

Отчет по лабораторной работе №2

Дисциплина: Архитектура компьютера

Малюга Валерия Васильевна

Содержание

1	Цель работы	1
2	Задание.....	1
3	Выполнение лабораторной работы	4
3.1	Настройка GitHub.....	4
3.2	Базовая настройка Git.....	4
3.3	Создание SSH-ключа	5
3.4	Создание рабочего пространства и репозитория курса на основе шаблона.....	6
3.5	Создание репозитория курса на основе шаблона	6
3.6	Настройка каталога курса	7
3.7	Выполнение заданий для самостоятельной работы	8
4	Выводы.....	10

1 Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git.

2 Задание

1. Настройка GitHub.
2. Базовая настройка Git.
3. Создание SSH-ключа.
4. Создание рабочего пространства и репозитория курса на основе шаблона.
5. Создание репозитория курса на основе шаблона.
6. Настройка каталога курса.
7. Выполнение заданий для самостоятельной работы. # Теоретическое введение

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. 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. 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. 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. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Система контроля версий Git представляет собой набор программ командной строки. Доступ к ним можно получить из терминала посредством ввода команды git с различными опциями. Благодаря тому, что Git является распределённой системой контроля версий, резервную копию локального хранилища можно сделать простым копированием или архивацией.

Основные команды git{#tbl:std-dir}

Команда	Описание
git init	создание основного дерева репозитория
git pull	получение обновлений (изменений) текущего дерева из центрального репозитория
git push	отправка всех произведённых изменений локального дерева в центральный репозиторий
git status	просмотр списка изменённых файлов в текущей директории

Команда	Описание
<code>git diff</code>	просмотр текущих изменений
<code>git add</code>	добавить все изменённые и/или созданные файлы и/или каталоги
<code>git add имена_файлов</code>	добавить конкретные изменённые и/или созданные файлы и/или каталоги
<code>git rm имена_файлов</code>	удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог остаётся в локальной директории)
<code>git commit -am 'Описание коммита'</code>	сохранить все добавленные изменения и все изменённые файлы
<code>git checkout -b имя_ветки</code>	создание новой ветки, базирующейся на текущей
<code>git checkout имя_ветки</code>	переключение на некоторую ветку (при переключении на ветку, которой ещё нет в локальном репозитории, она будет создана и связана с удалённой)
<code>git push origin имя_ветки</code>	отправка изменений конкретной ветки в центральный репозиторий
<code>git merge --no-ff имя_ветки</code>	слияние ветки с текущим деревом
<code>git branch -d имя_ветки</code>	удаление локальной уже слитой с основным деревом ветки

Команда	Описание
<code>git branch -D имя_ветки</code>	принудительное удаление локальной ветки
<code>git push origin :имя_ветки</code>	удаление ветки с центрального репозитория

3 Выполнение лабораторной работы

3.1 Настройка GitHub

Создала учетную запись на сайте <https://github.com/> (рис. 1).

