

# **Научное программирование**

**Отче по лабораторной работе № 2**

Меньшов Иван Сергеевич НПМмд-02-21

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Теоретические сведения</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>
<b>5</b>	<b>Контрольные вопросы</b>	<b>17</b>

# List of Figures

3.1	рисунок 1 . . . . .	6
3.2	рисунок 2 . . . . .	7
3.3	рисунок 3 . . . . .	7
3.4	рисунок 4 . . . . .	8
3.5	рисунок 5 . . . . .	8
3.6	рисунок 6 . . . . .	9
3.7	рисунок 7 . . . . .	10
3.8	рисунок 8 . . . . .	10
3.9	рисунок 9 . . . . .	10
3.10	рисунок 10 . . . . .	11
3.11	рисунок 11 . . . . .	12
3.12	рисунок 12 . . . . .	12
3.13	рисунок 13 . . . . .	13
3.14	рисунок 14 . . . . .	14
3.15	рисунок 15 . . . . .	14
3.16	рисунок 16 . . . . .	14
3.17	рисунок 17 . . . . .	15
3.18	рисунок 18 . . . . .	15

# 1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

## 2 Теоретические сведения

Вся теоритическая часть по использованию языка разметки Markdown была взята из инструкции по лабораторной работе №2 на сайте: <https://esystem.rudn.ru/pluginfile.php/1284/markdown.pdf>

### 3 Выполнение лабораторной работы

1. Создадим учётную запись на <https://github.com>.

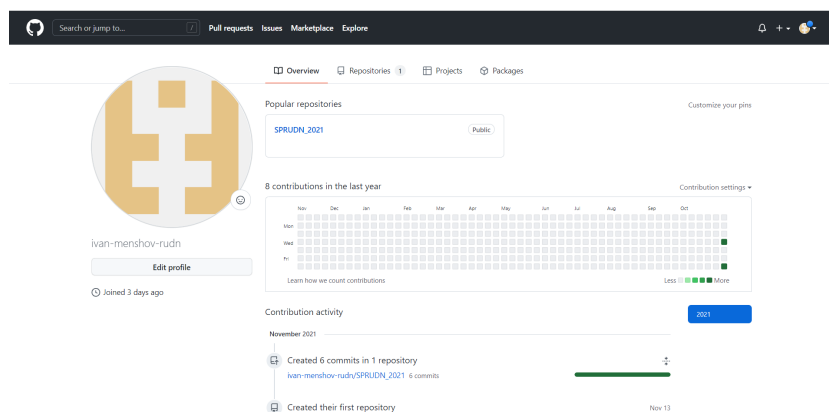
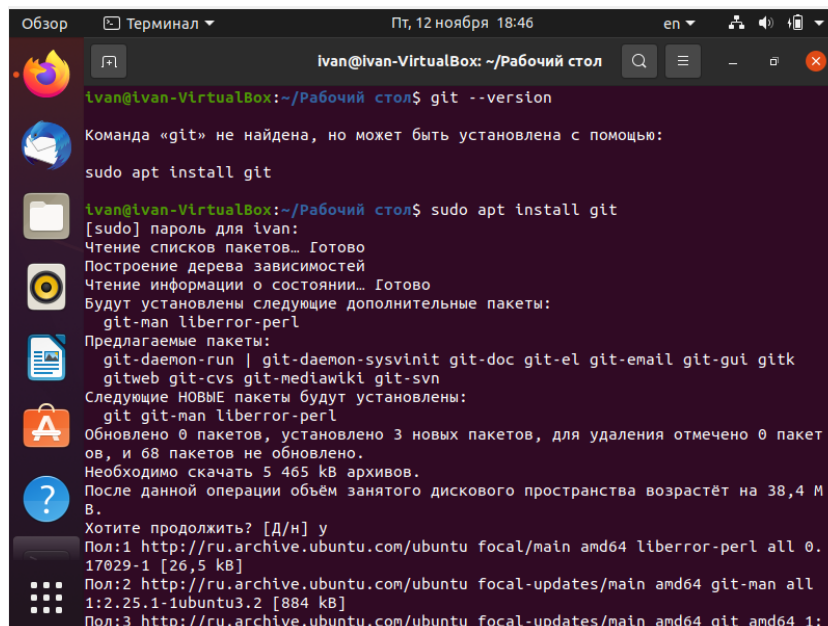


