

Научная презентация

Лабораторная работа №3

Дикач А.О.

23.02.2023г.

Российский университет дружбы народов, Москва, Россия

Информация

- Дикач Анна Олеговна
- ученик НПИбд-01-22
- Российский университет дружбы народов
- [1132222009@pfur.ru]
- https://github.com/ANNdamn/study_2022-2023_os-intro

- Базовые сведения о Markdown
- Развитие навыков быстрого составления отчётов и презентаций

- Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.


```
---  
## Front matter  
title: "Очёт по лабораторной работе"  
subtitle: "Лабораторная работа №2"  
author: "Дикач Анна Олеговна"
```

Рис. 1: оформление титульного листа

```
---  
  
# Цель работы  
  
Изучить идеологию и приминение средств контроля версий, а также освоить умения по работе с git.
```

Рис. 2: формулировка целей

Выполнение лабораторной работы

Так как аккаунт в [Github](#) был создан и настроен в прошлом семестре, приступаю к созданию [репозитория](#) курса на основе шаблона (рис. [-@fig:001])

![созданный аккаунт в [Github](#)]([image/pic1.png](#)) { #fig:001 width=70% }

Рис. 3: скриншот Git

Описание хода выполнения лабораторной работы

```
## Создание репозитория курса на основе шаблона

1. Создаю шаблон рабочего пространства (рис. [-@fig:002]) (рис. [-@fig:003])

![создание репозитория курса](image/pic2.png){ #fig:002 width=70% }

![создание репозитория курса](image/pic3.png){ #fig:003 width=70% }

## Настройка каталога курса

2. Перехожу в каталог курса (рис. [-@fig:004])

![команда для перехода в каталог курса ](image/pic4.png){ #fig:004 width=70% }

3. Удаляю лишние файлы (рис. [-@fig:005])

![команда для удаления лишних файлов ](image/pic5.png){ #fig:005 width=70% }

4. Создаю необходимые каталоги (рис. [-@fig:006])

![команда для создания каталогов ](image/pic6.png){ #fig:006 width=70% }

5. Отправляю файлы на сервер (рис. [-@fig:007])

![команда для отправки файлов на сервер](image/pic7.png){ #fig:007 width=70% }

![команда для отправки файлов на сервер](image/pic8.png){ #fig:008 width=70% }

6. Итог проведённой работы (рис. [-@fig:009])

![скринкаст проведённой работы](image/pic9.png){ #fig:009 width=70% }
```

Рис. 4: описание лабораторной работы

Ответ на контрольные вопросы

Контрольные вопросы

1. Системы контроля версий (VCS) применяются при работе нескольких человек над одним проектом. Система управления версиями позволяет хранить версии одного и того же документа, возвращаться к ним и следить за изменениями и фиксировать их.

2. хранилище – место хранения файлов с которым могут взаимодействовать все участники
commit – делает для проекта снимок текущего состояния изменений, добавленных в раздел проиндексированных файлов.
история – показывает операции, которые были произведены с объектом. имеет большое количество опций для поиска коммитов по разным критериям
рабочая копия – является снимком одной версии проекта. файлы извлекаются из сжатой базы данных в каталоге git и помещаются на диск для того чтобы их могли редактировать другие пользователи

3. Централизованные системы контроля версий представляют собой приложения типа клиент-сервер, когда репозиторий проекта существует в единственном экземпляре и хранится на сервере. Доступ к нему осуществляется через специальное клиентское приложение. Примеры:
CVS, Subversion

Децентрализованные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При этом можно выделить центральный репозиторий (условно), в который будут отправляться изменения из локальных и, с ними же эти локальные репозитории будут синхронизироваться. Примеры: Git и Mercurial

4. создания и работа с файлами, выгрузка их в хранилище. не требует создания рабочих копий. можно сделать проект конфедикальным, не доступным для других пользователей git.

5. создание репозитория, обновить проект, скопировать последнюю версию, внести изменения, разрешить конфликты, создать ветку.

6. основными задачами являются: проверка ветки, переключение на новую, написание кода, просмотр статусов, создание коммита, отправка

7.
git add – добавляет содержимое рабочего каталога в индекс
git status – показывает состояние файлов
git diff – используется для вычисления разницы между
git difftool – запускает внешнюю утилиту сравнения для показа различий двух деревьев
git commit – берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во вращающийся диск, формируя новую версию проекта. В этот момент создается файл commit в каталоге git, который содержит информацию о файлах, добавленных в индекс, и о том, кто их добавил.

Выводы

Создала аккаунт в [Github](#), изучила его идеологию и применение средств контроля версий.

Рис. 6: ВЫВОД

Научилась оформлять отчёт в Markdown, привыкла к его особенностям оформления, создала первый отчёт