

Отчет 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.

Скриншот отчета в Markdown:

```

1  ---
2  # Front matter
3  lang: ru-RU
4  title: "Лабораторная работа 3"
5  subtitle: "Markdown"
6  author: "Шеожев 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
      paragraph (no associated penalty node) Increasing the value makes tex try to
      have fewer lines in the paragraph.
30     - \interlinepenalty=0 # value of the penalty (node) added after each line of a
      paragraph.
31     - \hyphenpenalty=50 # the penalty for line breaking at an automatically
      inserted hyphen
32     - \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen
33     - \binoppenalty=700 # the penalty for breaking a line at a binary operator
34     - \relpenalty=500 # the penalty for breaking a line at a relation
35     - \clubpenalty=150 # extra penalty for breaking after first line of a paragraph
36     - \widowpenalty=150 # extra penalty for breaking before last line of a paragraph
37     - \displaywidowpenalty=50 # extra penalty for breaking before last line before
      a display math
38     - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line
39     - \predisplaypenalty=10000 # penalty for breaking before a display
40     - \postdisplaypenalty=0 # penalty for breaking after a display
41     - \floatingpenalty = 20000 # penalty for splitting an insertion (can only be
      split footnote in standard LaTeX)
42     - \raggedbottom # or \flushbottom
43     - \usepackage{float} # keep figures where there are in the text

```

{ width=100% } Рис. 1 (Отчет 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 Шеожев Аслан Аскерович.
49
50 # Цель работы:
51 Изучить идеологию и применение средств контроля версий, а также освоить умения по
  работе с git.
52 # Задание:
53 * Создать базовую конфигурацию для работы с git.
54 * Создать ключ SSH.
55 * Создать ключ PGP.
56 * Настроить подписи git.
57 * Зарегистрироваться на Github.
58 * Создать локальный каталог для выполнения заданий по предмету.
59 # Ход работы:
60 1. Ознакомился с основными командами git и установил все необходимое ПО
  (get-flow, gh). Настроил github. Создал учетную запись и заполнил основные данные.
61 2. Задал имя и email владельца репозитория, настроил utf-8 в выводе сообщений
  git, задал имя начальной ветки (будем называть её master). Создал ключи ssh по
  алгоритму rsa с ключём размера 4096 бит и по алгоритму ed25519.
62 ![Рис. 1 (ввод основных сведений о программисте, а также настройка -utf8)](image/
  4.jpg) { width=100% } Рис. 1 (ввод основных сведений о программисте, а также
  настройка -utf8)
63 3. Создал ключи ppg. Тип – RSA and RSA. Размер – 4096. Срок действия не истекает
  никогда
64 ![Рис. 2 (создал ключи gpg)](image/8.jpg) { width=100% } Рис. 2 (создал ключи
  gpg)
65 4. Добавил ppg и ssh ключи в github.
66 ![Рис. 3 (введенные ключи на github)](image/7.jpg) { width=100% } Рис. 3
  (введенные ключи на github)
67 5. Используя введенный email, указал git применять его при подписи коммитов.
68 ![Рис. 4 (для подписи коммитов)](image/11.jpg) { width=100% } Рис. 4 (для подписи
  коммитов)
69 6. Приступил к настройке gh.
70 7. Создал шаблон рабочего пространства, по примеру из условия лабораторной
  работы. А после настроил каталог курса.
71 ![Рис. 5 (результат на странице репозитория в github)](image/10.jpg) { width=100%
  } Рис. 5 (результат на странице репозитория в github)
72 # Вывод:
73 Во второй лабораторной работе я изучил идеологию и применение средств контроля
  версий, освоил умения по работе с git.
74 # Контрольные вопросы:
75 1. Что такое системы контроля версий (VCS) и для решения каких задач они
  предназначены?
76 - Система контроля версий (VCS) нужны для работы нескольких людей над одним

```

{ width=100% } Рис. 2 (Отчет 2)


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

{ width=100% } Рис. 3 (Отчет 3)

Вывод:

Я научился оформлять отчёты с помощью легковесного языка разметки Markdown.

Контрольные вопросы:

Не предусмотрены.