Figure 3.1: рисунок 1

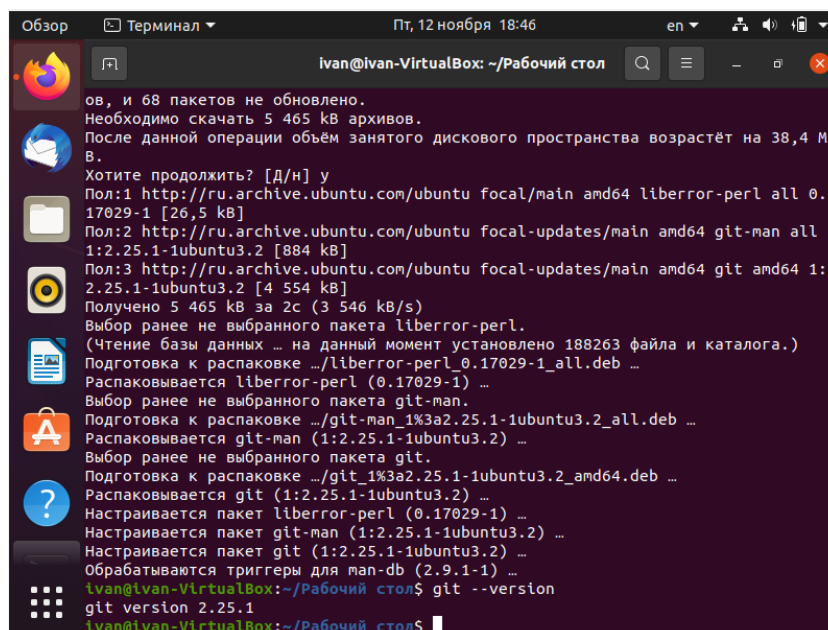
2. Установим git на наш компьютер.



```
ivan@ivan-VirtualBox: ~/Рабочий стол
ivan@ivan-VirtualBox:~/Рабочий стол$ git --version
Команда «git» не найдена, но может быть установлена с помощью:
sudo apt install git

ivan@ivan-VirtualBox:~/Рабочий стол$ sudo apt install git
[sudo] пароль для ivan:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
git-man liberror-perl
Предлагаемые пакеты:
git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
gitweb git-cvs git-mediawiki git-svn
Следующие НОВЫЕ пакеты будут установлены:
git git-man liberror-perl
Обновлено 0 пакетов, установлено 3 новых пакетов, для удаления отмечено 0 пакет
ов, и 68 пакетов не обновлено.
Необходимо скачать 5 465 kB архивов.
После данной операции объем занятого дискового пространства возрастёт на 38,4 М
В.
Хотите продолжить? [Д/н] у
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.
17029-1 [26,5 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all
1:2.25.1-1ubuntu3.2 [884 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:
```

