

# **Очѐт по лабораторной работе**

**Лабораторная работа №2**

Дикач Анна Олеговна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Создание репозитория курса на основе шаблона . . . . .	6
2.2	Настройка каталога курса . . . . .	7
2.3	Контрольные вопросы . . . . .	9
<b>3</b>	<b>Выводы</b>	<b>11</b>

## Список иллюстраций

2.1	созданный аккаунт в Github . . . . .	6
2.2	создание репозитория курса . . . . .	6
2.3	создание репозитория курса . . . . .	7
2.4	команда для перехода в каталог курса . . . . .	7
2.5	команда для удаления лишних файлов . . . . .	7
2.6	команда для создания каталогов . . . . .	7
2.7	команда для отправки файлов на сервер . . . . .	8
2.8	команда для отправки файлов на сервер . . . . .	8
2.9	скринкаст проведённой работы . . . . .	8

## Список таблиц

# 1 Цель работы

Изучить идеологию и применение средств контроля версий, а также освоить умения по работе с git.

## 2 Выполнение лабораторной работы

Так как аккаунт в Github был создан и настроен в прошлом семестре, приступаю к созданию репозитория курса на основе шаблона (рис. 2.1)

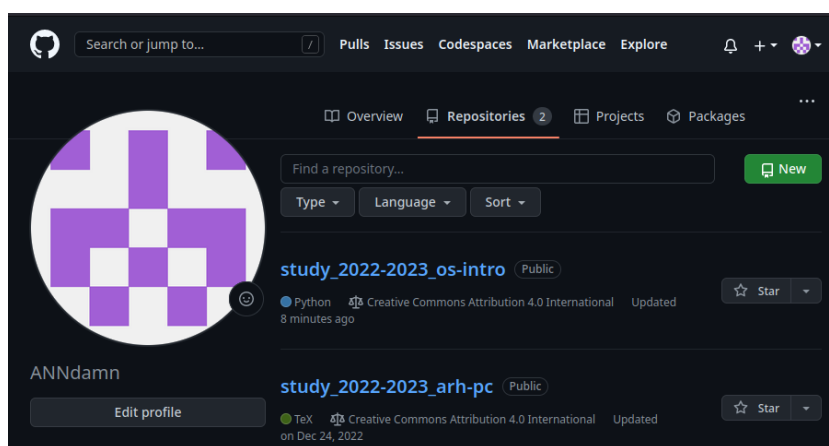


Рис. 2.1: созданный аккаунт в Github

### 2.1 Создание репозитория курса на основе шаблона

1. Создаю шаблон рабочего пространства (рис. 2.2) (рис. 2.3)

```
[aodikach@fedora ~]$ mkdir -p ~/work/study/2022-2023/"Операционные системы"  
[aodikach@fedora ~]$ cd ~/work/study/2022-2023/"Операционные системы"
```



Рис. 2.2: создание репозитория курса

```
[aodikach@fedora Операционные системы]$ gh repo create study_2022-2023_os-intro --template=yamadharma/course-directory-student-template --public
/ Created repository ANMDam/study_2022-2023_os-intro on GitHub
[aodikach@fedora Операционные системы]$ git clone --recursive git@github.com:ANMDam/study_2022-2023_os-intro.git os-intro
Клонирование в «os-intro»...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 27 (delta 1), reused 11 (delta 0), pack-reused 0
Получение объектов: 100% (27/27), 16.93 KiB | 361.00 KiB/c, 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/aodikach/work/study/2022-2023/Операционные системы/os-intro/template/presentation»...
remote: Enumerating objects: 82, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 82 (delta 28), reused 77 (delta 23), pack-reused 0
Получение объектов: 100% (82/82), 92.99 KiB | 914.00 KiB/c, готово.
Определение изменений: 100% (24/24), готово.
Клонирование в «/home/aodikach/work/study/2022-2023/Операционные системы/os-intro/template/report»...
remote: Enumerating objects: 101, done.
remote: Counting objects: 100% (101/101), done.
remote: Compressing objects: 100% (78/78), done.
remote: Total 101 (delta 49), reused 88 (delta 27), pack-reused 0
Получение объектов: 100% (101/101), 327.25 KiB | 1.64 MiB/c, готово.
Определение изменений: 100% (40/40), готово.
Submodule path 'template/presentation': checked out 'b1be3800ee91f5009264cb755d116174540b753e'
Submodule path 'template/report': checked out '1d1b61dcac9c287a83917b82e3aeef11a33b1e3b2'
```

