

Отчёт по лабораторной работе №2

Управление версиями

Содикжонов Темурбек НПИбд-01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	9
4	Контрольные вопросы	10
	Список литературы	14

List of Figures

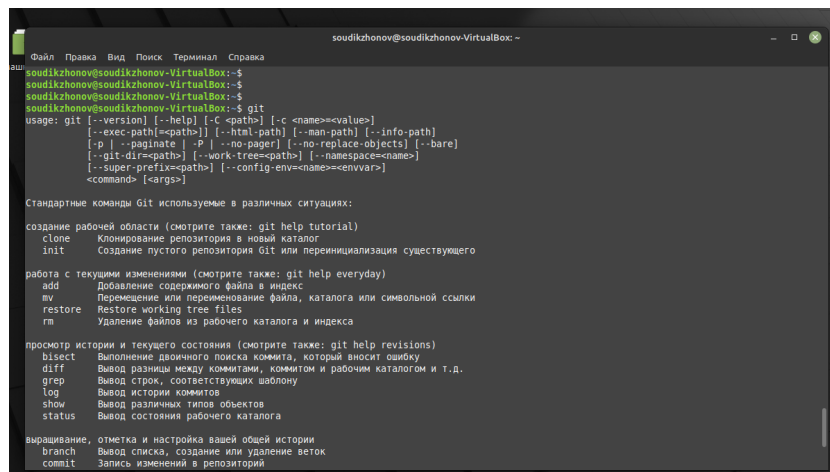
2.1	Загрузка пакетов	5
2.2	Параметры репозитория	5
2.3	rsa-4096	6
2.4	ed25519	6
2.5	GPG ключ	6
2.6	GPG ключ	7
2.7	Параметры репозитория	7
2.8	Связь репозитория с аккаунтом	8
2.9	Загрузка шаблона	8
2.10	Первый коммит	8

1 Цель работы

Целью данной работы является изучение идеологии и применения средств контроля версий и освоение умений работать с git.

2 Выполнение лабораторной работы

Устанавливаем git, git-flow и gh.



```
soudikzhonov@soudikzhonov-VirtualBox:~$ git help
usage: git [-version] [-help] [-c <name>=<value>]
          [--exec-path<path>] [--html-path] [--man-path] [--info-path]
          [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir<path>] [--work-tree<path>] [--namespace<name>]
          [--super-prefix<path>] [--config-env<name>=<envvar>]
          <command> [<args>]

Стандартные команды Git используются в различных ситуациях:

создание рабочей области (смотрите также: git help tutorial)
  clone  Клонирование репозитория в новый каталог
  init   Создание пустого репозитория git или перенициализация существующего

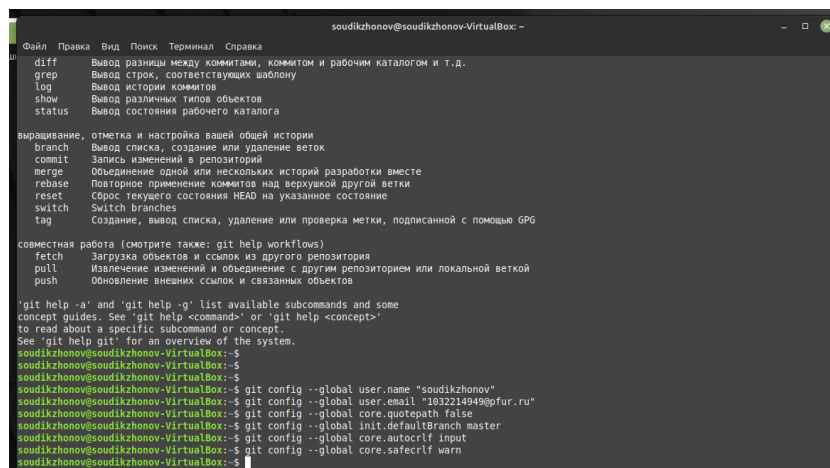
работа с текущими изменениями (смотрите также: git help everyday)
  add    Добавление содержимого файла в индекс
  mv     Перемещение или переименование файла, каталога или символической ссылки
  restore Restore working tree files
  rm     Удаление файлов из рабочего каталога и индекса

просмотр истории и текущего состояния (смотрите также: git help revisions)
  bisect Выполнение двоичного поиска коммита, который вносит ошибку
  diff   Вывод разницы между коммитами, коммитом и рабочим каталогом и т.д.
  grep   Вывод строк, соответствующих шаблону
  log    Вывод истории коммитов
  show   Вывод различных типов объектов
  status Вывод состояния рабочего каталога

выращивание, отметка и настройка вашей общей истории
  branch Вывод списка, создание или удаление веток
  commit Запись изменений в репозиторий
```

