

Отчёт по лабораторной работе 2

Хамбалеев Булат Галимович

1 октября, 2022

Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

Лабораторная работа подразумевает использование Markdown для оформления отчёта.

Выполнение лабораторной работы

Выполнение лабораторной работы

1. Установим настройки шаблона.(рис. 1)

```
---  
# Front matter  
title: "Отчет по лабораторной работе номер 1"  
author: "Хамбалиев Булат Галимович"  
  
# Generic options  
lang: ru-RU  
toc-title: "Содержание"  
  
# Pdf output format  
toc: true # Table of contents  
toc-depth: 2  
lof: true # List of figures  
lot: true # List of tables  
fontsize: 12pt  
linestretch: 1.5  
papersize: a4  
documentclass: scrreprt  
## I18n  
polyglossia-lang:  
  name: russian  
  options:  
    - spelling=modern  
    - babelshorthands=true  
polyglossia-otherlangs:  
  name: english  
### Fonts  
mainfont: Ubuntu  
romanfont: Ubuntu  
sansfont: Ubuntu  
monofont: Ubuntu  
mainfontoptions: Ligatures=TeX  
romanfontoptions: Ligatures=TeX  
sansfontoptions: Ligatures=TeX,Scale=MatchLowercase  
monofontoptions: Scale=MatchLowercase,Scale=0.9
```

2. Запишем цели, задание и теорию.(рис.2)

```
- hyperref=auto
- language=auto
- autolang=other*
- citestyle=gost-numeric
## Misc options
indent: true
header-includes:
- \linepenalty=10 # the penalty added to the badness of each line within a paragraph (no associate
- \interlinepenalty=0 # value of the penalty (node) added after each line of a paragraph.
- \hyphenpenalty=50 # the penalty for line breaking at an automatically inserted hyphen
- \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen
- \binoppenalty=700 # the penalty for breaking a line at a binary operator
- \relpenalty=500 # the penalty for breaking a line at a relation
- \clubpenalty=150 # extra penalty for breaking after first line of a paragraph
- \widowpenalty=150 # extra penalty for breaking before last line of a paragraph
- \displaywidowpenalty=50 # extra penalty for breaking before last line before a display math
- \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line
- \predisplaypenalty=10000 # penalty for breaking before a display
- \postdisplaypenalty=0 # penalty for breaking after a display
- \floatingpenalty = 20000 # penalty for splitting an insertion (can only be split footnote in sta
- \raggedbottom # or \flushbottom
- \usepackage{float} # keep figures where there are in the text
- \floatplacement{figure}{N} # keep figures where there are in the text
...

# Цель работы

Изучить идеологию и применение средств контроля версий. Освоить умения по работе с git.

# Задание

Лабораторная работа подразумевает использование git для создания репозитория для лабораторных работ.

# Теория

Git (произносится «гит») – распределённая система управления версиями. Проект был создан Линусом Торвалдом.

# Выполнение работы

1. Создадим базовую конфигурацию для работы с git.(рис 1-2)
```

3. Запишем ход работы.(рис 3)

```
![рис.8. Ввод ключа.](images/8.jpg){ #fig:008 width=90% }

![рис.9. Ввод ключа.](images/9.jpg){ #fig:009 width=90% }

# Библиография

1. ТУИС РУДН

# Выводы

Во время выполнения лабораторной работы я освоил на практике git и создал репозиторий для лабораторной работы.

#Контрольные вопросы

1) Что такое системы контроля версий (VCS) и для решения каких задач они предназначаются?

• Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом.

2) Объясните следующие понятия VCS и их отношения: хранилище, commit, история, рабочая копия.

• Хранилище (repository), или репозиторий, —место хранения всех версий и служебной информации. Commit — запись изменений. История — последовательность записей. Рабочая копия — текущая версия файлов.

3) Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры.

• Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда есть один центральный сервер, который хранит все данные. Клиенты работают с сервером.

• В децентрализованных системах каждый узел принимает свое собственное решение. Конечное поведение системы определяется большинством узлов.

• В классических системах контроля версий используется централизованная модель, предполагающая наличие центрального сервера.

4) Опишите действия с VCS при одиночной работе с хранилищем.

• Создадим локальный репозиторий. Сначала сделаем предварительную конфигурацию, указав имя и email в файле ~/.gitconfig

git config --global user.name "Имя Фамилия"

git config --global user.email "work@mail"
```

4. Запишем ответы на контрольные вопросы. (рис. 4)

```
git add имена_файлов
- удалить файл и/или каталог из индекса репозитория (при этом файл и/или каталог
остаётся в локальной директории):

git rm имена_файлов

- сохранение добавленных изменений:

- сохранить все добавленные изменения и все изменённые файлы:
git commit -am 'Описание коммита'

- сохранить добавленные изменения с внесением комментария через встроенный
редактор:

git commit

- создание новой ветки, базирующейся на текущей:
git checkout -b имя_ветки

- переключение на некоторую ветку:

git checkout имя_ветки

- отправка изменений конкретной ветки в центральный репозиторий:

git push origin имя_ветки

- слияние ветки с текущим деревом:

git merge --no-ff имя_ветки

8) Приведите примеры использования при работе с локальным и удалённым репозиториями.

- Использование git при работе с локальными репозиториями (добавления текстового документа в локальный
репозиторий)

git add hello.txt

git commit -am 'Новый файл'

9) Что такое и зачем могут быть нужны ветки (branches)?
```


Спасибо за внимание