Figure 3.2: рисунок 2

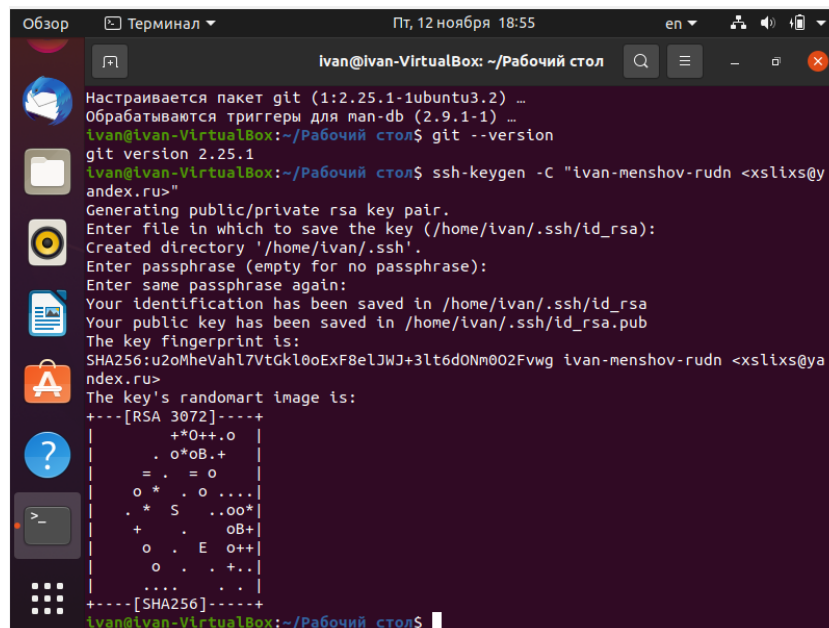


```
ов, и 68 пакетов не обновлено.
Необходимо скачать 5 465 kB архивов.
После данной операции объем занятого дискового пространства возрастёт на 38,4 М
В.
Хотите продолжить? [Д/н] у
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.
17029-1 [26,5 kB]
Пол:2 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all
1:2.25.1-1ubuntu3.2 [884 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:
2.25.1-1ubuntu3.2 [4 554 kB]
Получено 5 465 kB за 2с (3 546 kB/s)
Выбор ранее не выбранного пакета liberror-perl.
(Чтение базы данных ... на данный момент установлено 188263 файла и каталога.)
Подготовка к распаковке .../liberror-perl_0.17029-1_all.deb ...
Распаковывается liberror-perl (0.17029-1) ...
Выбор ранее не выбранного пакета git-man.
Подготовка к распаковке .../git-man_1%3a2.25.1-1ubuntu3.2_all.deb ...
Распаковывается git-man (1:2.25.1-1ubuntu3.2) ...
Выбор ранее не выбранного пакета git.
Подготовка к распаковке .../git_1%3a2.25.1-1ubuntu3.2_amd64.deb ...
Распаковывается git (1:2.25.1-1ubuntu3.2) ...
Настраивается пакет liberror-perl (0.17029-1) ...
Настраивается пакет git-man (1:2.25.1-1ubuntu3.2) ...
Настраивается пакет git (1:2.25.1-1ubuntu3.2) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
ivan@ivan-VirtualBox:~/Рабочий стол$ git --version
git version 2.25.1
ivan@ivan-VirtualBox:~/Рабочий стол$
```

Figure 3.3: рисунок 3

3. Настроим систему контроля версий git, как это указано в инструкции к 1-ой лабораторной работе с использованием сервера репозитория <https://github.com/>. Для этого необходимо сгенерировать пару ключей (приватный и открытый), а