Figure 2.1: Загрузка пакетов

Зададим имя и email владельца репозитория, кодировку и прочие параметры.



```
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global user.name "soudikzhonov"
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global user.email "1032214949@pfur.ru"
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global core.quotePath false
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global init.defaultBranch master
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global core.autocrlf input
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global core.safecrlf warn
```

Figure 2.2: Параметры репозитория

Создаем SSH ключи

```
soudikzhonov@soudikzhonov-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/soudikzhonov/.ssh/id_rsa):
Created directory /home/soudikzhonov/.ssh/.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/soudikzhonov/.ssh/id_rsa
Your public key has been saved in /home/soudikzhonov/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:9AMX7wJlVsf3o+CmLCmHlRqQ67IDFbP847ALCKSBY soudikzhonov@soudikzhonov-VirtualBox
The key's randomart image is:
+--[RSA 4096]-----+
| .E               |
| o.   o           |
| .o.   + =        |
| +.   = o = o     |
| +   o B S . B =  |
|   % O   + B O    |
|   * @ o o o +    |
|   . o + . o o .  |
|   .   .   .   .  |
|   .   .   .   .  |
+-----[SHA256]-----+
soudikzhonov@soudikzhonov-VirtualBox:~$
```

Figure 2.3: rsa-4096

```
soudikzhonov@soudikzhonov-VirtualBox:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/soudikzhonov/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/soudikzhonov/.ssh/id_ed25519
Your public key has been saved in /home/soudikzhonov/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:kncIKha7vbc+G6YlGNbpfe+nFX9qh5oN3fc5IMvw soudikzhonov@soudikzhonov-VirtualBox
The key's randomart image is:
+--[ED25519 256]---+
| . .              |
| o.o              |
| o.o+o            |
| . o..+oo.        |
| .o+5..   .       |
| o+o ...   .      |
| .o.o .+ o = +    |
| ..o+.o + +o     |
| .o+o*..+o+E+    |
+-----[SHA256]-----+
soudikzhonov@soudikzhonov-VirtualBox:~$
```

