Лабораторная работа 1

Седов Никита Сергеевич НПИбд-01-19

10 февраля, 2022, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

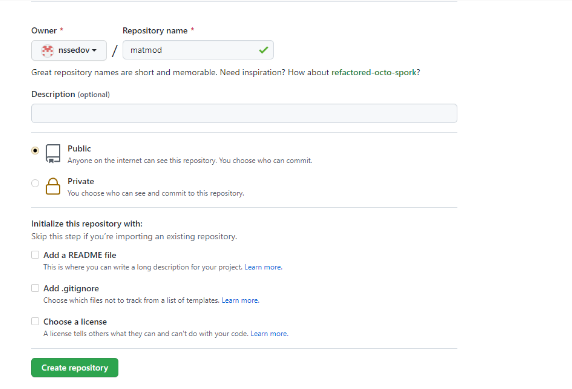
Целью данной работы является изучение идеологии и применения средств контроля версий.

Задачи лабораторной работы

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

Процесс выполнения лабораторной работы

Создаем учетную запись на github.com и репозиторий



Owner * Repository name *

nssedov / matmod ✓

Great repository names are short and memorable. Need inspiration? How about refactored-octo-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.

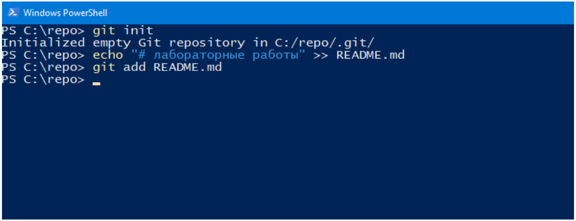
□ Add a README file
This is where you can write a long description for your project. Learn more.

□ 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.

Figure 1: Создание репозитория

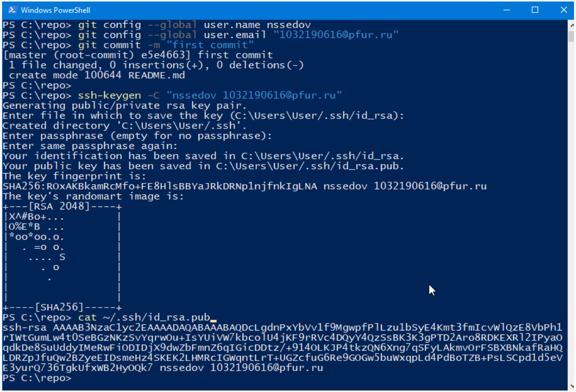
Инициализируем локальный репозиторий



```
PS C:\repo> git init
Initialized empty Git repository in C:/repo/.git/
PS C:\repo> echo "# лабораторные работы" >> README.md
PS C:\repo> git add README.md
PS C:\repo> _
```

Figure 2: Инициализация репозитория

Создаем SSH-ключ

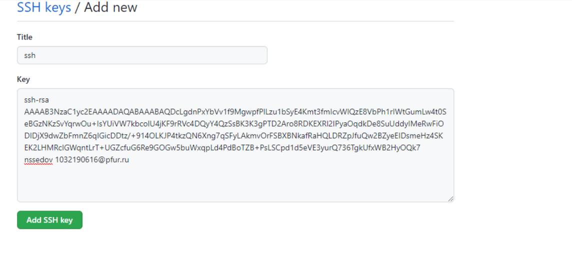


Windows PowerShell

```
PS C:\repo> git config --global user.name nssedov
PS C:\repo> git config --global user.email "1032190616@pfur.ru"
PS C:\repo> git commit -m "first commit"
[master (root-commit) e5e4663] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
PS C:\repo>
PS C:\repo> ssh-keygen -C "nssedov 1032190616@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:ROxAKBkamRcMfo+FE8HlsBBYaJRkDRNp1njfnkIgLNA nssedov 1032190616@pfur.ru
The key's randomart image is:
+---[RSA 2048]----+
|X^#Bo+...        |
|O%E*B ...        |
|*oo*oo.o.        |
|  . =o o.        |
|   .... S        |
|     . o         |
|      .          |
|                 |
|                 |
+----[SHA256]-----+
PS C:\repo> cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDcLgdnPxYbVv1f9MgwpfPlLzu1bSyE4Kmt3fmIcvWlQzE8VbPh1
rIWtGumLw4t0SeBGzNKzSvYqrwOu+IsYUiVW7kbcolU4jKF9rRVc4DQyY4QzSsBK3K3gPTD2Aro8RDKEXRl2IPyaO
qdkDe8SuUddyIMeRwFiODIDjX9dwZbFmnZ6qIGicDDtz/+914OLKJP4tkzQN6Xng7qSFyLAkmvOrFSBXBNkafRaHQ
LDRZpJfuQw2BZyeEIDsmeHz4SKEK2LHMRcIGWqntLrT+UGZcfuG6Re9GOGw5buWxqpLd4PdBoTZB+PsLSCpd1d5eV
E3yurQ736TgkUfxWB2HyOQk7 nssedov 1032190616@pfur.ru
PS C:\repo>
```