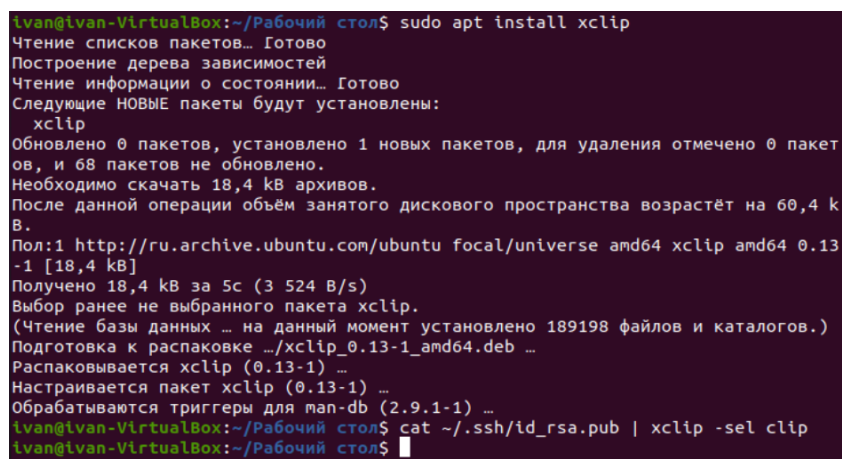
затем вставить их в SSH-ключи на github.



```
Обзор Терминал Пт, 12 ноября 18:55 en
ivan@ivan-VirtualBox: ~/Рабочий стол

Настраивается пакет git (1:2.25.1-1ubuntu3.2) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
ivan@ivan-VirtualBox:~/Рабочий стол$ git --version
git version 2.25.1
ivan@ivan-VirtualBox:~/Рабочий стол$ ssh-keygen -C "ivan-menshov-rudn <xslixs@ya
ndex.ru>"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ivan/.ssh/id_rsa):
Created directory '/home/ivan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ivan/.ssh/id_rsa
Your public key has been saved in /home/ivan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:u2oMheVahl7VtGkL0oExF8eLJWJ+3lt6dONm002Fvwg ivan-menshov-rudn <xslixs@ya
ndex.ru>
The key's randomart image is:
+---[RSA 3072]-----+
|          +*0+++.o          |
|      .  o*oB.+          |
|    = .  = o          |
|  o * .  o ....          |
| . * S  ..oo*          |
| + .  oB+          |
| o .  E o++          |
| o .  . +..          |
| ..... . .          |
+-----[SHA256]-----+
ivan@ivan-VirtualBox:~/Рабочий стол$
```

Figure 3.4: рисунок 4



```
ivan@ivan-VirtualBox:~/Рабочий стол$ sudo apt install xclip
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  xclip
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакет
ов, и 68 пакетов не обновлено.
Необходимо скачать 18,4 kB архивов.
После данной операции объем занятого дискового пространства возрастет на 60,4 k
B.
Пол:1 http://ru.archive.ubuntu.com/ubuntu focal/universe amd64 xclip amd64 0.13
-1 [18,4 kB]
Получено 18,4 kB за 5с (3 524 B/s)
Выбор ранее не выбранного пакета xclip.
(Чтение базы данных ... на данный момент установлено 189198 файлов и каталогов.)
Подготовка к распаковке ./xclip_0.13-1_amd64.deb ...
Распаковывается xclip (0.13-1) ...
Настраивается пакет xclip (0.13-1) ...
Обрабатываются триггеры для man-db (2.9.1-1) ...
ivan@ivan-VirtualBox:~/Рабочий стол$ cat ~/.ssh/id_rsa.pub | xclip -sel clip
ivan@ivan-VirtualBox:~/Рабочий стол$
```


Figure 3.5: рисунок 5



## SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



**SSH\_RUDN**  
SHA256:WZAHCSzBH7/ZTDx0CV0U2FgRsFZtYK8wr3UA9S+DfH8  
Added on 10 Nov 2021  
Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Figure 3.6: рисунок 6

4. Создадим структуру каталога лабораторных работ согласно пункту М.2.

5. Подключение репозитория к github

Создадим репозиторий на GitHub.