Figure 2.4: ed25519

Создаем GPG ключ

```
судик
Файл Правка Вид Поиск Терминал Справка
судикzhonov@soudikzhonov-VirtualBox:~$ gpg --full-gen-key
gpg: на клавиатуре, движения мыши, обращения к дискам); это даст генератору
случайных чисел больше возможностей получить достаточное количество энтропии.
gpg: /home/soudikzhonov/.gnupg/trustdb.gpg: создана таблица доверия
gpg: ключ BE4B271E150820BF помечен как абсолютно доверенный
gpg: создан каталог /home/soudikzhonov/.gnupg/openpgp-revocs.d/
gpg: сертификат отныне записан в /home/soudikzhonov/.gnupg/openpgp-revocs.d/48D3ED1EF70838335828A052BE4B271E150820BF.rev'.
Открытый и секретный ключи созданы и подписаны.

pub   rsa4096 2022-08-29 [SC]
      48D3ED1EF70838335828A052BE4B271E150820BF
uid          soudikzhonov <1032214949@pfur.ru>
sub   rsa4096 2022-08-29 [E]

soudikzhonov@soudikzhonov-VirtualBox:~$ gpg --list-secret-keys --keyid-format LONG
soudikzhonov@soudikzhonov-VirtualBox:~$ gpg --check-trustdb
gpg: проверка таблицы доверия
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: гноуина: 0 доверенных: 1 подписанных: 0 Доверие: 0-, 0q, 0n, 0m, 0f, 1u
/home/soudikzhonov/.gnupg/pubring.kbx
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 1.4.1
Comment: GnuPG v2.2.40
mQINBGMpGABEAC6dVNs6pNBYV33+E/OC0w1u0/y+Zdr0E41jR+hF05LjK+6LI
Tf2h1aC/Wk1duzuzQ0uAYtJf6b35GrK9BKUvns45fM02945a8019HTW3fJara
Ug5o8bTtVADbryVfH32VnUf1rVtYctnqdy4fcmKTKpnkds8BwdF2yTtLJH566
-----END PGP PUBLIC KEY BLOCK-----

soudikzhonov@soudikzhonov-VirtualBox:~$ gpg --armor --export BE4B271E150820BF
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 1.4.1
Comment: GnuPG v2.2.40
mQINBGMpGABEAC6dVNs6pNBYV33+E/OC0w1u0/y+Zdr0E41jR+hF05LjK+6LI
Tf2h1aC/Wk1duzuzQ0uAYtJf6b35GrK9BKUvns45fM02945a8019HTW3fJara
Ug5o8bTtVADbryVfH32VnUf1rVtYctnqdy4fcmKTKpnkds8BwdF2yTtLJH566
-----END PGP PUBLIC KEY BLOCK-----
```

Figure 2.5: GPG ключ

Добавляем GPG ключ в аккаунт

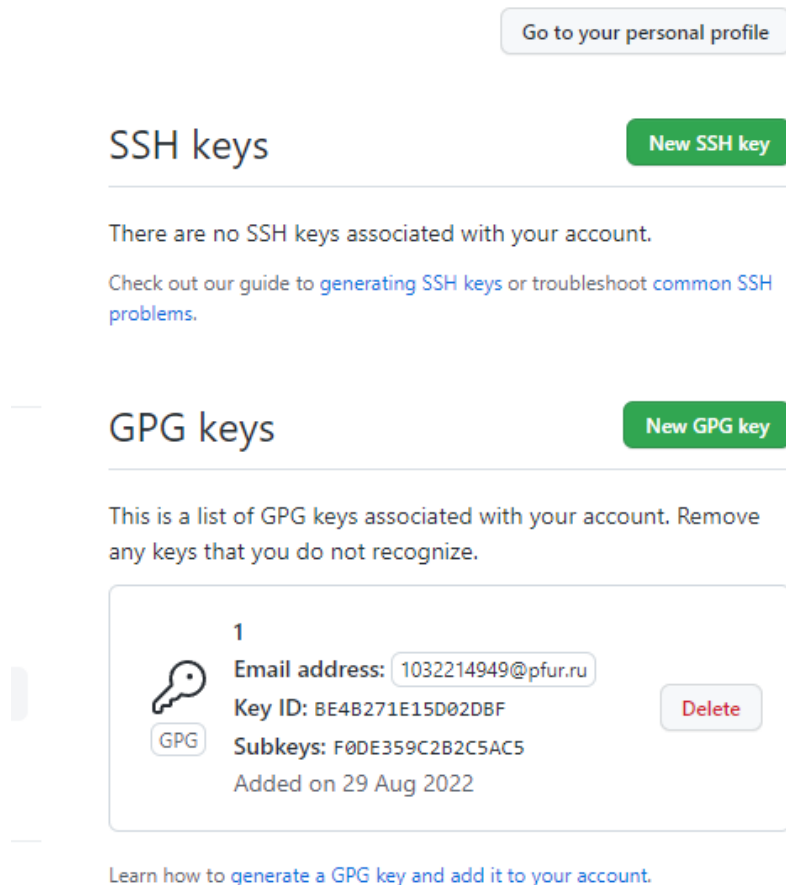


Figure 2.6: GPG ключ

Настройка автоматических подписей коммитов git

```
soudikzhonov@soudikzhonov-VirtualBox:~$  
soudikzhonov@soudikzhonov-VirtualBox:~$  
soudikzhonov@soudikzhonov-VirtualBox:~$  
soudikzhonov@soudikzhonov-VirtualBox:~$  
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global user.signingkey BE4B271E15D02DBF  
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global commit.gpgsign true  
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global gpg.program $(which gpg2)  
soudikzhonov@soudikzhonov-VirtualBox:~$
```

