

# **Лабораторная работа №2**

Симонова Виктория Игоревна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
4.1	Установка ПО . . . . .	9
4.2	Создание ключей SSH . . . . .	10
4.3	Создание ключа PGP . . . . .	10
4.4	Добавление ключа PGP в Github . . . . .	11
4.5	Настройка автоматических подписей . . . . .	13
4.6	Создание репозитория курса на основе шаблона . . . . .	14
<b>5</b>	<b>Выводы</b>	<b>16</b>
<b>6</b>	<b>Ответы на контрольные вопросы</b>	<b>17</b>
	<b>Список литературы</b>	<b>20</b>

## Список иллюстраций

4.1	Установка git и gh . . . . .	9
4.2	email, name, utf, master, params . . . . .	9
4.3	генерация по алгоритму rsa . . . . .	10
4.4	генерация по алгоритму ed25529 . . . . .	10
4.5	PGP . . . . .	11
4.6	Вывод списка ключей . . . . .	11
4.7	Копирование . . . . .	12
4.8	Мой аккаунт . . . . .	12
4.9	Добавление нового ключа . . . . .	12
4.10	Добавленный ключ . . . . .	13
4.11	Авторизация и завершение авторизации в gh через браузер . . .	13
4.12	Авторизация и завершение авторизации в gh . . . . .	13
4.13	Основа шаблона . . . . .	14
4.14	Клонирование . . . . .	14
4.15	Удаление и создание . . . . .	14
4.16	Отправка файлов на сервер . . . . .	15
4.17	Отправка на сервер . . . . .	15

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - изучение применения средств контроля версий git.

## 2 Задание

1. Создать базовую конфигурацию для работы с git
2. Создать ключь SHH
3. Создать ключь GPG
4. Настроить подписи к коммитам
5. Зарегистрироваться на github
6. Создать локальный каталог для выполнения заданий по предмету

### 3 Теоретическое введение

Системы контроля версий. Общие понятия Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или

заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.



## 4 Выполнение лабораторной работы

### 4.1 Установка ПО

Устанавливаю необходимое ПО git и gh (рис. 4.1).

```
[visimonova@visimonova ~]$ sudo dnf install git
Последняя проверка окончания срока действия метаданных: 3:04:58 назад, Ср 28 фев 2024 21:30:38.
Пакет git-2.44.0-1.fc39.x86_64 уже установлен.
Зависимости разрешены.
Нет действий для выполнения.
Выполнено!
[visimonova@visimonova ~]$ sudo dnf install gh
Последняя проверка окончания срока действия метаданных: 3:06:13 назад, Ср 28 фев 2024 21:30:38.
Зависимости разрешены.
=====
Пакет      Архитектура  Версия      Резпозиторий  Размер
=====
Установка:
gh         x86_64       2.43.1-1.fc39  updates      9.1 М
=====
Результат транзакции
=====
Установка 1 Пакет
=====
Объем загрузки: 9.1 М
Объем изменений: 46 М
Продолжить? [д/н]: y
Загрузка пакетов:
gh-2.43.1-1.fc39.x86_64.rpm                3.8 MB/s | 9.1 MB  00:02
-----
Общий размер                                3.0 MB/s | 9.1 MB  00:03
Проверка транзакции
Проверка транзакции успешно завершена.
Идет проверка транзакции
Тест транзакции проведен успешно.
Выполнение транзакции
Подготовка      :
Установка       : gh-2.43.1-1.fc39.x86_64      1/1
Запуск скрипта  : gh-2.43.1-1.fc39.x86_64      1/1
```

Рис. 4.1: Установка git и gh

Задаю для имя и почту владельца репозитория. Настраиваю кодировку для корректного вывода сообщений в git. Задаю имя ветки. Параметр autoclaf, параметр safecrlf (рис. 4.2).

```
[visimonova@visimonova ~]$ git config --global user.name "Victoria Simonova"
[visimonova@visimonova ~]$ git config --global user.email "vskas2008.vs@gmail.com"
[visimonova@visimonova ~]$ git config --global core.quotepath false
[visimonova@visimonova ~]$ git config --global init.defaultBranch master
[visimonova@visimonova ~]$ git config --global core.autocrlf input
[visimonova@visimonova ~]$ git config --global core.safecrlf warn
```