Рабочий каталог будем обозначать как `laboratory`. Вначале нужно перейти в этот каталог:

```
-cd laboratory
```

Инициализируем системы `git`:

```
-git init
```

Создаём заготовку для файла `README.md`:

```
-echo "# Лабораторные работы" » README.md
```

```
-git add README.md
```

Делаем первый коммит и выкладываем на github:

```
-git commit -m "first commit"
```

```
-git remote add origin git@github.com:/sciproc-intro.git
```

```
-git push -u origin master
```

Результат проделанных операций представлен ниже.

```

ivan@ivan-VirtualBox:~/Рабочий стол$ cd ~
ivan@ivan-VirtualBox:~$ cd work
ivan@ivan-VirtualBox:~/work$ cd 2021-2022
ivan@ivan-VirtualBox:~/work/2021-2022$ ls
maths_inf_sec_basics  scientific_programming
ivan@ivan-VirtualBox:~/work/2021-2022$ cd scientific_programming
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ ls
group-project  laboratory  personal-project
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ laboratory
laboratory: команда не найдена
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git init
Периинициализирован существующий репозиторий Git в /home/ivan/work/2021-2022/scientific_programming/.git/
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ echo "# Лабораторные работы" >> README.md
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git add README.md
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git commit -n "first commit"
[master 65df38b] first commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git remote add origin https://github.com/ivan-menshov-rudn/SPRUDN_2021.git
fatal: внешний репозиторий origin уже существует
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git remote add origin https://github.com/ivan-menshov-rudn/SPRUDN_2021.git
fatal: внешний репозиторий origin уже существует
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git push -u origin master

```

Figure 3.7: рисунок 7

```

ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git push -u origin master
Username for 'https://github.com': ivan-menshov-rudn
Password for 'https://ivan-menshov-rudn@github.com':
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (3/3), готово.
Запись объектов: 100% (5/5), 491 байт | 491.00 КиБ/с, готово.
Всего 5 (изменения 0), повторно использовано 0 (изменения 0)
To https://github.com/ivan-menshov-rudn/SPRUDN_2021.git
 * [new branch]      master -> master
Ветка «master» отслеживает внешнюю ветку «master» из «origin».
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$

```

Figure 3.8: рисунок 8

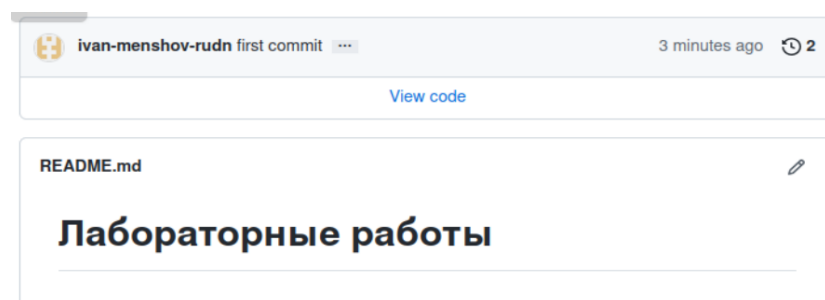


Figure 3.9: рисунок 9

## 6. Первичная конфигурация

Добавим файл лицензии:

`-wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O`

Добавим шаблон игнорируемых файлов. Просмотрим список имеющихся шаблонов: `-curl -L -s https://www.gitignore.io/api/list`

Затем скачаем шаблон, например, для C:

`-curl -L -s https://www.gitignore.io/api/c » .gitignore`

Можно это же сделать через web-интерфейс на сайте <https://www.gitignore.io/>.

Добавим новые файлы:

`-git add .` Выполним коммит:

`-git commit -a` Отправим на github:

`-git push`

Результат проделанных операций представлен ниже.

```
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ wget https://creativecommons.org/licenses/by/4.0/legalcode.txt -O
--2021-11-12 21:01:57-- https://creativecommons.org/licenses/by/4.0/legalcode.
txt
Распознаётся creativecommons.org (creativecommons.org)... 104.20.150.16, 172.67
.34.140, 104.20.151.16, ...
Подключение к creativecommons.org (creativecommons.org)[104.20.150.16]:443... с
оединение установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: нет данных [text/plain]
Сохранение в каталог: «legalcode.txt».

legalcode.txt          [ <=>          ] 18,22K  --.-KB/s   за 0,003s

2021-11-12 21:01:59 (6,88 MB/s) - «legalcode.txt» сохранён [18657]

--2021-11-12 21:01:59-- http://-/
Распознаётся - (-)... ошибка: Неизвестное имя или служба.
wget: не удастся разрешить адрес «-»
--2021-11-12 21:01:59-- http://o/
Распознаётся o (o)... ошибка: Временный сбой в разрешении имен.
wget: не удастся разрешить адрес «o»
ЗАВЕРШЕНО --2021-11-12 21:01:59--
Общее время: 1,9s
Загружено: 1 файл, 18K за 0,003s (6,88 MB/s)
```

