Отчёт по лабораторной работе №3.

Markdown

Александр Андреевич Шуплецов

Содержание

1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

2 Теоретическое введение

- Титульный лист. Первый лист работы оформляется строго по образцу, который обычно приводится в методических пособиях по вашему предмету. В нем не просто требуется указать такие элементы, как название образовательного учреждения, вид работы и сведения об исполнителе, но и расположить их в строгом соответствии со стандарта- ми. – Реферат. Реферат фактически является кратким представлением всего вашего отчета и содержит ряд статистических сведений. В нем нужно указать количество частей, страниц работы, иллюстраций, приложений, таблиц, использованных литературных источников и приложений. Здесь же приводится перечень ключевых слов работы и собственно текст реферата. Последний подразумевает основные элементы работы от поставленных целей до результатов и рекомендаций по их внедрению. В практике вузов в отчеты по лабораторным работам реферат обычно не включают. – Введение. Во введении типовой лабораторной работы обычно прописывают цели проводимого исследования и задачи, выполнение которых поможет достичь постав- ленных целей. В то же время существуют работы, в которых студенты становятся настоящими первооткрывателями. Приходилось ли вам хотя бы однажды испытывать чувство крайнего любопытства и нетерпения при проведении лабораторной работы? Ощущать, что буквально через пару минут вы найдете ответ на вопрос, на который еще никто и никогда не находил ответа? Именно для таких исследований пишется раз- вернутое введение с доказательством актуальности и новизны изучаемой темы. Чтобы

действительно провести исследование в той области, в которой, как говорится, еще не ступала нога человека, во введении вам понадобится привести оценку современного состояния рассматриваемой проблемы и обосновать необходимость ее решения. – Основная часть. Так как в разных вузах и в разных дисциплинах существуют свои тонкости проведения лабораторных работ, содержание основной части подробно описывают в соответствующих методичках. Важно, чтобы в этом разделе работы была отражена ее суть, описана методика и результаты проделанной работы. В основной части прописывают следующие элементы: – цели проводимого исследования; – задачи, выполнение которых поможет достичь поставленных целей; – ход работы, в котором описываются выполненные действия; – прочие разделы, предусмотренные методическими материалами по изучаемой дисциплине. – Заключение. В этой части работы вам потребуется сделать выводы по полученным в хо- де лабораторной работы результатам. Для этого оцените, насколько полно выполнены поставленные задачи. В сложных работах могут присутствовать и другие элементы, например, рекомендации для дальнейшего применения результатов проведённой работы.

3 Задание

– Сделайте отчёт по предыдущей лабораторной работе в формате Markdown. – В качестве отчёта просьба предоставить отчёты в 3 форматах: pdf, docx и md (в архиве, поскольку он должен содержать скриншоты, Makefile и т.д.)

4 Выполнение лабораторной работы

1. Оформим титульный лист.

```
## Front matter
title: "Отчёт по лабораторной работе №2."
subtitle: "Первоначальная настройка Git"
author: "Александр Андреевич Шуплецов"
```

Figure 1: титульный лист

2. Оформим цели работы.

```
68
69 # Цель работы
70
71 Целью данной работы является изучить идеологию и применение средств контроля версий и освоить умения по работе с git.
72
```

Figure 2: цель работы

3. Оформим теоретическое введение.

```
73 # Теоретическое введение
74 Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для
   участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта,
   если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется
   специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом
   предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения
   между последовательными версиями, что позволяет уменьшить объём хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним
   файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка
   не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с
   файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки
   ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к
   которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git,
   Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.
```