Figure 2.7: Параметры репозитория

Настройка gh

```
soudikzhonov@soudikzhonov-VirtualBox:~$ git config --global gpg.program $(which gpg2)
soudikzhonov@soudikzhonov-VirtualBox:~$ gh auth login
? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations? SSH
? Upload your SSH public key to your GitHub account? /home/soudikzhonov/.ssh/id_rsa.pub
? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Login with a web browser

| First copy your one-time code: DF38-C3AC
Press Enter to open github.com in your browser...
2022-08-29T11:38:56Z ERROR viaduct::backend::ffi Missing HTTP status
2022-08-29T11:38:56Z ERROR viaduct::backend::ffi Missing HTTP status
Authentication complete.
- gh config set -h github.com git_protocol ssh
  Configured git protocol
Uploaded the SSH key to your GitHub account: /home/soudikzhonov/.ssh/id_rsa.pub
Logged in as soudikzhonov
soudikzhonov@soudikzhonov-VirtualBox:~$
soudikzhonov@soudikzhonov-VirtualBox:~$
```

Figure 2.8: Связь репозитория с аккаунтом

Загрузка шаблона репозитория и синхронизация

```
soudikzhonov@soudikzhonov-VirtualBox: ~/work/study/2021-2022/Операционные системы
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы$ git clone --recursive git@github.com:soudikzhonov/study_2021-2022-os-intro.git os-intro
Клонирование в «os-intro»...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:diY3vvv6TujJhbpZlsF/zLDA8zPMSvHdkr4UvCQqU.
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.
remote: Enumerating objects: 20, done.
remote: Counting objects: 100% (20/20), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 2), reused 15 (delta 2), pack-reused 0
Получение объектов: 100% (20/20), 12.49 КБ | 2.50 МБ/с, готово.
Определение изменений: 100% (2/2), готово.
Получили «template/presentation» (https://github.com/yamadharma/academic-presentation-markdown-template.git) зарегистрирован по пути «template/presentation»
Получили «template/report» (https://github.com/yamadharma/academic-laboratory-report-template.git) зарегистрирован по пути «template/report»
Клонирование в «/home/soudikzhonov/work/study/2021-2022/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (71/71), done.
remote: Compressing objects: 100% (49/49), done.
remote: Total 71 (delta 23), reused 68 (delta 20), pack-reused 0
Получение объектов: 100% (71/71), 88.89 КБ | 14.81 МБ/с, готово.
Определение изменений: 100% (23/23), готово.
Клонирование в «/home/soudikzhonov/work/study/2021-2022/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 78, done.
remote: Counting objects: 100% (78/78), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 78 (delta 31), reused 69 (delta 22), pack-reused 0
Получение объектов: 100% (78/78), 292.27 КБ | 3.52 МБ/с, готово.
Определение изменений: 100% (31/31), готово.
Submodule path 'template/presentation': checked out '3eae6b7586f8a9aded2b506cd1018e625b228093'
Submodule path 'template/report': checked out 'df7b2ef80f8def3b9a496f8695277469a1a7842a'
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы$
```