Figure 3.10: рисунок 10

```

ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ curl -L -s https://www.gitignore.io/api/list
ic,ic-bitrix,a-frame,actionsript,ada
adobe,advancedinstaller,adventuregamestudio,agda,al
alteraquartusii,altium,amplify,android,androidstudio
angular,anjuta,ansible,ansibletower,apachecordova
apachehadoop,appbuilder,appcelerator titanium,appcode,appcode+all
appcode+inl,appengine,aptanastudio,arcanist,archive
archives,archlinuxpackages,aspnetcore,assembler,ate
atmelstudio,ats,audio,automationstudio,autotools
autotools+strict,awr,azurefunctions,azurite,backup
ballerina,basercms,basic,batch,bazaar
bazel,bitrise,bitrix,bittorrent,blackbox
bloop,bluej,bookdown,bower,bricxcc
buck,c,c++,cake,cakephp
cakephp2,cakephp3,calabash,carthage,certificates
ceylon,cfwheels,chefcookbook,chocolatey,clea
clion,clion+all,clion+inl,clojure,clovd9
cmake,cocoapods,cocos2dx,cocoscreator,codeblocks
codecomposerstudio,codeigniter,codeio,codekit,codesniffer
coffeescript,commonlisp,compodoc,composer,compressed
compressedarchive,compression,conan,concrete5,coq
cordova,craftcms,crashlytics,crbasic,crossbar
crystal,cs-cart,csharp,cuda,cvs
cypressio,d,dart,darteditor,data
database,datarecovery,dbeaver,defold,delphi
dframe,diff,direnv,diskimage,django
dm,docfx,docpress,docz,dotenv
dotfilessh,dotnetcore,dotsettings,dreamweaver,dropbox

```

Figure 3.11: рисунок 11

```

zukencr8000ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ curl -L -s https://www.gitignore.io/api/c >> .gitignore
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git add .
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git commit -a
[master 8d597c4] Heloo
 2 files changed, 455 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 legalcode.txt
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git push
Username for 'https://github.com': ivan-menshov-rudn
Password for 'https://ivan-menshov-rudn@github.com':
Перечисление объектов: 5, готово.
Подсчет объектов: 100% (5/5), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (4/4), 6.48 Киб | 552.00 Киб/с, готово.
Всего 4 (изменения 0), повторно использовано 0 (изменения 0)
To https://github.com/ivan-menshov-rudn/SPRUDN_2021.git
 65df38b..8d597c4 master -> master
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$

