

Отчет по лабораторной работе №1

**Настройка рабочего пространства. Система контроля версий Git. Язык
разметки Markdown.**

Желдакова Виктория Алексеевна

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
4 Выполнение лабораторной работы	8
5 Выводы	15
Список литературы	16

Список иллюстраций

4.1	Иерархия рабочего пространства	8
4.2	Копирование репозитория	8
4.3	Установка Chocolatey	9
4.4	Установка make и запуск	9
4.5	Установка Git	10
4.6	Генерация ключа	10
4.7	Настройка SSH-агента	10
4.8	Добавление нового ключа	11
4.9	Клоинрование репозитория	11
4.10	Установка pandoc	12
4.11	Установка pandoc-crossref	12
4.12	Конвертация отчета	12
4.13	Отчет в формате docx	13
4.14	Отчет в формате pdf	13
4.15	Презентация в формате pdf	14
4.16	Презентация в формате html	14

Список таблиц

1 Цель работы

Настройка рабочего пространства для лабораторных работ. Практическое использование системы контроля версий Git и языка разметки Markdown.

2 Задание

Создать и настроить репозиторий курса на основе шаблона. Установить связь между компьютером и Github с помощью SSH-ключа. Сконвертировать отчет из расширения .md в .docx и .pdf. Сконвертировать презентацию из расширения .md в .html и .pdf.

3 Теоретическое введение

Git — абсолютный лидер по популярности среди современных систем управления версиями. Это развитый проект с активной поддержкой и открытым исходным кодом. Система Git была изначально разработана в 2005 году Линусом Торвальдсом — создателем ядра операционной системы Linux. Git применяется для управления версиями в рамках колоссального количества проектов по разработке ПО, как коммерческих, так и с открытым исходным кодом.

Markdown — облегчённый язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций.

Команда Описание команды

init	Создает репозиторий в текущем каталоге
status	Показывает состояния файлов в рабочем каталоге и индексе
commit	Берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.
add	Добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита
push	Используется для установления связи с удалённым репозиторием, вычисления локальных изменений отсутствующих в нём, и собственно их передачи в вышеупомянутый репозиторий

4 Выполнение лабораторной работы

Создаём каталог по приведённому шаблону (рис. 4.1).



Рис. 4.1: Иерархия рабочего пространства

Копируем шаблон репозитория (рис. 4.2).

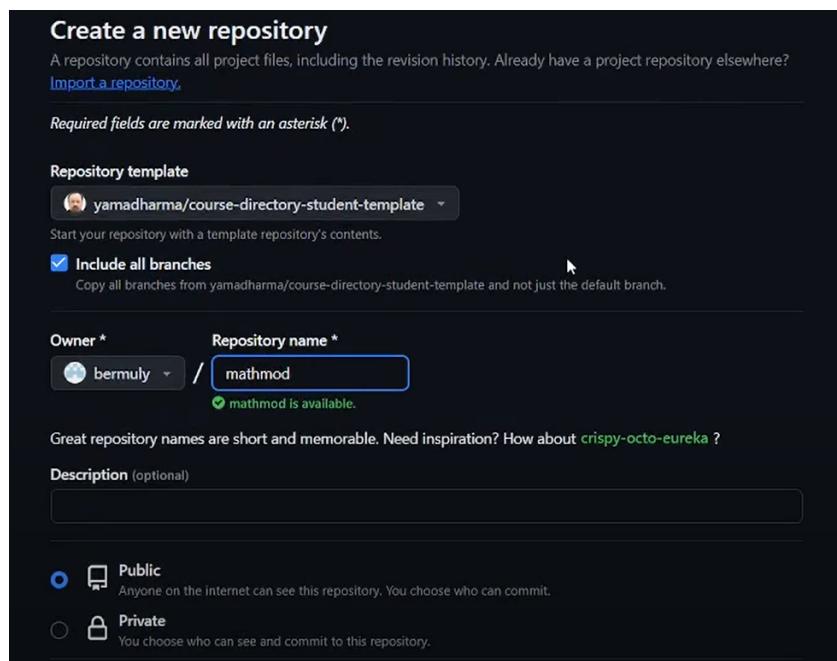
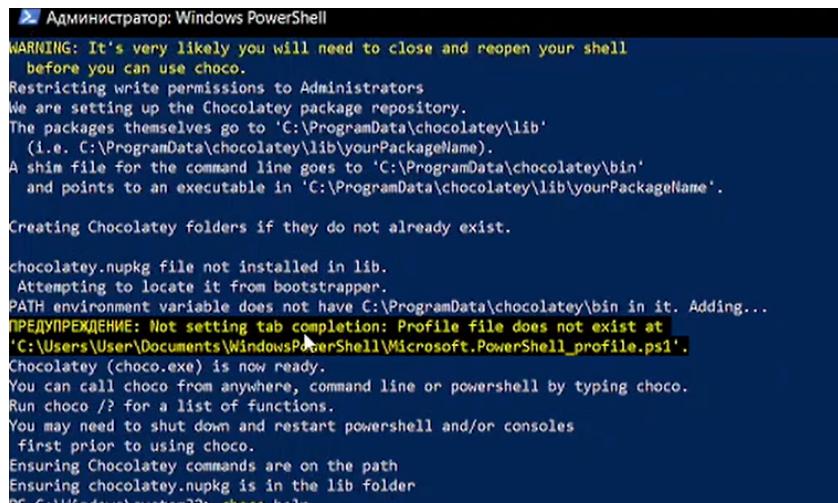