Figure 2.9: Загрузка шаблона

Подготовка репозитория и коммит изменений

```
soudikzhonov@soudikzhonov-VirtualBox: ~/work/study/2021-2022/Операционные системы/os-intro
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ cd ~/work/study/2021-2022/Операционные системы/os-intro
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ rm package.json
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ make COURSE=os-intro
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git add .
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git commit -am 'feat(main): make course structure'
[master 1b0d3ed] feat(main): make course structure
16 files changed, 1580 insertions(+), 14 deletions(-)
create mode 100644 labs/lab(01..15)/presentation/Makefile
create mode 100644 labs/lab(01..15)/presentation/presentation.md
create mode 100644 labs/lab(01..15)/report/Makefile
create mode 100644 labs/lab(01..15)/report/bib/cite.bib
create mode 100644 labs/lab(01..15)/report/image/placeimg_800_600_tech.jpg
create mode 100644 labs/lab(01..15)/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 labs/lab(01..15)/report/report.md
delete mode 100644 package.json
create mode 100644 project-personal/stage(1..6)/presentation/Makefile
create mode 100644 project-personal/stage(1..6)/presentation/presentation.md
create mode 100644 project-personal/stage(1..6)/report/Makefile
create mode 100644 project-personal/stage(1..6)/report/bib/cite.bib
create mode 100644 project-personal/stage(1..6)/report/image/placeimg_800_600_tech.jpg
create mode 100644 project-personal/stage(1..6)/report/pandoc/csl/gost-r-7-0-5-2008-numeric.csl
create mode 100644 project-personal/stage(1..6)/report/report.md
create mode 100644 structure
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$ git push
Пересчет объектов: 100% (20/20), готово.
При скатии изменений используется до 2 потоков
Сжатие объектов: 100% (14/14), готово.
Запись объектов: 100% (19/19), 266.46 КБ | 2.15 МБ/с, готово.
Всего 19 (изменений 2), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:soudikzhonov/study_2021-2022-os-intro.git
  05d69d2..1b0d3ed master -> master
soudikzhonov@soudikzhonov-VirtualBox:~/work/study/2021-2022/Операционные системы/os-intro$
```

Figure 2.10: Первый коммит

3 Вывод

Мы приобрели практические навыки работы с сервисом github.

4 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. 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

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

- хранилище - пространство на накопителе где расположен репозиторий
- commit - сохранение состояния хранилища
- история - список изменений хранилища (коммитов)
- рабочая копия - локальная копия сетевого репозитория, в которой работает программист. 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.

Распределенные 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.

Один пользователь работает над проектом и по мере необходимости делает коммиты, сохраняя определенные этапы.

5. Опишите порядок работы с общим хранилищем VCS.

Несколько пользователей работают каждый над своей частью проекта. При этом каждый должен работать в своей ветки. При завершении работы ветка пользователя сливается с основной веткой проекта.

6. Каковы основные задачи, решаемые инструментальным средством git?

- Ведение истории версий проекта: журнал (log), метки (tags), ветвления (branches).

- Работа с изменениями: выявление (diff), слияние (patch, merge).
- Обеспечение совместной работы: получение версии с сервера, загрузка обновлений на сервер.

7. Назовите и дайте краткую характеристику командам git.

- git config - установка параметров
- git status - полный список изменений файлов, ожидающих коммита
- git add . - сделать все измененные файлы готовыми для коммита.
- git commit -m "[descriptive message]" - записать изменения с заданным сообщением.
- git branch - список всех локальных веток в текущей директории.
- git checkout [branch-name] - переключиться на указанную ветку и обновить рабочую директорию.
- git merge [branch] — соединить изменения в текущей ветке с изменениями из заданной.
- git push - запустить текущую ветку в удаленную ветку.
- git pull - загрузить историю и изменения удаленной ветки и произвести слияние с текущей веткой.

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

- git remote add [имя] [url] — добавляет удалённый репозиторий с заданным именем;
- git remote remove [имя] — удаляет удалённый репозиторий с заданным именем;
- git remote rename [старое имя] [новое имя] — переименовывает удалённый репозиторий;
- git remote set-url [имя] [url] — присваивает репозиторию с именем новый адрес;

- `git remote show [имя]` — показывает информацию о репозитории.

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

Ветвление — это возможность работать над разными версиями проекта: вместо одного списка с упорядоченными коммитами история будет расходиться в определённых точках. Каждая ветвь содержит легковесный указатель HEAD на последний коммит, что позволяет без лишних затрат создать много веток. Ветка по умолчанию называется `master`, но лучше назвать её в соответствии с разрабатываемой в ней функциональностью.

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

Зачастую нам не нужно, чтобы Git отслеживал все файлы в репозитории, потому что в их число могут входить:

Список литературы

1. Лекция Системы контроля версий
2. GitHub для начинающих