```

Figure 3.12: рисунок 12

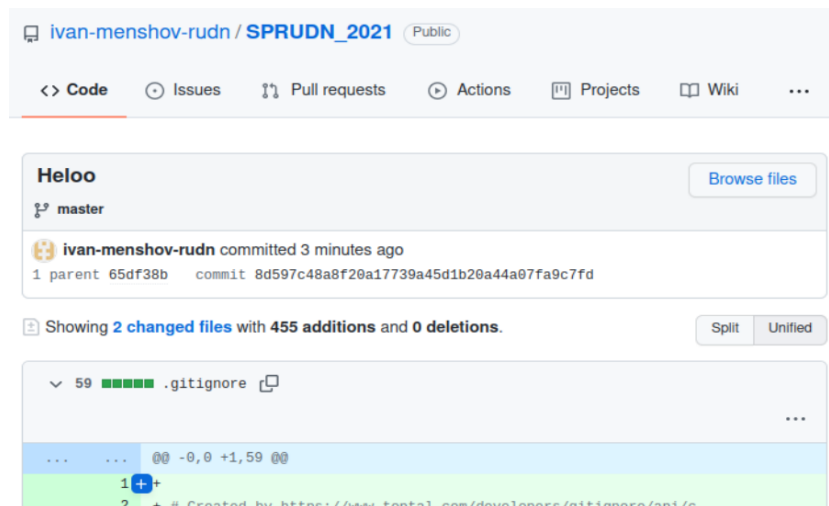


Figure 3.13: рисунок 13

## 7. Конфигурация git-flow

– Инициализируем git-flow

git flow init

Префикс для ярлыков установим в v.

– Проверьте, что Вы на ветке develop:

git branch

– Создадим релиз с версией 1.0.0

git flow release start 1.0.0

– Запишем версию:

echo "1.0.0" » VERSION – Добавим в индекс:

git add .

git commit -am 'chore(main): add version'

– Зальём релизную ветку в основную ветку

git flow release finish 1.0.0

– Отправим данные на github

git push -all

git push -tags

– Создадим релиз на github.

```

ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ 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? []
Hooks and filters directory? [/home/ivan/work/2021-2022/scientific_programming/.git/hooks]
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git branch
* develop
  master
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git flow release
start 1.0.0
M      laboratory
Переключено на новую ветку «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'

```

Figure 3.14: рисунок 14

```

ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ echo "1.0.0" >> VERSION
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git add .
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git commit -am 'chore(main): add version'
[release/1.0.0 925ecf3] chore(main): add version
1 file changed, 1 insertion(+)
create mode 100644 VERSION

```

Figure 3.15: рисунок 15

```

ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git flow release
finish 1.0.0
Branches 'master' and 'origin/master' have diverged.
And local branch 'master' is ahead of 'origin/master'.
M      laboratory
Уже на «master»
Ваша ветка опережает «origin/master» на 2 коммита.
(используйте «git push», чтобы опубликовать ваши локальные коммиты)
M      laboratory
Переключено на ветку «develop»
Merge made by the 'recursive' strategy.
VERSION | 1 +
1 file changed, 1 insertion(+)
create mode 100644 VERSION
Ветка release/1.0.0 удалена (была 925ecf3).

Summary of actions:
- Release branch 'release/1.0.0' has been merged into 'master'
- The release was tagged '1.0.0'
- Release tag '1.0.0' has been back-merged into 'develop'
- Release branch 'release/1.0.0' has been locally deleted
- You are now on branch 'develop'

```

Figure 3.16: рисунок 16

```

ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git push --all
Username for 'https://github.com': ivan-menshov-rudn
Password for 'https://ivan-menshov-rudn@github.com':
Перечисление объектов: 6, готово.
Подсчет объектов: 100% (6/6), готово.
Сжатие объектов: 100% (4/4), готово.
Запись объектов: 100% (5/5), 500 байтов | 500.00 КиБ/с, готово.
Всего 5 (изменения 3), повторно использовано 0 (изменения 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/ivan-menshov-rudn/SPRUDN_2021.git
   8d597c4..8db756f  master -> master
* [new branch]      develop -> develop
ivan@ivan-VirtualBox:~/work/2021-2022/scientific_programming$ git push --tags
Username for 'https://github.com': ivan-menshov-rudn
Password for 'https://ivan-menshov-rudn@github.com':
Перечисление объектов: 1, готово.
Подсчет объектов: 100% (1/1), готово.
Запись объектов: 100% (1/1), 163 байта | 163.00 КиБ/с, готово.
Всего 1 (изменения 0), повторно использовано 0 (изменения 0)
To https://github.com/ivan-menshov-rudn/SPRUDN_2021.git
* [new tag]         1.0.0 -> 1.0.0