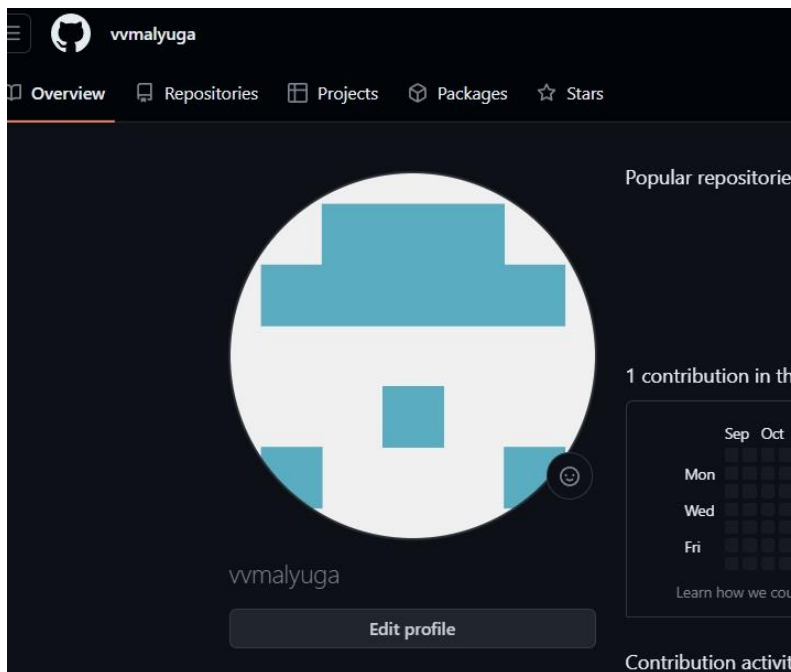


Рис. 1: Профиль в github

3.2 Базовая настройка Git

Открыла терминал и ввела следующие команды, указав мои имя и email. Настроила utf-8 в выводе сообщений git. Задала имя начальной ветки (master) и параметры autocrlf, safecrlf (рис. 2).

```

error: key does not contain a section: vvmalyuga
[vvmalyuga@vvmalyuga ~]$ git config --global user.name "vvmalyuga"
[vvmalyuga@vvmalyuga ~]$ git config --global user.email "1132236050@pfur.ru"
[vvmalyuga@vvmalyuga ~]$ git config --global core.quotepath false
[vvmalyuga@vvmalyuga ~]$ git config --global init.defaultBranch master
[vvmalyuga@vvmalyuga ~]$ git config --global core.autocrlf input
[vvmalyuga@vvmalyuga ~]$ git config --global core.safecrlf warn
[vvmalyuga@vvmalyuga ~]$

```

Рис. 2: Настройка git

3.3 Создание SSH-ключа

Сгенерировала пару ключей (приватный и открытый) (рис. 3). Просмотрела ключ с помощью команды cat, скопировала его, вставила в поле для ключа на сайте (рис. 4). Авторизовала ключ (рис. 5).

```

[vvmalyuga@vvmalyuga ~]$ ssh-keygen -C "vvmalyuga 1132236050@pfur.ru"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vvmalyuga/.ssh/id_rsa):
Created directory '/home/vvmalyuga/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vvmalyuga/.ssh/id_rsa
Your public key has been saved in /home/vvmalyuga/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RApCSEnrqIxp6gWc+qEXzCyy1THa6I1FNBCELuLFr3g vvmalyuga 1132236050@pfur.ru
The key's randomart image is:
+---[RSA 3072]-----+
|oBBo. . |
|o..o. o |
|... .. |
|*..O+ . |
|=O.+..o S |
|BoX +. |
|*Bo+. |
|++=E. |
|+oo |
+---[SHA256]-----+

```

Рис. 3: Создание SSH ключа

```

[vvmalyuga@vvmalyuga arch-pc]$ cat /home/vvmalyuga/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCyXrAH9uSfqxE0e7Sr6aa2bE9A2z1hApSs/Z0o9S2rY/PXj8W4/eXSeQjp/
qW/rhw7kK+JBY0Drs25mspXfuSrZXWb1KZ3EmtgHGfADU0ZFxcTR6HEWY939n//v06VKhvShjwTB0vfcJfLm0j3vU/2wIvE
zKD3Pe917TiM1093W2yNXM18Htq+BD/jEYzFYHnVhsqXQZ3T1HVLa1iQXcxbKXt5AwH5HmMyFMfzH4q+ovT573mtX2cng5WP
cnZqxIo/ZKPoPsoDrzj1aNoerGG46WGQzVK64pD2egDPi6D0ajWw5j2tcENTNjcsX9MFLVIOpr0+QUxkz4/wm+/T3SY6/CNT0
V31G1025MALWqn4LnY0hYZL94oxnpKW1YoQVAudTBnwTviU/CzPLMMLDkgr4CJ29C7uEvNwx7FKyMbHqGLE53ymeVWoo0Wb7m
1YktQP27k93AcIJA48Ku6HkflSv0Bqgo3r2zIHBhwPQsHtFG/WCo4rA3GShkXm8xBM= vvmalyuga 1132236050@pfur.ru
[vvmalyuga@vvmalyuga arch-pc]$

```

Рис. 4: Открытый ключ SSH

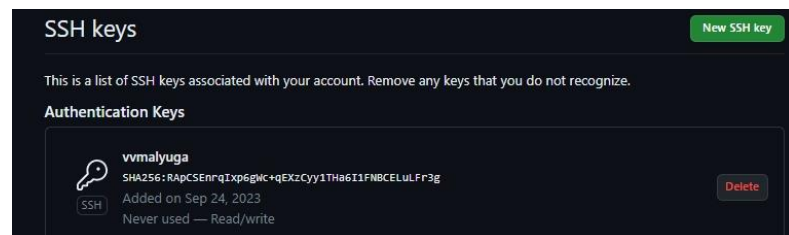


Рис. 5: Лист авторизованных ключей

3.4 Создание рабочего пространства и репозитория курса на основе шаблона

Открыла терминал и создала каталог для предмета «Архитектура компьютера». Проверила с помощью команды ls (рис. 6).

```
[vvmalyuga@vvmalyuga ~]$ mkdir -p ~/work/study/2023-2024/"Архитектура компьютера"
[vvmalyuga@vvmalyuga ~]$ ls ~/work/study/2023-2024
'Архитектура компьютера'
```