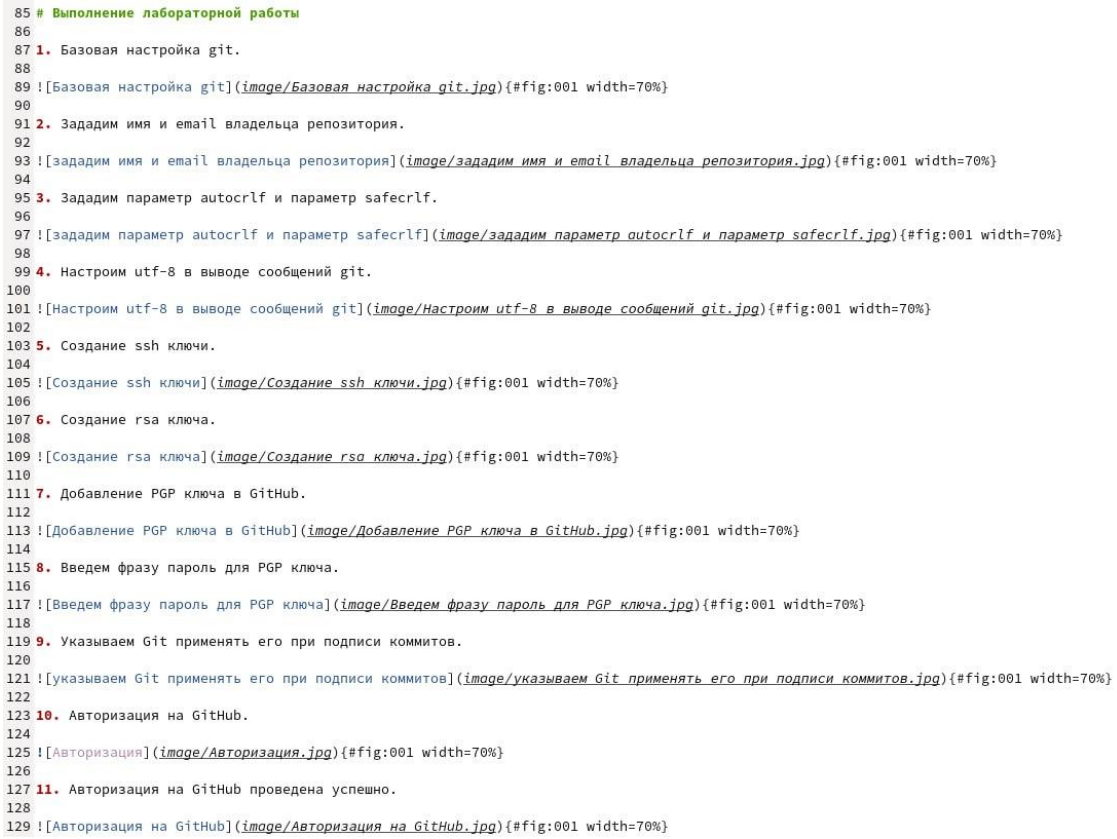
Figure 3: теоретическое введение

4. Оформим задание работы.

```
76 # Задание
77
78 1. Создать базовую конфигурацию для работы с git.
79 2. Создать ключ SSH.
80 3. Создать ключ PGP.
81 4. Настроить подписи git.
82 5. Зарегистрироваться на Github.
83 6. Создать локальный каталог для выполнения заданий по предмету.
84
```

Figure 4: задание

5. Оформим выполнение работы.



```
85 # Выполнение лабораторной работы
86
87 1. Базовая настройка git.
88
89 ![Базовая настройка git](image/Базовая настройка git.jpg){#fig:001 width=70%}
90
91 2. Зададим имя и email владельца репозитория.
92
93 ![зададим имя и email владельца репозитория](image/зададим имя и email владельца репозитория.jpg){#fig:001 width=70%}
94
95 3. Зададим параметр autocrlf и параметр safecrlf.
96
97 ![зададим параметр autocrlf и параметр safecrlf](image/зададим параметр autocrlf и параметр safecrlf.jpg){#fig:001 width=70%}
98
99 4. Настроим utf-8 в выводе сообщений git.
100
101 ![Настроим utf-8 в выводе сообщений git](image/Настроим utf-8 в выводе сообщений git.jpg){#fig:001 width=70%}
102
103 5. Создание ssh ключи.
104
105 ![Создание ssh ключи](image/Создание ssh ключи.jpg){#fig:001 width=70%}
106
107 6. Создание rsa ключа.
108
109 ![Создание rsa ключа](image/Создание rsa ключа.jpg){#fig:001 width=70%}
110
111 7. Добавление PGP ключа в GitHub.
112
113 ![Добавление PGP ключа в GitHub](image/Добавление PGP ключа в GitHub.jpg){#fig:001 width=70%}
114
115 8. Введем фразу пароль для PGP ключа.
116
117 ![Введем фразу пароль для PGP ключа](image/Введем фразу пароль для PGP ключа.jpg){#fig:001 width=70%}
118
119 9. Указываем Git применять его при подписи коммитов.
120
121 ![указываем Git применять его при подписи коммитов](image/указываем Git применять его при подписи коммитов.jpg){#fig:001 width=70%}
122
123 10. Авторизация на GitHub.
124
125 ![Авторизация](image/Авторизация.jpg){#fig:001 width=70%}
126
127 11. Авторизация на GitHub проведена успешно.
128
129 ![Авторизация на GitHub](image/Авторизация на GitHub.jpg){#fig:001 width=70%}
```

Figure 5: выполнение работы

6. Оформим выводы работы.

```
151 # Выводы
152
153 Я изучил идеологию и применение средств контроля версий, освоил умения по работе с git.
```

Figure 6: выводы

7. Оформим контрольные вопросы.

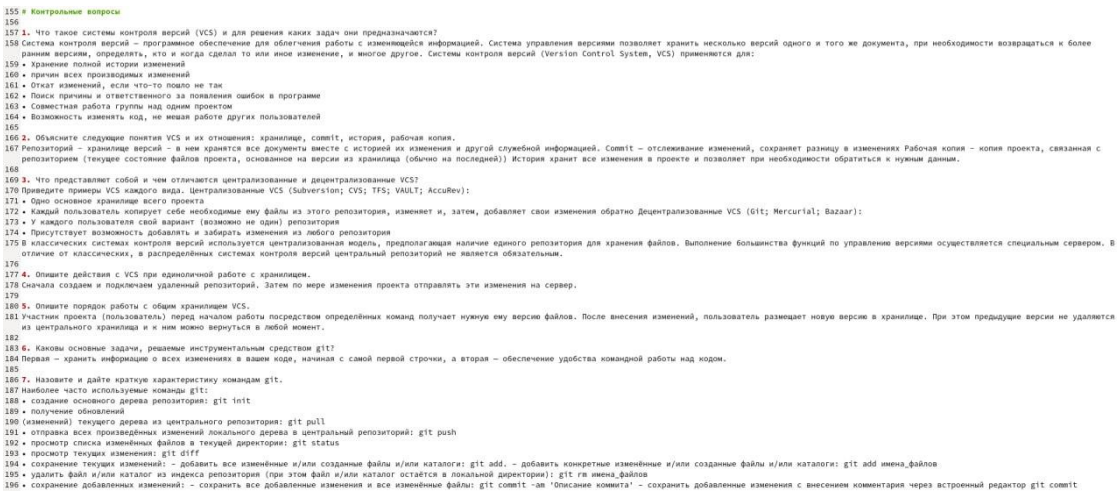


Figure 7: контрольные вопросы

8. Оформим список литературы.

```
212 # Список литературы{.unnumbered}
213
214 Кулябов Д.С. "Материалы к лабораторной работе"
```

Figure 8: список литературы

5 Выводы

Я научился оформлять отчёты с помощью легковесного языка разметки Markdown.

Список литературы

Кулябов Д.С. “Материалы к лабораторной работе”