

# **Лабораторная работа №3**

**Markdown**

Козлов Всеволод Павлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>11</b>

## Список иллюстраций

3.1	Создание титульного листа . . . . .	7
3.2	Прописал цель работы и задание . . . . .	8
3.3	Ход лабораторной работы №2 . . . . .	9
3.4	Ответы на контрольные вопросы . . . . .	9
3.5	Формулировка вывода . . . . .	10

## Список таблиц

# 1 Цель работы

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

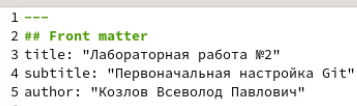
## 2 Задание

Объяснить создание отчета по второй лабораторной работе в Markdown.

## 3 Выполнение лабораторной работы

### 1) Создание титульного листа:

Прописал номер лабораторной работы, ее название и автора (рис. [3.1])



```
1 ---
2 ## Front matter
3 title: "Лабораторная работа №2"
4 subtitle: "Первоначальная настройка Git"
5 author: "Козлов Всеволод Павлович"
6
```

Рис. 3.1: Создание титульного листа

### 2) Цель работы и задание:

Сформулировал цель работы и задание (рис. [3.2])

```
68
69 # Цель работы
70
71 Изучить идеологию и применение средств контроля версий, освоить умения по работе с git.
72
73 # Задание
74
75 1) Создать базовую конфигурацию для работы с git.
76 2) Создать ключ SSH.
77 3) Создать ключ PGP.
78 4) Зарегистрироваться на Github.
79 5) Создать локальный каталог для выполнения заданий по предмету.
80
```

Рис. 3.2: Прописал цель работы и задание

### 3) Ход выполнения второй лабораторной работы:

Создание описания хода лабораторной работы №2 (частичный скриншот хода лабораторной работы) (рис. [3.3])



```

81 # Выполнение лабораторной работы
82
83 1) Установка программного обеспечения:
84
85 Произвел установку Git через терминал (рис. [-@fig:001])
86
87 ![Установка Git](image/001_g_d.png) { #fig:001 width=70% }
88
89 Произвел установку gh (рис. [-@fig:002])
90
91 ![Установка gh](image/002_gh_d.png) { #fig:002 width=70% }
92
93 2) Базовая настройка Git:
94
95 Задал имя и почту пользователя (рис. [-@fig:003])
96
97 ![Имя и почта пользователя](image/003_settings.png) { #fig:003 width=70% }
98
99 Настроил utf-8 в выводе сообщений git (рис. [-@fig:004])
100
101 ![Настройка utf-8](image/004_utf.png) { #fig:004 width=70% }
102
103 Создал GPG ключ (рис. @fig:005)
104
105 ![Создание GPG ключа](image/005_gpg_key.png) { #fig:005 width=70% }
106
107 Экспортировал GPG ключ (рис. [-@fig:006])
108
109 ![Экспорт ключа](image/006_key_export.png) { #fig:006 width=70% }
110
111 Скопировал GPG ключ (рис. @[-fig:007])
112
113 ![Копирование ключа](image/007_key_kopy.png) { #fig:007 width=70% }
114
115 Передал GPG ключ на Github (рис. [-@fig:008])
116
117 ![Передача ключа на Github](image/008_key_to_git.png) { #fig:008 width=70% }
118
119 Настройка автоматических подписей коммитов git (рис. [-@fig:009])

```

Рис. 3.3: Ход лабораторной работы №2

Отвечал на контрольные вопросы (частичный скриншот ответов на вопросы)  
(рис. [3.4])

```

166 Ответы на контрольные вопросы:
167
168 1) Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?
169
170 Система контроля версий – программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. Системы контроля версий (Version Control System, VCS) применяются для:
• Хранение полной истории изменений
• причин всех производимых изменений
• Откат изменений, если что-то пошло не так
• Поиск причины и ответственного за появления ошибок в программе
• Совместная работа группы над одним проектом
• Возможность изменять код, не мешая работе других пользователей
171
172 2) Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.
173
174 Репозиторий – хранилище версий – в нем хранятся все документы вместе с историей их изменения и другой служебной информацией. Commit – отслеживание изменений, сохраняет разницу в изменениях Рабочая копия – копия проекта, связанная с репозиторием (текущее состояние файлов проекта, основанное на версии из хранилища (обычно на последней)) История хранит все изменения в проекте и позволяет при необходимости обратиться к нужным данным.
175

```

Рис. 3.4: Ответы на контрольные вопросы

4) Вывод к лабораторной работе №2:

Сформулировал вывод ко второй лабораторной работе (рис. [3.5])

```
207 /
208 # Выводы
209
210 Изучил идеологию и применение средств контроля версий, освоил умения по работе с git.
211
```

Рис. 3.5: Формулировка вывода

## 4 Выводы

Научился оформлять отчёты с помощью легковесного языка разметки Markdown.