Рис. 6: Создание каталога «Архитектура компьютера» и проверка

3.5 Создание репозитория курса на основе шаблона

Перехожу на страницу репозитория с шаблоном курса <https://github.com/yamadharma/course-directory-student-template>. Далее выбрала Use this template (рис. 7). В открывшемся окне задала имя репозитория study_2023-2024_arhpc и создала репозиторий (рис. 8).

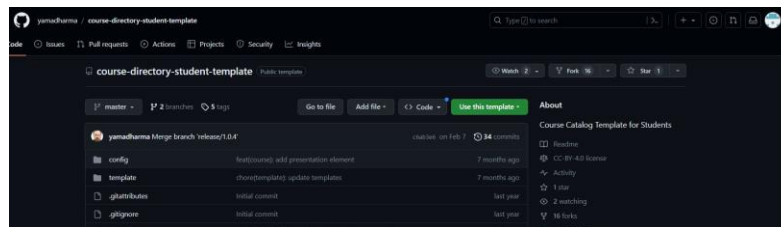


Рис. 7: Страница репозитория с шаблоном курса

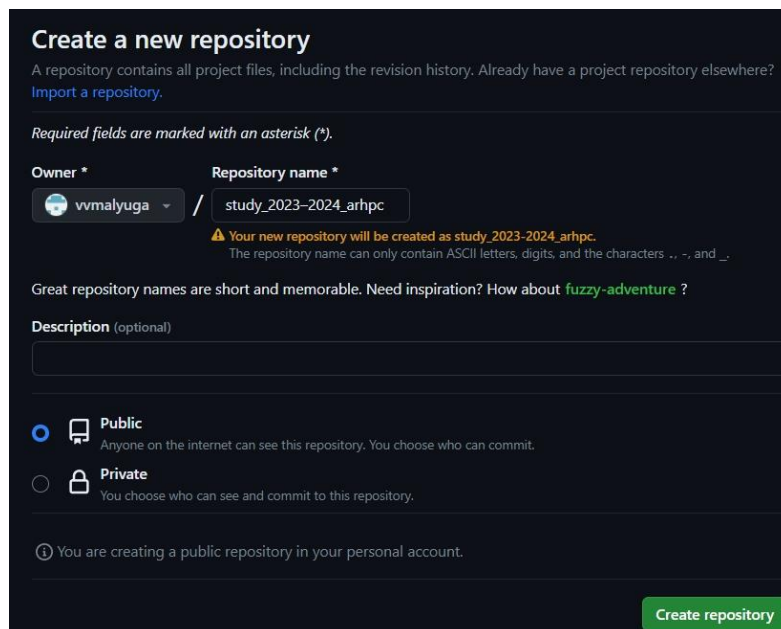


Рис. 8: Страница создания репозитория study_2023-2024_arhpc

Открыла терминал и перешла в каталог курса. Клонировала созданный репозиторий, скопировав ссылку для клонирования на странице созданного репозитория Code -> SSH (рис. 9).

```
[vvmalyuga@vvmalyuga ~]$ cd ~/work/study/2023-2024/"Архитектура компьютера"
[vvmalyuga@vvmalyuga Архитектура компьютера]$ git clone --recursive git@github.com:vvmalyuga/study_2023-2024_arhpc.git arch-pc
Cloning into 'arch-pc'...
The authenticity of host 'github.com (140.82.121.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvC0qU.
```

Рис. 9: Клонирование созданного репозитория

3.6 Настройка каталога курса

Перешла в каталог курса, удалила лишние файлы с помощью команды `rm` (рис. 10).

```
[vvmalyuga@vvmalyuga Архитектура компьютера]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[vvmalyuga@vvmalyuga arch-pc]$ rm package.json
```