Рис. 4.2: Копирование репозитория

Устанавливаем менеджер пакетов Chocolatey (рис. 4.3).



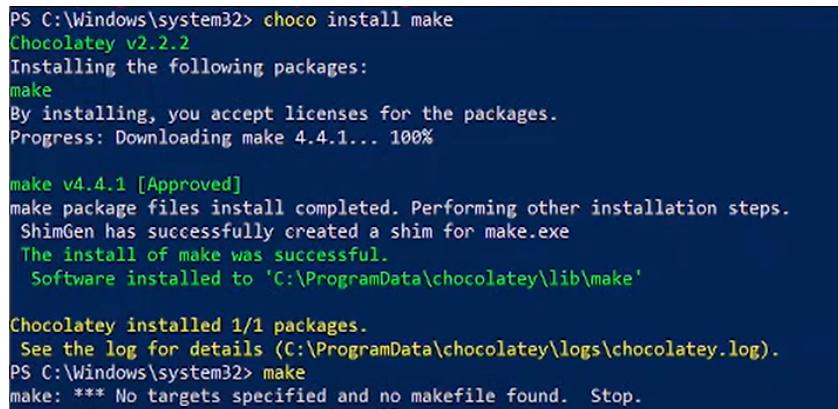
```
Administrator: Windows PowerShell
WARNING: It's very likely you will need to close and reopen your shell
         before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
  (i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
  and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
ПРЕДУПРЕЖДЕНИЕ: Not setting tab completion: Profile file does not exist at
'C:\Users\User\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
PS C:\Windows\system32>
```

Рис. 4.3: Установка Chocolatey

Устанавливаем make и проверяем работоспособность (рис. 4.4).



```
PS C:\Windows\system32> choco install make
Chocolatey v2.2.2
Installing the following packages:
make
By installing, you accept licenses for the packages.
Progress: Downloading make 4.4.1... 100%

make v4.4.1 [Approved]
make package files install completed. Performing other installation steps.
ShimGen has successfully created a shim for make.exe
The install of make was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\make'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32> make
make: *** No targets specified and no makefile found. Stop.
```

Рис. 4.4: Установка make и запуск

Устанавливаем Git (рис. 4.5).

```

Software installed to 'C:\ProgramData\chocolatey\extensions\chocolatey-compatibility'
Progress: Downloading chocolatey-core.extension 1.4.0... 100%
chocolatey-core.extension v1.4.0 [Approved]
chocolatey-core.extension package files install completed. Performing other installation steps.
  Installed/updated chocolatey-core extensions.
    The install of chocolatey-core.extension was successful.
      Software installed to 'C:\ProgramData\chocolatey\extensions\chocolatey-core'
Progress: Downloading git.install 2.43.0... 100%
git.install v2.43.0 [Approved]
git.install package files install completed. Performing other installation steps.
The package git.install wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): Y

Using Git LFS
Installing 64-bit git.install...
git.install has been installed.
git.install installed to 'C:\Program Files\Git'
  git.install can be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
  see the changes (or in powershell/cmd.exe just type "refreshenv").
  The install of git.install was successful.
  Software installed to 'C:\Program Files\Git'
Progress: Downloading git 2.43.0... 100%
git v2.43.0 [Approved]
git package files install completed. Performing other installation steps.
  The install of git was successful.
  Software installed to 'C:\ProgramData\chocolatey\lib\git'

Chocolatey installed 4/4 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