Рис. 2.3: создание репозитория курса

## 2.2 Настройка каталога курса

### 2. Перехожу в каталог курса (рис. 2.4)

```
[aodikach@fedora Операционные системы]$ cd ~/work/study/2022-2023/"Операционные системы"/os-intro
```

Рис. 2.4: команда для перехода в каталог курса

### 3. Удаляю лишние файлы (рис. 2.5)

```
aodikach@fedora os-intro]$ rm package.json
```

Рис. 2.5: команда для удаления лишних файлов

### 4. Создаю необходимые каталоги (рис. 2.6)

```
[aodikach@fedora os-intro]$ echo os-intro > COURSE
[aodikach@fedora os-intro]$ make
```

Рис. 2.6: команда для создания каталогов

### 5. Отправляю файлы на сервер (рис. 2.7)

```

[oodikach@fedora os-intro]$ git add .
[oodikach@fedora os-intro]$ git commit -am 'feat(main): make course structure'
[master b180fc1] feat(main): make course structure
361 files changed, 10927 insertions(+), 14 deletions(-)
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
create mode 100644 labs/lab01/report/Makefile
create mode 100644 labs/lab01/report/bib/cite.bib
create mode 100644 labs/lab01/report/image/placement_800_600_tech.jpg
create mode 100644 labs/lab01/report/pandoc/csl/gost-r-7-8-5-2008-numeric.csl
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab01/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/_init_.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab01/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab01/report/pandocxnos/pandocattributes.py
create mode 100644 labs/lab01/report/report.md
create mode 100644 labs/lab02/presentation/Makefile
create mode 100644 labs/lab02/presentation/image/kulyabov.jpg
create mode 100644 labs/lab02/presentation/presentation.md
create mode 100644 labs/lab02/report/Makefile
create mode 100644 labs/lab02/report/bib/cite.bib
create mode 100644 labs/lab02/report/image/placement_800_600_tech.jpg
create mode 100644 labs/lab02/report/pandoc/csl/gost-r-7-8-5-2008-numeric.csl
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_eqnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_fignos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_secnos.py
create mode 100755 labs/lab02/report/pandoc/filters/pandoc_tablenos.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/_init_.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/core.py
create mode 100644 labs/lab02/report/pandoc/filters/pandocxnos/main.py
create mode 100644 labs/lab02/report/pandocxnos/pandocattributes.py
create mode 100644 labs/lab02/report/report.md
create mode 100644 labs/lab03/presentation/Makefile
create mode 100644 labs/lab03/presentation/image/kulyabov.jpg
create mode 100644 labs/lab03/presentation/presentation.md
create mode 100644 labs/lab03/report/Makefile
create mode 100644 labs/lab03/report/bib/cite.bib

```

Рис. 2.7: команда для отправки файлов на сервер

```

[oodikach@fedora os-intro]$ git push
Enumerating objects: 40, done.
Подсчет объектов: 100% (40/40), готово.
Сканирование объектов: 100% (39/39), готово.
Запись объектов: 100% (38/38), 342,39 Кб | 2,76 Мб/с, готово.
Всего 38 (изменений 4), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To github.com:ANNdamn/study_2022-2023_os-intro.git
 38e19f9..b180fc1 master -> master
[oodikach@fedora os-intro]$