Рис. 10: Удаление лишних файлов

Создала необходимые каталоги с помощью `git add`, комментирую и сохраняю изменения на сервере как добавление курса с помощью `git commit` (рис. 11). Отправила файлы на сервер с помощью `push` (рис. 12). Проверила правильность создания иерархии рабочего пространства в локальном репозитории и на странице github (рис. 13).

```
[vvmalyuga@vvmalyuga arch-pc]$ echo arch-pc > COURSE
[vvmalyuga@vvmalyuga arch-pc]$ make
[vvmalyuga@vvmalyuga arch-pc]$ git add .
[vvmalyuga@vvmalyuga arch-pc]$ git commit -am 'feat(main): make course structure'
[master c6435d9] feat(main): make course structure
199 files changed, 54725 insertions(+), 14 deletions(-)
create mode 100644 labs/README.md
create mode 100644 labs/README.ru.md
create mode 100644 labs/lab01/presentation/Makefile
create mode 100644 labs/lab01/presentation/image/kulyabov.jpg
create mode 100644 labs/lab01/presentation/presentation.md
```

Рис. 11: Создание необходимых каталогов

```
create mode 100644 presentation/report/pandoc/filters/pandocxnos/pandocattributes.py
create mode 100644 presentation/report/report.md
[vvmalyuga@vvmalyuga arch-pc]$ git push
Enumerating objects: 37, done.
Counting objects: 100% (37/37), done.
Delta compression using up to 2 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (35/35), 342.13 KiB | 2.67 MiB/s, done.
Total 35 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:vvmalyuga/study_2023-2024_arhpc.git
b2b7a0e..c6435d9 master -> master
[vvmalyuga@vvmalyuga arch-pc]$
```

Рис. 12: Отправка файлов на сервер

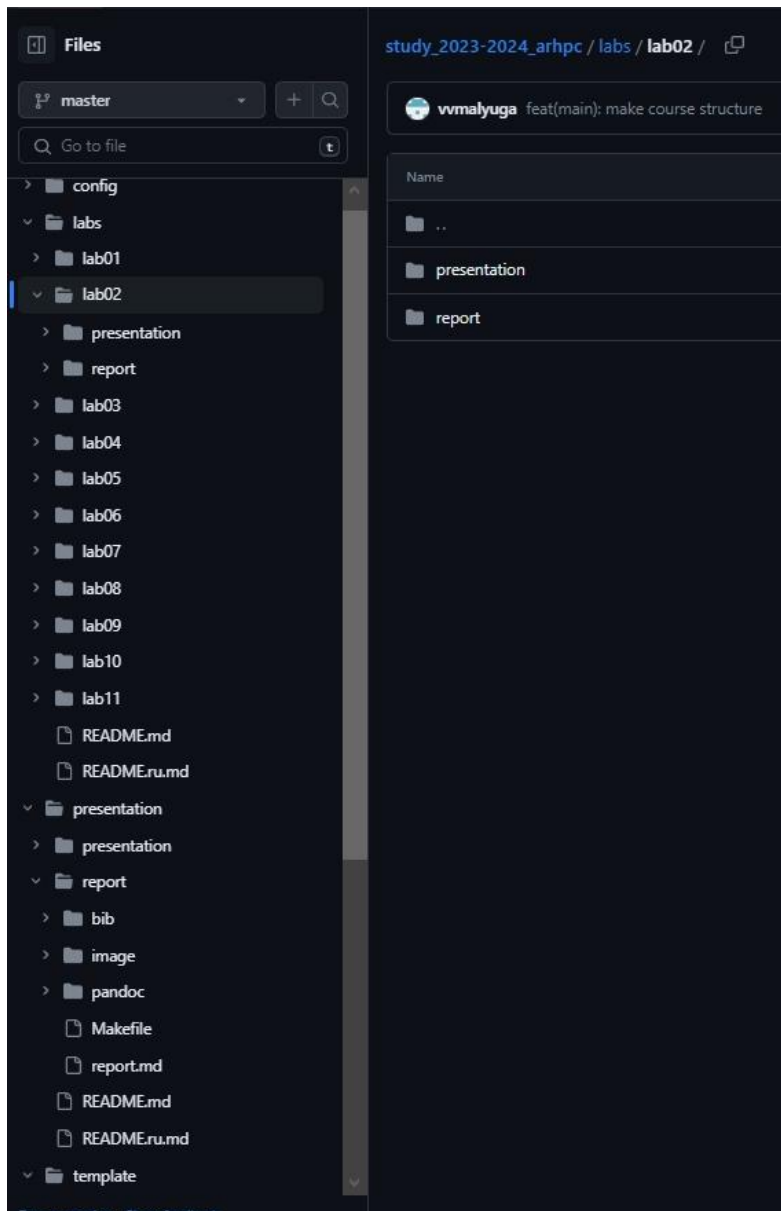


Рис. 13: Страница репозитория

3.7 Выполнение заданий для самостоятельной работы

1. Перехожу в директорию labs/lab02/report с помощью утилиты cd. Создаю в каталоге файл для отчета по второй лабораторной работе с помощью утилиты touch (рис. 14).

```
[vmmalyuga@vmmalyuga arch-pc]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/
lab02/report
[vmmalyuga@vmmalyuga report]$ touch Л02_Маюга_отчет.doc
[vmmalyuga@vmmalyuga report]$ nautilus ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/
labs/lab02/report/Л02_Маюга_отчет.doc
** Message: ...: Connecting to org.freedesktop.Tracker3.Miner.Files
```