```

Рис. 4.5: Установка Git

Генерируем SSH-ключ (рис. 4.6).

```

PS C:\Users\User\Documents\work\study\2023-2024\Математическое моделирование\
mathmod> ssh-keygen -t ed25519 -C v.zhel3@yandex.ru
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\User/.ssh/id_ed25519):
Created directory 'C:\Users\User/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\User/.ssh/id_ed25519.
Your public key has been saved in C:\Users\User/.ssh/id_ed25519.pub.
The key fingerprint is:
SHA256:Of3gMxsNrL+fMOVdmwY0+LyFXYdRct1JFrVFFtcGzNW v.zhel3@yandex.ru
The key's randomart image is:
+--[ED25519 256]--+
|   o o.*oo=o|
| o.+ = ++E+|
| .O o o . +*|
| .* @ . . . *|
| .B S . . |
| .B o . |
| +.* . |
| .+ o . |
| .o . |
+---[SHA256]----+

```

Рис. 4.6: Генерация ключа

Настройка агента и добавление ключа (рис. 4.7).

```

PS C:\Windows\system32> Get-Service ssh-agent | Set-Service -StartupType Automatic -PassThru | Start-Service
PS C:\Windows\system32> ssh-add c:/Users/User/.ssh/id_ed25519
Identity added: c:/Users/User/.ssh/id_ed25519 (v.zhel3@yandex.ru)

```

Рис. 4.7: Настройка SSH-агента

Добавляем SSH-ключ в профиль Github (рис. 4.8).