Рис. 4.2: email, name, utf, master, parametrs

## 4.2 Создание ключей SSH

Создание ключа SSH размер 4096 бит по алгоритму rsa (рис. 4.3).

```
[visimonova@visimonova ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/visimonova/.ssh/id_rsa):
Created directory '/home/visimonova/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/visimonova/.ssh/id_rsa
Your public key has been saved in /home/visimonova/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:f0eISMmf6CuIBu9CvqwfNuHABBOMYlRrlx3mPnIDEec visimonova@visimonova.net
The key's randomart image is:
+---[RSA 4096]-----+
|Bo.. o.+ |
|++ . O . |
|o. o + E |
|o . . o. |
|... ..=S . |
|. . . o+o+ + |
|oo= . . = . |
|oo+o. . o . |
|oB+ . o+.o |
+----[SHA256]-----+
```

Рис. 4.3: генерация по алгоритму rsa

Создание ключа SSH по алгоритму ed25519 (рис. 4.4).

```
[visimonova@visimonova ~]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/visimonova/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/visimonova/.ssh/id_ed25519
Your public key has been saved in /home/visimonova/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:c2h87h/qEfIsU7PGDLxJKP659Y+E8RHhNM2xBmht2H0 visimonova@visimonova.net
The key's randomart image is:
+--[ED25519 256]--+
|      =B+.. |
|      ==+.+E |
|      . . = O. |
|      o . = |
|      . . * S o |
|      . . / X . |
|      . B % . |
|      . o B o |
|      +. +.. |
+----[SHA256]-----+
```

Рис. 4.4: генерация по алгоритму ed25519

## 4.3 Создание ключа PGP

Генерирую ключ PGP выбираю тип RSA and RSA, задаю максимальную длину ключа 4096, ставлю неограниченный срок действия. Отвечаю на вопросы о личной информации и ввожу пароль для защиты новго ключа. (рис. 4.5).

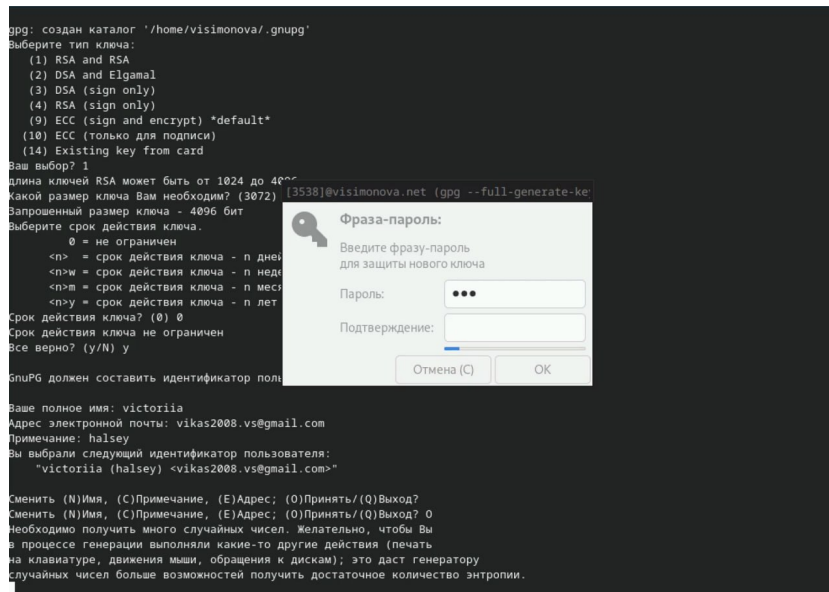


Рис. 4.5: PGP

## 4.4 Добавление ключа PGP в Github

Добавляю список созданных ключей , ищу отпечаток ключа , копирую (рис. 4.6).

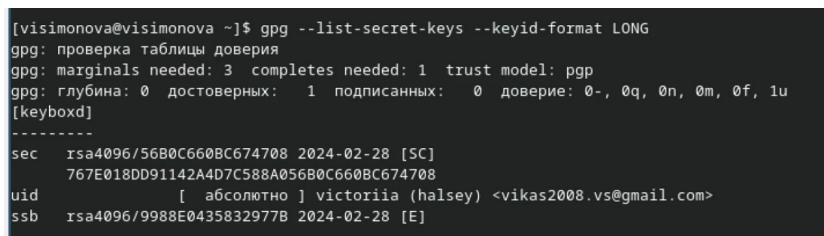



Рис. 4.6: Вывод списка ключей

Копирую ключ (рис. 4.7).

Рис. 4.7: Копирование

The screenshot displays the GitHub profile of a user named ViSimonova. The profile picture is a circular avatar with a pink and white geometric design. The page shows the user's public repositories, with 'study\_2023-2024\_anti-pc' listed as the only one, and a bar chart showing 75 contributions in the last year.