Рис. 14: Создание текстового документа для оформления отчета

Оформляю отчет по лабораторной работе №2 в текстовом процессоре LibreOffice Writer.

- Первая лабораторная работа у меня находится в подкаталоге домашней директории Downloads. Перехожу в каталог [...] /arch-pc/labs/lab01/report и копирую сюда файл Л01_Малюга_отчет.pdf с помощью команды cp и проверяю выполнение с помощью ls (рис. 15).

```
[vvmalyuga@vvmalyuga report]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[vvmalyuga@vvmalyuga report]$ ls
bib image Makefile pandoc report.md
[vvmalyuga@vvmalyuga report]$ cp ~/Downloads/Л01_Малюга_отчет.pdf /home/vvmalyuga/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab01/report
[vvmalyuga@vvmalyuga report]$ ls
bib image Makefile pandoc report.md Л01_Малюга_отчет.pdf
[vvmalyuga@vvmalyuga report]$
```

Рис. 15: Копирование отчета по лабораторной работе №1 в каталог

- Перешла в каталог arch-pc, чтобы отправить все добавленные файлы в репозиторий. Использую команды git add (добавляю все измененные файлы), и затем git commit -m «Add existing files» (сохраняю все измененные файлы), и в конце git push, чтобы отправить все изменения на сервер (рис. 16).

```
[vvmalyuga@vvmalyuga report]$ cd ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc
[vvmalyuga@vvmalyuga arch-pc]$ git add .
[vvmalyuga@vvmalyuga arch-pc]$ git commit -m "Add existing files"
[master ff6112d] Add existing files
3 files changed, 1 insertion(+)
create mode 100644 labs/lab01/report/Л01_Малюга_отчет.pdf
create mode 100644 labs/lab02/report/.~lock.Л02_Малюга_отчет.doc#
create mode 100644 labs/lab02/report/Л02_Малюга_отчет.doc
[vvmalyuga@vvmalyuga arch-pc]$ git push
Enumerating objects: 16, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 2 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.62 MiB | 1.94 MiB/s, done.
Total 10 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To github.com:vvmalyuga/study_2023-2024_arhpc.git
 c6435d9..ff6112d master -> master
[vvmalyuga@vvmalyuga arch-pc]$
```

Рис. 16: Добавление и отправка файлов в центральный репозиторий

Проверила выполнение работы на сайте github. Действительно, все добавилось (рис. 17).

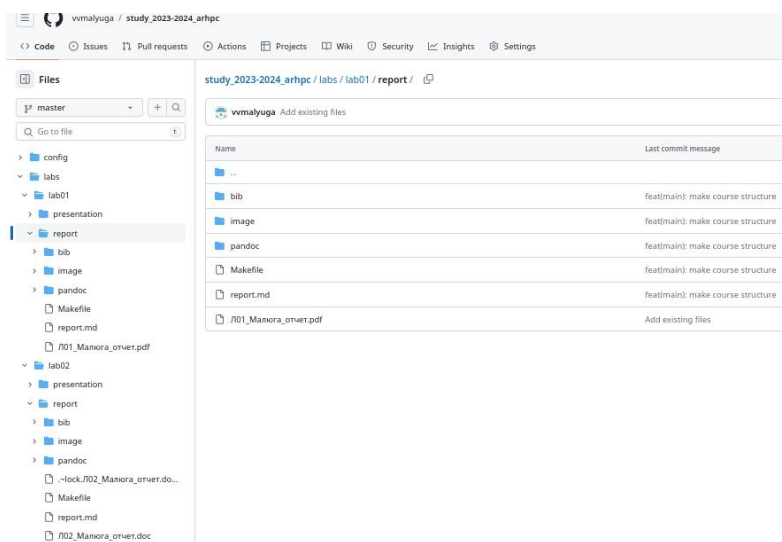


Рис. 17: Страница с добавленными файлами в подкаталогах lab01 и lab02

4 Выводы

Изучила идеологию и средства контроля версий, также приобрела практические навыки по работе с системой git.