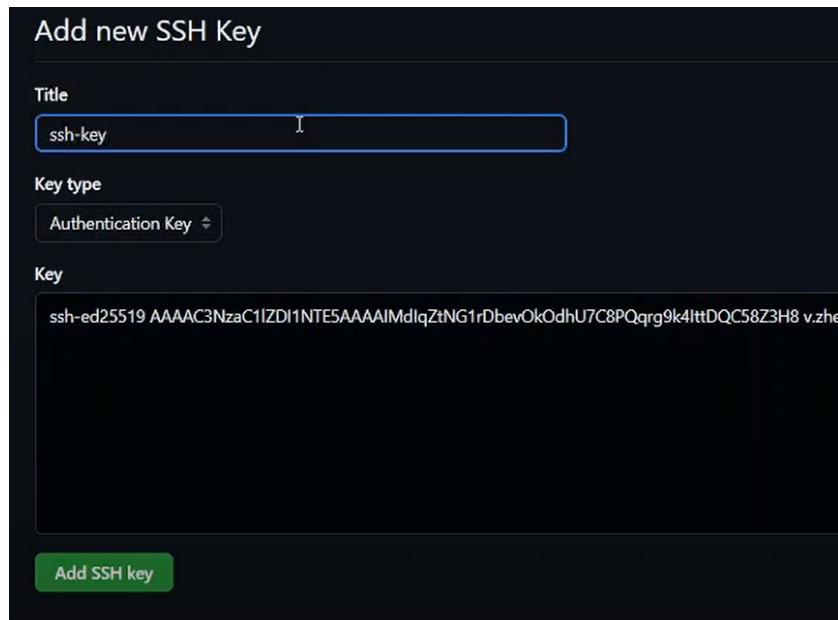


Рис. 4.8: Добавление нового ключа

Клонируем репозиторий на компьютер (рис. 4.9).

```
PS C:\Users\User\Documents\work\study\2023-2024\Математическое моделирование>
git clone --recursive git@github.com:bermuly/study_2023-2024_mathmod.git mathmod
Cloning into 'mathmod'...
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 16 (delta 0), pack-reused 0
S: 28% (9/32)
Receiving objects: 100% (32/32), 19.16 KiB | 1.47 MiB/s, done.
Resolving deltas: 100% (1/1), done.
Submodule 'template/presentation' (https://github.com/yamadharma/academic-presentation-markdown-template.git) registered for path 'template/presentation'
Submodule 'template/report' (https://github.com/yamadharma/academic-laboratory-report-template.git) registered for path 'template/report'
Cloning into 'C:/Users/User/Documents/work/study/2023-2024/Математическое моделирование/mathmod/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
Receiving objects: 100% (95/95), 96.99 KiB | 1003.00 KiB/s, done.
Resolving deltas: 100% (34/34), done.
cloning into 'C:/Users/User/Documents/work/study/2023-2024/Математическое моделирование/mathmod/template/report'...
```

Рис. 4.9: Клоинрование репозитория

Устанавливаем pandoc для конвертирования файлов (рис. 4.10).

```

PS C:\Windows\system32> choco install pandoc
Chocolatey v2.2.2
Installing the following packages:
pandoc
By installing, you accept licenses for the packages.
Progress: Downloading pandoc 3.1.11.1... 100%
pandoc v3.1.11.1 [Approved]
pandoc package files install completed. Performing other installation steps.
The package pandoc wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): Y

Installing 64-bit pandoc...
pandoc has been installed.
pandoc installed to 'C:\Users\User\AppData\Local\Pandoc'
Added C:\ProgramData\chocolatey\bin\pandoc.exe shim pointed to 'c:\users\user\appdata\local\pandoc\pandoc.exe'.
Added C:\ProgramData\chocolatey\bin\pandoc-citeproc.exe shim pointed to 'c:\users\user\appdata\local\pandoc\pandoc-citeproc.exe'.
    pandoc may be able to be automatically uninstalled.
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type 'refreshenv').
The install of pandoc was successful.
    Software installed to 'C:\Users\User\AppData\Local\Pandoc'

```

Рис. 4.10: Установка pandoc

Устанавливаем pandoc-crossref (рис. 4.11).

```

PS C:\Windows\system32> choco install pandoc-crossref
Chocolatey v2.2.2
Installing the following packages:
pandoc-crossref
By installing, you accept licenses for the packages.
Progress: Downloading pandoc-crossref 0.3.4.1... 100%
pandoc-crossref v0.3.4.1 [Approved]
pandoc-crossref package files install completed. Performing other installation steps.
The package pandoc-crossref wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): Y

Downloading pandoc-crossref
    from 'https://github.com/lierdakil/pandoc-crossref/releases/download/v0.3.4.1/windows-x86_64-pandoc_2.7.zip'
Progress: 100% - Completed download of C:\Users\User\AppData\Local\Temp\chocolatey\pandoc-crossref\0.3.4.1\windows-x86_64-pandoc_2.7.zip (17.18 MB).
Download of windows-x86_64-pandoc_2.7.zip (17.18 MB) completed.
Hashes match.
Extracting C:\Users\User\AppData\Local\Temp\chocolatey\pandoc-crossref\0.3.4.1\windows-x86_64-pandoc_2.7.zip to C:\ProgramData\chocolatey\lib\pandoc-crossref\tools...
C:\ProgramData\chocolatey\lib\pandoc-crossref\tools
ShimGen has successfully created a shim for pandoc-crossref.exe
The install of pandoc-crossref was successful.
    Software installed to 'C:\ProgramData\chocolatey\lib\pandoc-crossref\tools'

```

Рис. 4.11: Установка pandoc-crossref

Конвертируем отчет в docx и pdf(рис. 4.12).

```

PS C:\Users\User\Documents\work\study\2023-2024\Математическое моделирование\mathmod\labs\lab01\report> make
pandoc "report.md" --filter pandoc-crossref --number-sections --citeproc -o "report.docx"
pandoc "report.md" --filter pandoc-crossref --pdf-engine=xelatex --pdf-engine-opt=---shell-escape --citeproc --number-sections -o "report.pdf"
[WARNING] [makePDF] LaTeX Warning: Empty bibliography on input line 291.

```

Рис. 4.12: Конвертация отчета

Получаем отчет в формате docx (рис. 4.13).

