

Презентация по лабораторной работе №3

Работу выполнил Шеожев Аслан Аскерович

Цель работы:

- Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

Задание:

- Сделать отчёт по предыдущей лабораторной работе в формате Markdown.
- В качестве отчёта предоставить отчёты в 3 форматах: pdf, docx и md.

Ход работы:

1. Изучил базовые сведения о Markdown.
2. Установил и настроил необходимое ПО.
3. Записал отчет в Markdown.
4. Перевел файл в pdf и docx.

Скриншоты работы:

```
1 ---
2 # Front matter
3 lang: ru-RU
4 title: "Лабораторная работа 1"
5 subtitle: "Markdown"
6 author: "Васильев Аслан Аскерович"
7
8 # Formatting
9 toc-title: "Содержание"
10 toc: true # Table of contents
11 toc_depth: 2
12 fontsize: 12pt
13 linestretch: 1.5
14 papersize: a4paper
15 documentclass: scrreprt
16 polyglossia-lang: russian
17 polyglossia-otherlangs: english
18 mainfont: PT Serif
19 romanfont: PT Serif
20 sansfont: PT Sans
21 monofont: PT Mono
22 mainfontoptions: Ligatures=TeX
23 romanfontoptions: Ligatures=TeX
24 sansfontoptions: Ligatures=TeX,Scale=MatchLowercase
25 monofontoptions: Scale=MatchLowercase
26 indent: true
27 pdf-engine: lualatex
28 header-includes:
29 - \linepenalty=10 # the penalty added to the badness of each line within a
30   paragraph (no associated penalty node) Increasing the value makes tex try to
31   have fewer lines in the paragraph.
32 - \interlinepenalty=0 # value of the penalty (node) added after each line of a
33   paragraph.
34 - \hyphenpenalty=50 # the penalty for line breaking at an automatically
35   inserted hyphen
36 - \xhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen
37 - \binoppenalty=700 # the penalty for breaking a line at a binary operator
38 - \relpenalty=500 # the penalty for breaking a line at a relation
39 - \clubpenalty=150 # extra penalty for breaking after first line of a paragraph
40 - \widowpenalty=150 # extra penalty for breaking before last line of a paragraph
41 - \displaywidowpenalty=50 # extra penalty for breaking before last line before
42   a display math
43 - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line
44 - \predisplaypenalty=10000 # penalty for breaking before a display
45 - \postdisplaypenalty=0 # penalty for breaking after a display
46 - \floatingpenalty = 20000 # penalty for splitting an insertion (can only be
47   split footnote in standard LaTeX)
48 - \raggedbottom # or \flushbottom
49 - \usepackage{float} # keep figures where there are in the text
```

Рис. 1

```
43 - \usepackage{float} # keep figures where there are in the text
44 - \floatplacement{figure}{H} # keep figures where there are in the text
45 ---
46
47 # Отчет по лабораторной работе #2.
48 ## Работу выполнил Васильев Аслан Аскерович.
49
50 # Цель работы:
51 Изучить идеологию и применение средств контроля версий, а также освоить умения по
52 работе с git.
53 # Задание:
54 * Создать базовую конфигурацию для работы с git.
55 * Создать ключ SSH.
56 * Создать ключ PGP.
57 * Настроить подписи git.
58 * Зарегистрироваться на Github.
59 * Создать локальный каталог для выполнения заданий по предмету.
60
61 # Ход работы:
62 1. Ознакомился с основными командами git и установил все необходимое ПО
63   (get-flow, gh). Настроил github. Создал учетную запись и заполнил основные данные.
64 2. Задан имя и email владельца репозитория, настроил utf-8 в выводе сообщений
65   git, задан имя начальной ветки (будем называть её master). Создал ключи ssh по
66   алгоритму rsa с ключом размера 4096 бит и по алгоритму ed25519.
67   [[Рис. 1 (ввод основных сведений о программисте, а также настройка -utf8)]](image/
68   4.jpg) { width=100% } Рис. 1 (ввод основных сведений о программисте, а также
69   настройка -utf8)
70 3. Создал ключи pgp. Тип - RSA and RSA. Размер - 4096. Срок действия не истекает
71   никогда
72   [[Рис. 2 (создал ключи ppg)]](image/8.jpg) { width=100% } Рис. 2 (создал ключи
73   BPE)
74 4. Добавил ppg и ssh ключи в github.
75   [[Рис. 3 (введенные ключи на github)]](image/7.jpg) { width=100% } Рис. 3
76   (введенные ключи на github)
77 5. Используя введенный email, указал git применять его при подписи коммитов.
78   [[Рис. 4 (для подписи коммитов)]](image/11.jpg) { width=100% } Рис. 4 (для подписи
79   коммитов)
80 6. Приступил к настройке gh.
81 7. Создал шаблон рабочего пространства, по примеру из условия лабораторной
82   работы. А после настроил каталог курса.
83   [[Рис. 5 (результат на странице репозитория в github)]](image/10.jpg) { width=100%
84   } Рис. 5 (результат на странице репозитория в github)
85
86 # Вывод:
87 Во второй лабораторной работе я изучил идеологию и применение средств контроля
88 версий, освоил умения по работе с git.
89
90 # Контрольные вопросы:
91 1. Что такое системы контроля версий (VCS) и для решения каких задач они
92   предназначены?
93 2. Система контроля версий (VCS) нужна для работы нескольких людей над одним
```

Рис. 2

```
76 - Система контроля версий (VCS) нужна для работы нескольких людей над одним
77   проектом, совместная работа путем изменения файлов в репозитории.
78 2. Объясните следующие понятия VCS и их отношения: хранилище, commit, история,
79   рабочая копия.
80 - хранилище - это пространство для хранения файлов;
81 - commit - команда для записи индексированных изменений в репозитории;
82 - история - в ней расположены все коммиты, по которым можно отследить автора,
83   дату и сообщение;
84 - рабочая копия - все файлы, кроме .git/ это рабочая копия.
85 3. Что представляет собой и чем отличается централизованные и децентрализованные
86   VCS? Приведите примеры VCS каждого вида.
87 - Централизованные системы контроля версий - проект и его файлы на 1 сервере, а
88   децентрализованные системы контроля версий - при копировании данных данных
89   удаленного репозитория, происходит полное копирование данных в локальный
90   репозиторий.
91 4. Опишите действия с VCS при единовременной работе с хранилищем.
92 - Создание репозитория, добавление в них файлов, фиксация с помощью коммитов,
93   изменение файлов репозитория.
94 5. Опишите порядок работы с общим хранилищем VCS.
95 - Создание репозитория, добавление файлов в него файлов, фиксация коммитами,
96   изменение файлов репозитория, ожидание проверки коммитов.
97 6. Каковы основные задачи, решаемые инструментальным средством git?
98 - Систематизация и возможность разработки параллельно.
99 7. Назовите и дайте краткую характеристику командам git.
100 - Создание репозитория (git init), клонирование (git clone), добавление изменений
101   в индекс (git add), коммиты (git commit), удаление файла (git rm), удаление
102   изменений из индекса (git reset).
103 8. Приведите примеры использования при работе с локальным и удаленным
104   репозиториями.
105 - Для написания черновика по лабораторным работам я могу использовать репозиторий.
106 9. Что такое и зачем могут быть нужны ветви (branches)?
107 - Ветки служат для реализации параллельного программирования.
108 10. Как и зачем можно игнорировать некоторые файлы при commit?
109 - Игнорирование нужно для неотслеживания файлов служебного типа.
```

Рис. 3