Рис. 4.8: Мой аккаунт



Vishimonova (Vishimonova)

Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Add new GPG key

Title

Zsmr

Key

xUkuzmYgKZJXUqAqWfMLTNgEo+ wFw1tLwvW/C1D3ApGJCjyH1ShdHwVwQp0B  
OGs7hgxezCyrddQbwGatc4lOUzwMtG2q4ctidckEDfIWQyzbQJ4tjWNLEPazMnj  
36icCbZVQIqWNNWCoJaGueTnf+ 1NBCuo59ja87DFPKimSOL1W/ShH/TnuR844  
cRLytmcvgqvLVH7VSAhXX++3wftTxmRzLYWwWdgIkgeleqfQumFLd3H9tVUB  
PDwMeK6SPphd7GL6vARhewczj15SRpmfALCGepfzL2zQ1SDHdGhdGofdyo  
MGZ2jhfdnALGLPVSIEKhZtkN869kyushshmmLPWQqSiRWMSWeOkOCr+  
dlBZtH6actBCJVLwgNik8hmOK7khrRBdlVRgmMH3KRgGPu6as7oeTmlZh1h1f1f  
/QK2hNE5nd7eTA==  
=2UMp  
-----END PGP PUBLIC KEY BLOCK-----

Add GPG key

Рис. 4.9: Добавление нового ключа

Я добавила ключ на Github (рис. 4.10).

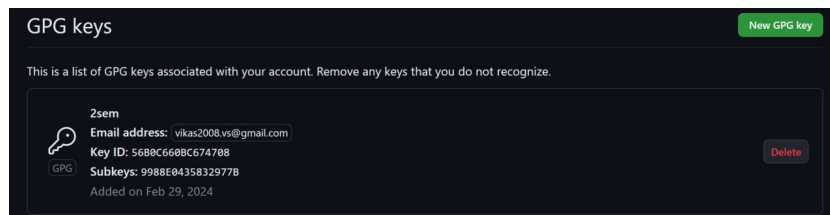


Рис. 4.10: Добавленный ключ

## 4.5 Настройка автоматических подписей

Настройка автоматических подписей комитов с помощью введённого эмэйла (рис. 4.11).

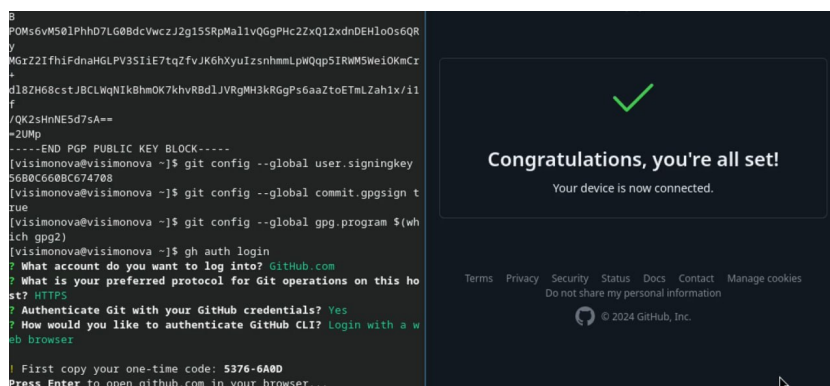


Рис. 4.11: Авторизация и завершение авторизации в gh через браузер

Сообщение о завершении авторизации (рис. 4.12).

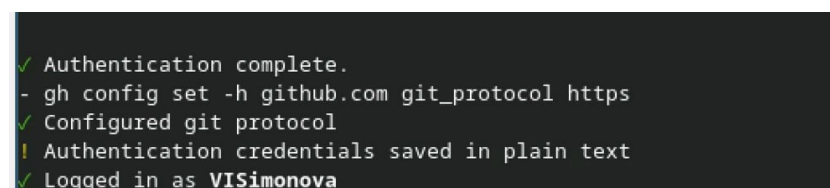


Рис. 4.12: Авторизация и завершение авторизации в gh

## 4.6 Создание репозитория курса на основе шаблона

Создаю свою директорию с помощью `mkdir` и перехожу в нее. Далее в терминале ввожу команду, чтобы создать репозиторий на основе заданного шаблона (рис. 4.13).

```
[visimonova@visimonova ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"/os-intro
[visimonova@visimonova ~]$ cd ~/work/study/2022-2023/"Операционные системы"
[visimonova@visimonova ~]$ cd ~/work/study/2022-2023/"Операционные системы"
[visimonova@visimonova Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadrama/course-directory-student-template --public
GraphQL: Could not resolve to a Repository with the name 'yamadrama/course-directory-student-template'. (repository)
[visimonova@visimonova Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadrama/course-directory-student-template --public
GraphQL: Could not resolve to a Repository with the name 'yamadrama/course-directory-student-template'. (repository)
[visimonova@visimonova Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
Created repository VISimonova/study_2022-2023_os-intro on GitHub
https://github.com/VISimonova/study_2022-2023_os-intro
[visimonova@visimonova Операционные системы]$ git clone --recursive git@github.com:owner/study_2022-2023_os-intro.git os-intro
bash: owner: Нет такого файла или каталога
[visimonova@visimonova Операционные системы]$ ls
os-intro
```