Шаблон отчёта по лабораторной работе

Простейший вариант

Дмитрий Сергеевич Кулябов

Содержание

1 Цель работы	1
2 Задание.....	1
3 Теоретическое введение	1
4 Выполнение лабораторной работы.....	2
5 Выводы.....	2
Список литературы	3

1

1 Цель работы

Здесь приводится формулировка цели лабораторной работы. Формулировки цели для каждой лабораторной работы приведены в методических указаниях.

Рис. 4.13: Отчет в формате docx

Получаем отчет в формате pdf (рис. 4.14).

4 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 4.1).



Рис. 4.1: Название рисунка

Рис. 4.14: Отчет в формате pdf

Конвертируем презентацию и получаем её в формате pdf (рис. 4.15).

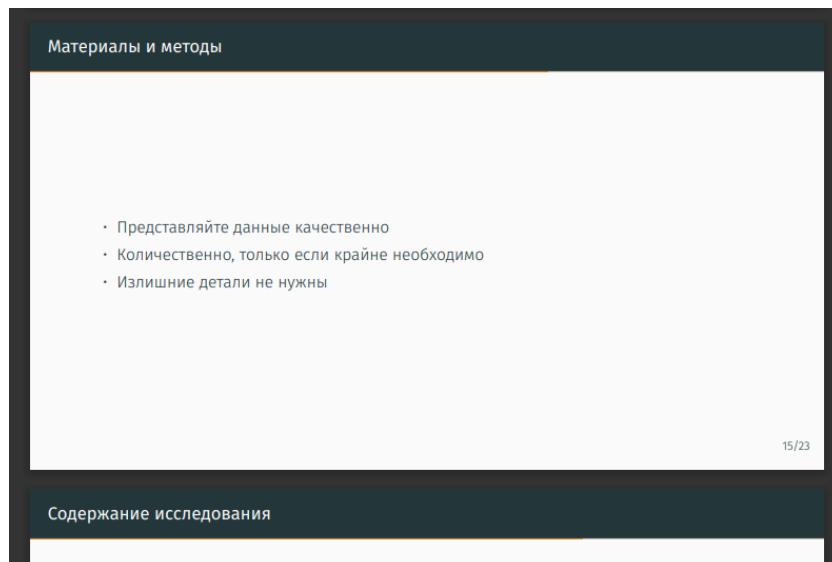


Рис. 4.15: Презентация в формате pdf

Получаем презентацию в формате html (рис. 4.16).

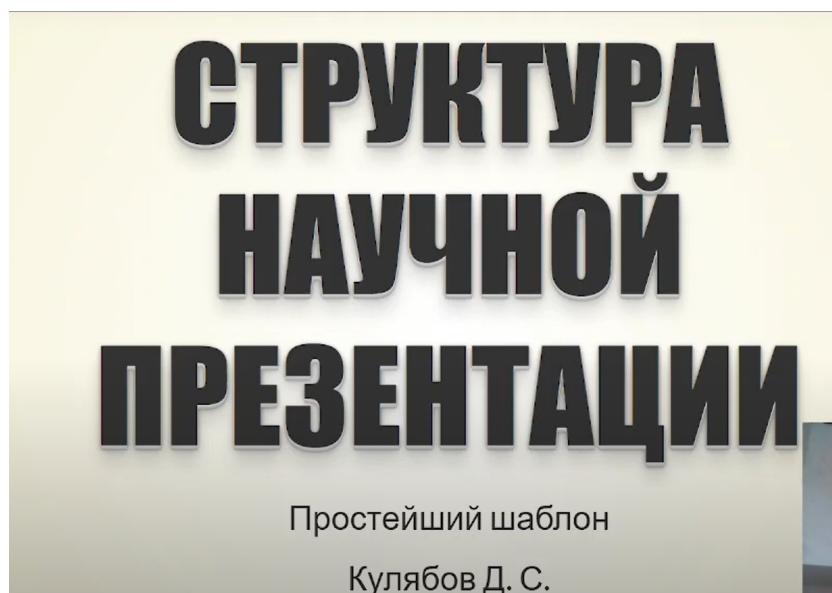


Рис. 4.16: Презентация в формате html

5 Выводы

Настроили рабочее пространство для лабораторных работ и попрактиковались с системой контроля версий Git и языком разметки Markdown.

Список литературы

- Документация по Git: <https://git-scm.com/docs/user-manual>
- Документация по Markdown: <https://www.markdownguide.org/>