```

Figure 3.17: рисунок 17

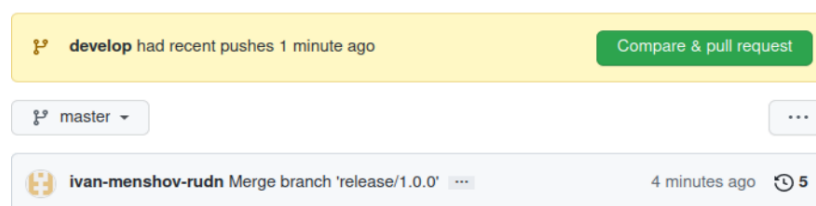


Figure 3.18: рисунок 18

## 4 Выводы

Были изучена идеология и применение средств контроля версий. Также был освоен язык разметки Markdown.



## 5 Контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

- Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом.

2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

- Хранилище (repository), или репозиторий, — место хранения всех версий и служебной информации. Commit («[трудовой] вклад», не переводится) — синоним версии; процесс создания новой версии. Рабочая копия (working copy) — текущее состояние файлов проекта, основанное на версии, загруженной из хранилища (обычно на последней).

3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.

- Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществлялся через специальное клиентское приложение. В качестве примеров таких программных продуктов можно привести CVS, Subversion. аспределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом

можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ним же эти локальные репозитории будут синхронизироваться. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией. После внесения достаточного количества изменений в локальную копию они (изменения) отправляются на сервер. При этом сервер, чаще всего, выбирается условно, т.к. в большинстве DVCS нет такого понятия как “выделенный сервер с центральным репозиторием”.

4. Опишите действия с VCS при единоличной работе с хранилищем.

5. Опишите порядок работы с общим хранилищем VCS.

6. Каковы основные задачи, решаемые инструментальным средством git?

- Возврат к любой версии кода из прошлого. Просмотр истории изменений. Совместная работа без боязни потерять данные или затереть чужую работу.

7. Назовите и дайте краткую характеристику командам git.

- Наиболее часто используемые команды git:

- создание основного дерева репозитория:

`git init`

- получение обновлений (изменений) текущего дерева из центрального репозитория: `git pull`

- отправка всех произведённых изменений локального дерева в центральный репозиторий:

`git push` – просмотр списка изменённых файлов в текущей директории: `git status`

- просмотр текущих изменений:

`git diff`

- добавить все изменённые и/или созданные файлы и/или каталоги: `git add` .

– добавить конкретные изменённые и/или созданные файлы и/или каталоги:  
git add имена\_файлов – удалить файл и/или каталог из индекса репозитория  
(при этом файл и/или каталог остаётся в локальной директории):

git rm имена\_файлов

– сохранение добавленных изменений:

– сохранить все добавленные изменения и все изменённые файлы: git commit  
-am 'Описание коммита'

– сохранить добавленные изменения с внесением комментария через встроен-  
ный редактор:

git commit

– создание новой ветки, базирующейся на текущей: git checkout -b имя\_ветки

– переключение на некоторую ветку:

git checkout имя\_ветки

– отправка изменений конкретной ветки в центральный репозиторий:

git push origin имя\_ветки

– слияние ветки с текущим деревом:

git merge --no-ff имя\_ветки

8. Приведите примеры использования при работе с локальным и удалённым  
репозиториями.

- См пункты 1.3.3 -1.3.4

9. Что такое и зачем могут быть нужны ветви (branches)?

- Ветки нужны для того, чтобы программисты могли вести совместную работу  
над проектом и не мешать друг другу при этом.

10. Как и зачем можно игнорировать некоторые файлы при commit?

- Игнорируемые файлы — это, как правило, артефакты сборки и файлы, ге-  
нерируемые машиной из исходных файлов в вашем репозитории, либо файлы,  
которые по какой-либо иной причине не должны попадать в коммиты. Для этого  
нужно указать название все игнорируемых файлов в файле с названием .gitignore