Рис. 4.13: Основа шаблона

Клонирую репозиторий к себе в директорию и переход туда (рис. 4.14).

```
[visimonova@visimonova Операционные системы]$ git clone --recursive https://github.com/VISimonova/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 32, done.
remote: Counting objects: 100% (32/32), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 32 (delta 1), reused 18 (delta 0), pack-reused 0
Получение объектов: 100% (32/32), 18.60 Киб | 218.00 Киб/с, готово.
Определение изменений: 100% (1/1), готово.
Подмодуль «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Подмодуль «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/visimonova/work/study/2022-2023/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 95, done.
remote: Counting objects: 100% (95/95), done.
remote: Compressing objects: 100% (67/67), done.
remote: Total 95 (delta 34), reused 87 (delta 26), pack-reused 0
Получение объектов: 100% (95/95), 96.09 Киб | 247.00 Киб/с, готово.
Определение изменений: 100% (34/34), готово.
Клонирование в «/home/visimonova/work/study/2022-2023/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 126, done.
remote: Counting objects: 100% (126/126), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 126 (delta 52), reused 108 (delta 34), pack-reused 0
Получение объектов: 100% (126/126), 335.00 Киб | 629.00 Киб/с, готово.
Определение изменений: 100% (52/52), готово.
Submodule path 'template/presentation': checked out '40a1761813e197d00e8443ff1ca72c68a304f24c'
Submodule path 'template/report': checked out '7c31ab8e5dfa8c8db2d67caeb8a19ef8028ced88e'
[visimonova@visimonova Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/os-intro
```

Рис. 4.14: Клонирование

Удаляю лишние файлы и создаю необходимые каталоги (рис. 4.15).

```
[visimonova@visimonova Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/os-intro
[visimonova@visimonova os-intro]$ rm package.json
[visimonova@visimonova os-intro]$ echo os-intro > COURSE
[visimonova@visimonova os-intro]$ make
Usage:
  make <target>

Targets:
  list           List of courses
  prepare        Generate directories structure
  submodule      Update submodules
[visimonova@visimonova os-intro]$
```

Рис. 4.15: Удаление и создание

Выполняю git add для отправки файлов на сервер , git commit для их комментирования ,далее ввожу пароль для разблокировки ключа (рис. 4.16).

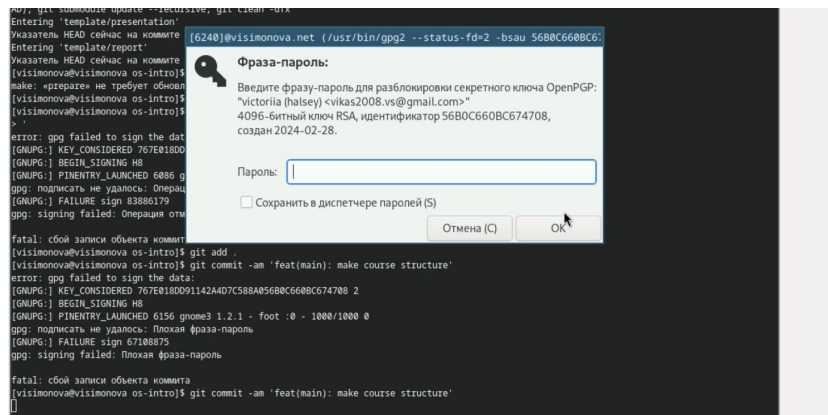


Рис. 4.16: Отправка файлов на сервер

Отправляю файлы на сервер с помощью git push (рис. 4.17).

