.ru
PS C:\Users\User\Documents\MatMod>
```

Figure 2.3: Создание SSH-ключа

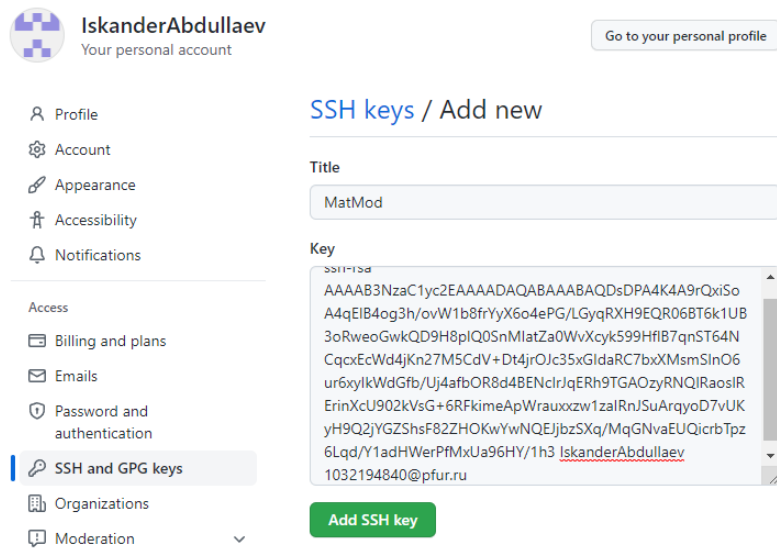


Figure 2.4: Добавление ключа на github.com

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

```

PS C:\Users\User\Documents\MatMod> git remote add origin git@github.com:iskanderAbdullaev/MatMod.git
PS C:\Users\User\Documents\MatMod> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\Users\User\Documents\MatMod> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\Users\User\Documents\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:\Users\User\Documents\MatMod> git commit -am "add license"
[master 383ed8b] add license
2 files changed, 355 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
PS C:\Users\User\Documents\MatMod> git push -u origin master
The authenticity of host 'github.com (140.82.121.3)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6tUJJhbpz1sF/zLDA0zPMSvHdK4UyvcOqU.
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), 7.72 KiB | 1.93 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
to github.com:iskanderAbdullaev/MatMod.git
* [new branch] master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\User\Documents\MatMod> git push
Everything up-to-date
PS C:\Users\User\Documents\MatMod>

```

Figure 2.5: Загрузка файлов

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

```

PS C:\Users\User\Documents\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/Documents/MatMod/.git/hooks]
PS C:\Users\User\Documents\MatMod> git branch
* develop
master
PS C:\Users\User\Documents\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\Documents\MatMod> echo "1.0.0" >> version
PS C:\Users\User\Documents\MatMod> git add .
PS C:\Users\User\Documents\MatMod> git commit -am "main: add version"
[release/1.0.0 ddd9alb] main: add version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
PS C:\Users\User\Documents\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 -> 16 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.
(Use 'git push' to publish your local commits)

```

Figure 2.6: Инициализация git-flow и начало релиза

```

PS C:\Users\User\Documents\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 | 158.00 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:IskanderAbdullaev/MatMod.git
 * 383ed8b..3649317 master -> master
 * [new branch] develop -> develop
PS C:\Users\User\Documents\MatMod> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 167 bytes | 167.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:IskanderAbdullaev/MatMod.git
 * [new tag] v1.0.0 -> v1.0.0
PS C:\Users\User\Documents\MatMod>

```

Figure 2.7: Завершение релиза и отправка изменений в сетевой репозиторий

Выполним объединение веток

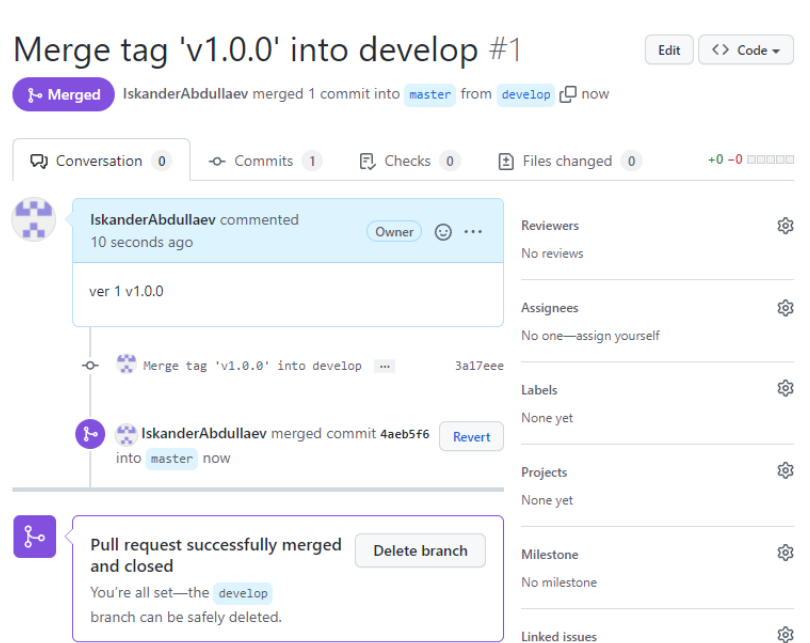


Figure 2.8: Объединение веток в сетевом репозитории



## 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