

Отчёт по лабораторной работе №3

Уткина Алина Дмитриевна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Создание заголовков	6
3.2	Создание неупорядоченных и упорядоченных списков	6
3.3	Создание изображения	7
3.4	Окончание работы, создание дополнительных форматов отчета .	8
4	Выводы	10

Список иллюстраций

3.1	Создание заголовка в формате markdown	6
3.2	Результат создания заголовка в формате pdf	6
3.3	Создание списков в формате markdown	7
3.4	Результат создания списков в формате pdf	7
3.5	Создание изображений в формате markdown	8
3.6	Результат создания изображений в формате pdf	8
3.7	Создание дополнительных форматов отчета	9
3.8	Удаление дополнительных форматов отчета	9

1 Цель работы

Целью данной работы является приобретение навыков оформления отчётов с помощью легковесного языка разметки Markdown.

2 Задание

Сделать отчёт по лабораторной работе №2 в формате Markdown.

3 Выполнение лабораторной работы

3.1 Создание заголовков

Чтобы создать заголовок, используется знак “#” или несколько таких знаков, в зависимости от уровня заголовка. Так для основного пункта мы используем один знак, а для его подпункта - два знака (рис. 3.1), (рис. 3.2).

```
19  
80 # Выполнение лабораторной работы  
81  
82 ## Установка программного обеспечения
```

Рис. 3.1: Создание заголовка в формате markdown

3 Выполнение лабораторной работы

3.1 Установка программного обеспечения

Установка программного обеспечения

Рис. 3.2: Результат создания заголовка в формате pdf

3.2 Создание неупорядоченных и упорядоченных списков

Для создания неупорядоченного списка можно использовать звездочки или тире, для упорядоченного - соответствующие цифры (рис. 3.3), (рис. 3.4). Для многоуровневых списков можно использовать отступы.

```

156
157 1. Системы контроля версий (VCS) разработаны специально для того, чтобы
    максимально упростить и упорядочить работу над проектом (вне зависимости
    от того, сколько человек в этом участвуют). СКВ дает возможность видеть,
    кто, когда и какие изменения вносил; позволяет формировать новые ветви
    проекта, объединять уже имеющиеся; настраивать контроль доступа к
    проекту; осуществлять откат до предыдущих версий.
158
159 2. Основные понятия:
160
161 - Хранилище (repository, сокр. репо), или репозиторий, — место хранения
    всех версий и служебной информации;
162 - Коммит (commit) — 1) синоним версии; 2) создание новой версии («сделать
    коммит», «закоммитить»);
163 - История разработки — совокупность всех версий файлов, над которыми
    ведется работа. Историей разработки в данном случае будет список
    изменений: создание файла, добавление изначального текста, исправление
    опечатки, добавление нового текста, объединение двух версий файла (при
    выполнении слияния);
164 - Рабочая копия (working copy или working tree) — текущее состояние
    файлов проекта, основанное на версии из хранилища (обычно на последней).
165

```

Рис. 3.3: Создание списков в формате markdown

1. Системы контроля версий (VCS) разработаны специально для того, чтобы максимально упростить и упорядочить работу над проектом (вне зависимости от того, сколько человек в этом участвуют). СКВ дает возможность видеть, кто, когда и какие изменения вносил; позволяет формировать новые ветви проекта, объединять уже имеющиеся; настраивать контроль доступа к проекту; осуществлять откат до предыдущих версий.
2. Основные понятия:
 - Хранилище (repository, сокр. репо), или репозиторий, — место хранения всех версий и служебной информации;
 - Коммит (commit) — 1) синоним версии; 2) создание новой версии («сделать коммит», «закоммитить»);
 - История разработки — совокупность всех версий файлов, над которыми ведется работа. Историей разработки в данном случае будет список изменений: создание файла, добавление изначального текста, исправление

Рис. 3.4: Результат создания списков в формате pdf

3.3 Создание изображения

Чтобы вставить изображение в документ можно с помощью непосредственного указания адреса изображения (рис. 3.5), (рис. 3.6). В квадратных скобках указывается подпись к изображению; в круглых скобках указывается URL-адрес или

относительный путь изображения, в фигурных скобках указывается идентификатор изображения (`#fig:fig1`) для ссылки на него по тексту и размер изображения относительно ширины страницы (`width=70%`).

```
83
84 Установим git (рис. @fig:001):
85
86 ![Установка git](image/1.jpg){#fig:001 width=70%}
87
88 Установим gh (рис. @fig:002):
89
90 ![Установка gh](image/2.jpg){#fig:002 width=70%}
91
```

Рис. 3.5: Создание изображений в формате markdown

Установим `git` (рис. 3.1):

```
[adutkina@fedora ~]$ sudo -i
[sudo] пароль для adutkina:
[root@fedora ~]# df install git
Последняя проверка окончания срока действия метаданных: 2:35:19 назад,
Пт 17 фев 2023 18:25:44.
Пакет git-2.39.1-1.fc36.x86_64 уже установлен.
Зависимости разрешены.
Отсутствуют действия для выполнения.
Выполнено!
```

Рис. 3.1: Установка `git`

Установим `gh` (рис. 3.2):

```
Установлен:
gh-2.22.1-1.fc36.x86_64
Выполнено!
[root@fedora ~]#
```

Рис. 3.2: Установка `gh`

Рис. 3.6: Результат создания изображений в формате pdf

3.4 Окончание работы, создание дополнительных форматов отчета

После заполнения шаблона сохраняем файл и переходим в терминал. При использовании команды `make` создаются два дополнительных формата с расширениями `.docx` и `.pdf` (рис. 3.7). Для удаления лишних файлов, созданных по шаблону, используется “`make clean`” (рис. 3.8).


```
[adutkina@fedora report]$ make
pandoc "report.md" --filter pandoc/filters/pandoc_fignos.py --filter pandoc/filters/pandoc_eqnos.py --filter pandoc/filters/pandoc_tablenos.py --filter pandoc/filters/pandoc_secnos.py --number-sections --citeproc -o "report.docx"
pandoc "report.md" --filter pandoc/filters/pandoc_fignos.py --filter pandoc/filters/pandoc_eqnos.py --filter pandoc/filters/pandoc_tablenos.py --filter pandoc/filters/pandoc_secnos.py --pdf-engine=lualatex --pdf-engine-opt=--shell-escape --citeproc --number-sections -o "report.pdf"
[adutkina@fedora report]$
```

Рис. 3.7: Создание дополнительных форматов отчета

```
[adutkina@fedora report]$ make clean
rm report.docx report.pdf *~
rm: невозможно удалить '*~': Нет такого файла или каталога
make: [Makefile:34: clean] Ошибка 1 (игнорирование)
```

Рис. 3.8: Удаление дополнительных форматов отчета

4 Выводы

В ходе данной лабораторной работы были приобретены навыки оформления отчётов с помощью легковесного языка разметки Markdown.