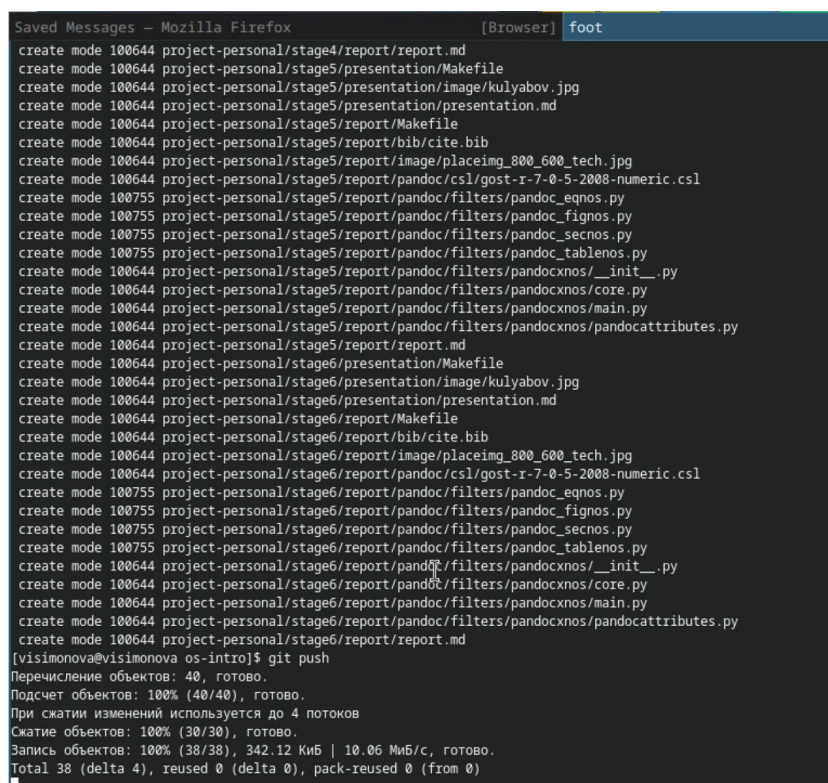


Рис. 4.17: Отправка на сервер

## 5 Выводы

При выполнении лабораторной работы я изучила систему контроля версий git и применение средств контроля версий.



## 6 Ответы на контрольные вопросы

1. Системы контроля версий - ПО для работы с изменяющейся информацией и сохранения её версий. Может использоваться для работы нескольких человек над проектом, позволяет посмотреть кто какие изменения закомит(полная история).
2. Хранилище -репозиторий, где хранятся версии и история всех изменений проекта. commit - позволяет отслеживать историю изменений, тк позволяет обратиться к нужным данным. Рабочая копия - это основная копия проека.

3.Централизованные VCS (например: CVS, TFS, AccuRev) – одно основное хранилище всего проекта. Каждый пользователь копирует себе необходимые ему файлы из этого репозитория, изменяет, затем добавляет изменения обратно в хранилище. Децентрализованные VCS (например: Git, Bazaar) – у каждого пользователя свой вариант репозитория (возможно несколько вариантов), есть возможность добавлять и забирать изменения из любого репозитория. В отличие от классических, в распределенных (децентрализованных) системах контроля версий центральный репозиторий не является обязательным.

4. Сначала создается и подключается удаленный репозиторий, затем по мере изменения проекта эти изменения отправляются на сервер.
5. Участник проекта перед началом работы получает нужную ему версию проекта в хранилище, с помощью определенных команд, после внесения изменений пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются. К ним можно вернуться в любой момент.

6. Хранение информации о всех изменениях в вашем коде, обеспечение удобства командной работы над кодом.

7. 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`

Сохранение текущих изменений: 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 встроенный редактор: `git commit`

создание новой ветки, базирующейся на текущей: `git checkout -b имя_ветки`

переключение на некоторую ветку: `git checkout имя_ветки` (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)

отправка изменений конкретной ветки в центральный репозиторий: `git push origin имя_ветки`

слияние ветки с текущим деревом: `git merge --no-ff имя_ветки`

Удаление ветки:

удаление локальной уже слитой с основным деревом ветки: `git branch -d имя_ветки`

принудительное удаление локальной ветки: `git branch -D имя_ветки`

удаление ветки с центрального репозитория: `git push origin :имя_ветки`

8. `git push -all` отправляем из локального репозитория все сохраненные изменения в центральный репозиторий, предварительно создав локальный репозиторий и сделав предварительную конфигурацию.
9. Ветвление - один из параллельных участков в одном хранилище, исходящих из одной версии, обычно есть главная ветка. Между ветками, т. е. их концами возможно их слияние. Используются для разработки новых функций.
10. Во время работы над проектом могут создаваться файлы, которые не следуют добавлять в репозиторий. Например, временные файлы. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл `.gitignore` с помощью сервисов.

## Список литературы

1. Лабораторная работа № 2 [Электронный ресурс] URL: <https://esystem.rudn.ru/mod/page/view>