

# Лабораторная работа 1

---

Чан Куок Кхань НПИбд-02-19

10 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](https://github.com)
2. Настроить репозиторий
3. Изучить механизм управления версиями

# **Процесс выполнения лабораторной работы**

---

# Создаем учетную запись на github.com и репозиторий


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

Owner \*

Repository name \*


 chankuokkhan ▾

/ matmodel ✓


Great repository names are short and unique. **matmodel is available.** Need inspiration? How about [friendly-spork?](#)

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**

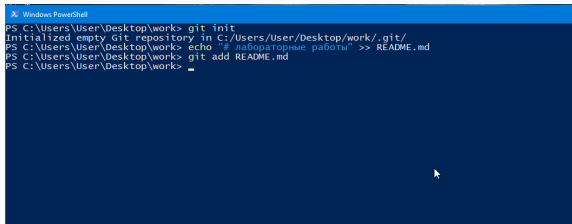
A license tells others what they can and can't do with your code. [Learn more.](#)

---

Create repository

**Figure 1:** Создание репозитория

# Инициализируем локальный репозиторий



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

Figure 2: Инициализация репозитория

# Создаем SSH-ключ

```
PS C:\Users\User\Desktop\work> git init
Initialized empty Git repository in C:\Users\User\Desktop\work/.git/
PS C:\Users\User\Desktop\work> echo # chankuokkhan >> README.md
PS C:\Users\User\Desktop\work> git add README.md
PS C:\Users\User\Desktop\work> git config --global user.name chankuokkhan
PS C:\Users\User\Desktop\work> git config --global user.email 1032189063@pfur.ru
PS C:\Users\User\Desktop\work> git commit -m "first commit"
[master (root-commit) 524280c] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
PS C:\Users\User\Desktop\work>
PS C:\Users\User\Desktop\work> ssh-keygen -C "chankuokkhan 1032189063@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:opjngcdv0DATEXPREFj5QIud2wafnDLHbuk3p1uk chankuokkhan 1032189063@pfur.ru
The key's randomart image is:
+---[RSA 2048]---+
  .==+
  .oo*
  o."o+
  + .o
  .+$.
  o+
  ++o+
  BO =
  oOE
+---[SHA256]-----+
PS C:\Users\User\Desktop\work> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBTBp6bduvq1rCNEzX3Qyvd/yw/p6sqv1ZLpnuh1dK-B9k6GpMBpfqo5bcgmIT9TX20-5d-lvL2y
z/dn6scnrl10023K3vTA1pT19e0s3bcU8qslaw6dwn13rcXX8D2b3kyoqpwM8R-Bd8U3jmeC4+Stuxqo18KYvMB6N2CFYRP2CH3Vde+FrC85L
D3hx1215o6Uu-SrCM8mAvJefvvr8a0f1xp8PvoaSouxyjpvjvB312A5Bb6U2N+hz/RNygqXvjdsTSQskcuAHC25hAqX2Ry1AHKgz0bYsu01iaQLVs81s7Q
L9mu1D7xylhktUwWmdp chankuokkhan 1032189063@pfur.ru
PS C:\Users\User\Desktop\work>
```

Figure 3: Создание SSH-ключа



SSH keys / Add new

---

Title

sss

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDFQjTbgWbdvngirC
NlZVX3SyOd/yw/pn6zGVi2Lpn+WiGk89K6q6MBEpfq05Cbcg
mHT9TXZQ+5d+hYL2yS/D6n8GAink1qTQO23K/5vTA1pT19eQ
x3bcUBqeaUOMMd6Mini3YtXX8D2b/KyopqWMH8F8d8U3jma
C8+Stuxgoa18EKYnB6H2CFYFRPJCHk1Vde+frCX65LDJhx1215o
WkU+SrOMBmAuvJeFvwt8a0ixp8Pvoa5ouXyJpvjvB3i2A5Bb6U
2N+h2/RNYgoqXvjdsTSQsKcUAhC25hvAqX2Ry1AHKgZQbYSu
OliaQLVs8ls7QL9nu1D7RyLhktUWwBnwD chankuokkhan
```

Add SSH key

**Figure 4:** Добавление ключа на github.com

# Загружаем служебные файлы

```
PS C:\Users\User\Desktop\work> git remote add origin git@github.com:chankuokkhan/matsmodel.git
PS C:\Users\User\Desktop\work> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\Users\User\Desktop\work> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\Users\User\Desktop\work> 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\Desktop\work> git commit -am "add license"
[master 44f7b46] add license
 2 files changed, 555 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 LICENSE
PS C:\Users\User\Desktop\work> git push -u origin master
The authenticity of host 'github.com [140.82.121.4]' can't be established.
ED25519 key fingerprint is SHA256:+D1Y3wuvV6TujhbpZ1sF/zLDA0zPMSvldkr4UvC0qu.
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% (3/3), done.
Writing objects: 100% (7/7), 7.71 KiB | 1.54 MiB/s, done.
Total (delta 0), reused 0 (delta 0), pack-reused 0
to github.com:chankuokkhan/matsmodel.git
 * [new branch] master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\User\Desktop\work> git push
Everything up-to-date
PS C:\Users\User\Desktop\work>
```

Figure 5: Загрузка файлов лицензии и gitignore

# Использование системы управления версиями

```
PS C:\Users\User\Desktop\work> 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/.git/hooks]
PS C:\Users\User\Desktop\work> git branch
* develop
  master
PS C:\Users\User\Desktop\work> 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> echo "1.0.0" >> version
PS C:\Users\User\Desktop\work> git add .
PS C:\Users\User\Desktop\work> git commit -am "chore(main): add version"
[release/1.0.0 7609166] chore(main): add version
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 version
PS C:\Users\User\Desktop\work> 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 1.0.0
1 file changed, 1 insertion(+), 1 deletion(-)
```

Figure 6: Инициализация git-flow и создание релиза

# Использование системы управления версиями

```
PS C:\Users\User\Desktop\work> 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), 498 bytes | 166.00 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:chankuokkhan/matmodel.git
  44f7b46..fa054d8  master -> master
* [new branch]      develop -> develop
PS C:\Users\User\Desktop\work> 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:chankuokkhan/matmodel.git
  * [new tag]         v1.0.0 -> v1.0.0
PS C:\Users\User\Desktop\work>
```

**Figure 7:** Отправка изменений в сетевой репозиторий

# Выполним объединение веток

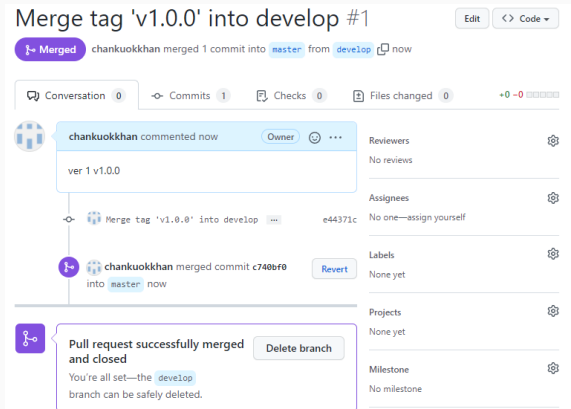


Figure 8: Объединение веток в сетевом репозитории

## **Выводы по проделанной работе**

---

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