Figure 3: Создание SSH-ключа

Создаем SSH-ключ



SSH keys / Add new

Title

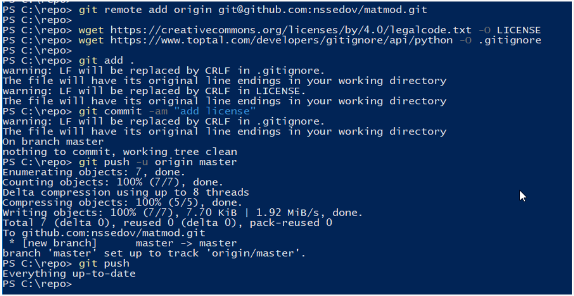
ssh

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDcLgdnPxYbVv1f9MgwpfPlLzu1bSyE4Kmt3fmIcvWlQzE8VbPh1rlWtGumLw4t0S
eBGzNKzSvYqrwOu+IsYUiVW7kbcolU4jKF9rRVc4DQyY4QzSsBK3K3gPTD2Aro8RDKEXRI2IPyaOqdkDe8SuUddylMeRwFiO
DIDjX9dwZbFmnZ6qIGicDDtz/+914OLKJP4tkzQN6Xng7qSFyLAkmvOrFSBXBNkafRaHQLDRZpJfuQw2BZyeEIDsmeHz4SK
EK2LHMRclGWqntLrT+UGZcfuG6Re9GOGw5buWxqpLd4PdBoTZB+PsLSCpd1d5eVE3yurQ736TgkUfxWB2HyOQk7
nssedov 1032190616@pfur.ru
```

Figure 4: Добавление ключа на github.com

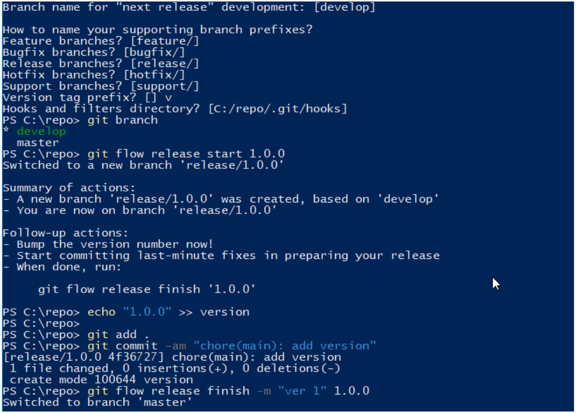
Загружаем служебные файлы



```
PS C:\repo> git remote add origin git@github.com:nssedov/matmod.git
PS C:\repo>
PS C:\repo> wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O LICENSE
PS C:\repo> wget https://www.toptal.com/developers/gitignore/api/python -O .gitignore
PS C:\repo>
PS C:\repo> 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:\repo> git commit -am "add license"
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
On branch master
nothing to commit, working tree clean
PS C:\repo> 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), 7.70 KiB | 1.92 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:nssedov/matmod.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\repo> git push
Everything up-to-date
PS C:\repo>
```

Figure 5: Загрузка файлов лицензии и gitignore

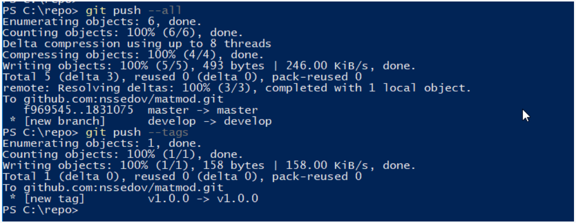
Использование системы управления версиями



```
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:/repo/.git/hooks]
PS C:\repo> git branch
* develop
  master
PS C:\repo> 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:\repo> echo "1.0.0" >> version
PS C:\repo>
PS C:\repo> git add .
PS C:\repo> git commit -am "chore(main): add version"
[release/1.0.0 4f36727] chore(main): add version
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 version
PS C:\repo> git flow release finish -m "ver 1" 1.0.0
Switched to branch 'master'
```

Figure 6: Инициализация git-flow и создание релиза

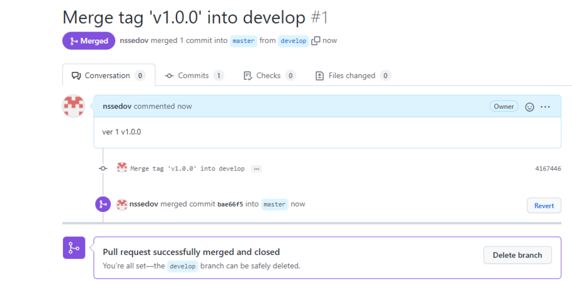
Использование системы управления версиями



```
PS C:\repo>
PS C:\repo> 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), 493 bytes | 246.00 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:nssedov/matmod.git
   f969545..1831075  master -> master
 * [new branch]      develop -> develop
PS C:\repo> git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 158 bytes | 158.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:nssedov/matmod.git
 * [new tag]         v1.0.0 -> v1.0.0
PS C:\repo>
```

Figure 7: Отправка изменений в сетевой репозиторий

Выполним объединение веток



Merge tag 'v1.0.0' into develop #1

Merged nssedov merged 1 commit into master from develop now

nssedov commented now

ver 1 v1.0.0

Merge tag 'v1.0.0' into develop ... 4167446

nssedov merged commit bae66f5 into master now

Pull request successfully merged and closed

You're all set—the develop branch can be safely deleted.

Figure 8: Объединение веток в сетевом репозитории

Выводы по проделанной работе

Вывод

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