```

Рис. 2.8: команда для отправки файлов на сервер

## 6. Итог проведённой работы (рис. 2.9)

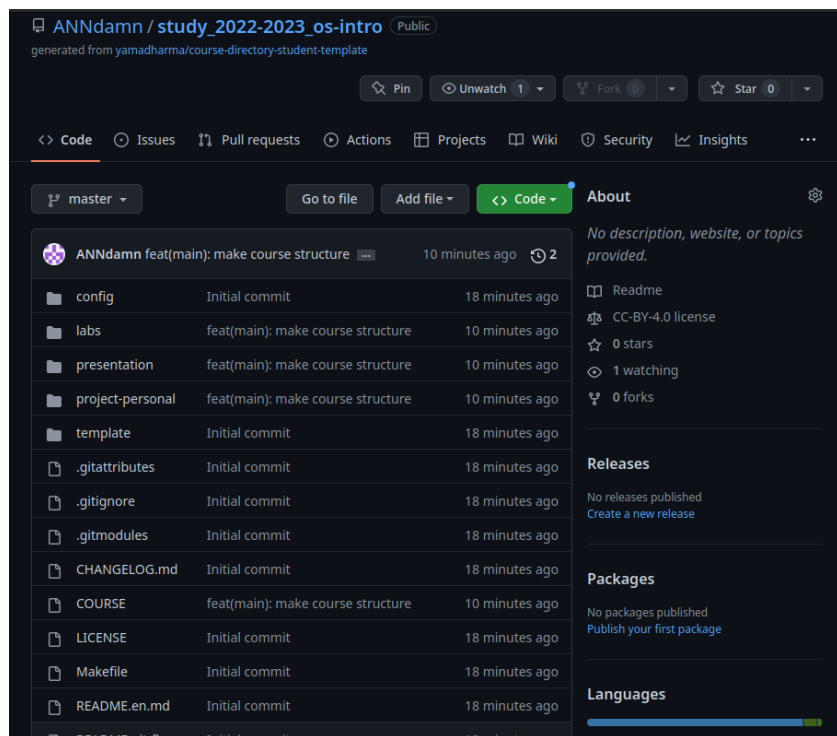


Рис. 2.9: скринкаст проведённой работы



## 2.3 Контрольные вопросы

1. Системы контроля версий (VCS) применяются при работе нескольких человек над одним проектом.

Система управления версиями позволяет хранить версии одного и того же документа, возвращаться к ним и следить за изменениями и фиксировать их.

2. хранилище - место хранения файлов с которым могут взаимодействовать все участники  
commit - делает для проекта снимок текущего состояния изменений, добавленных в раздел проиндексированных файлов. история - показывает операции, которые были произведены с объектом. имеет большое количество опций для поиска коммитов по разным критериям рабочая копия - является снимком одной версии проекта. файлы извлекаются из сжатой базы данных в каталоге Git и помещаются на диск для того чтобы их могли редактировать другие пользователи

3. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществляется через специальное клиентское приложение. Примеры: CVS, Subversion  
Децентрализованные 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. Примеры: Git и Mercurial

4. создания и работа с файлами, выгрузка их в хранилище. не требует создания рабочих копий. можно сделать проект конфедициальным, не доступным для других пользователей git.

5. создание репозитория, обновить проект, скопировать последнюю версию, внести изменения, разрешить конфликты, создать ветку.
6. основными задачами являются: проверка ветки, переключение на новую, написание кода, просмотр статуса, создание коммита, отправка
7. `git add` - добавляет содержимое рабочего каталога в индекс `git status` - показывает состояние файлов `git diff` - используется для вычисления разницы между `git diff` `git diff` - запускает внешнюю утилиту сравнения для показа различий двух деревьев `git commit` - берёт все данные, добавленные в индекс с помощью `git add`, и сохраняет их слепок во внутренней базе данных, а потом сдвигает указатель текущей ветки на этот слепок `git reset` - используется для отмены изменений `git rm` - используется в Git для удаления файлов из индекса и рабочей копии `git mv` - перемещает файлы, добавляет содержимое рабочего каталога в индекс нового файла и удаляет старый `git clean` - удаляет мусор из рабочего каталога
8. с помощью `git add` пользователь изменяет файл, проверяет состояние других файлов с помощью `git status`, с помощью `git diff` сверяет сделанную работу с работой своего партнёра, с помощью `git rm` удаляет рабочую копию своего партнёра.
9. ветви нужны для того чтобы программисты могли вести совместную работу и при этом не мешать друг другу своими действиями
10. некоторые файлы игнорируются для их исключения из истории git.

## 3 Выводы

Создала аккаунт в Github, изучила его идеологию и применение средств контроля версий.