Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины «Программирование на Python»
Вариант___

Выполнил:
Дудкин Константин Александрович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»
направление «Программное
обеспечение средств вычислительной
техники и автоматизированных
систем»,
очная форма обучения

_____
(подпись)

Руководитель практики:
Кандидат технических наук, доцент
кафедры инфокомуникаций, доцент
Воронкин Роман Александрович

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: Исследование основных возможностей Git и GitHub

Цель: Исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub

Порядок выполнения работы

1. Зашел на сайт github.com, вошел в свой аккаунт

2. Создал репозиторий «LW1»

3. Скачал git и проверил его версию

```
Code_Ra1der@CR-notebook MINGW64 ~
$ git version
git version 2.42.0.windows.2
```

Рисунок 1. Проверка версии git

4. Клонировал созданный репозиторий на свой компьютер

5. Проверил его статус

```
Code_Ra1der@CR-notebook MINGW64 ~/git/LW1 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

Рисунок 2. Использование команды git status

6. Изменил файл README.md, добавив в него свое ФИО и название группы

7. Написал простую программу на языке программирования Python и включил в репозиторий с .gitignore

8. Воспользовался командами «add .» и «commit», чтобы зафиксировать изменения в репозитории, а затем командой «push», чтобы опубликовать его

```
Code_Ra1der@CR-notebook MINGW64 ~/git/LW1 (main)
$ git add .

Code_Ra1der@CR-notebook MINGW64 ~/git/LW1 (main)
$ git commit -m "Added something"
[main 1d8c9a2] Added something
 3 files changed, 170 insertions(+), 1 deletion(-)
 create mode 100644 .gitignore
 create mode 100644 sum.py
```

Рисунок 3. Использование команд git add . и git commit

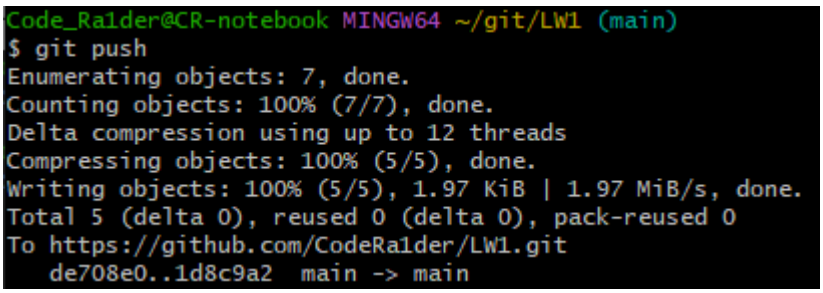

Рисунок 4. Использование команды git push

Ответы на вопросы

1. СКВ (Система контроля версий) – система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов

2. Главным недостатком локальных и централизованных СКВ является их централизованность на компьютере и сервере соответственно. Если компьютер или сервер выйдут из строя на время, то никто не сможет пользоваться контролем версий. А если данные СКВ будут повреждены, то будут потеряны не только текущая версия проекта, но и его прошлые версии тоже

3. Git относится к распределенным СКВ

4. Особенностью Git от других СКВ – собственный подход к файлам проекта. Хранение файлов в Git представляет собой сборник снимков миниатюрных файловых систем. Когда разработчик делает коммит, Git запоминает, какие файлы сохраняются, и делает их снимок и ссылку на них. Если некоторые файлы не были изменены, Git не делает их перезапись, а просто создает ссылку на них из предыдущей версии файла

5. Целостность хранимых данных в Git осуществляется путем использования хеш-суммы. Невозможно изменить файлы в Git так, чтобы он об этом не узнал. Для хеш-сумм используется SHA-1 хеш – строка длинной в 40 шестнадцатеричных символов

6. Файлы в Git могут быть в трех состояниях:

1) Зафиксированное (committed) – файлы в данном состоянии сохранены на локальной базе

2) Измененное (modified) – файлы в данном состоянии считаются измененными, но не зафиксированными

3) Подготовленное (staged) – файлы в данном состоянии готовы быть включены в следующий коммит

7. Профиль пользователя GitHub – учетная запись, в которой указана информация о самом пользователе, его репозиториях, активностях и личных данных на платформе GitHub

8. Репозитории бывают публичные – доступные для все пользователей GitHub, и приватные – доступные только владельцу и тем, кому он дал доступ к ним

9. Модель работы с GitHub представлена так:

- Создается репозиторий, выполняется работа над его настройкой и управление файлами

- Фиксируются изменения (коммит) и репозиторий отправляется на сервер (push), осуществляется работа с ветками

- Взаимодействие других пользователей с репозиторием

- Происходит слияние изменений (merge) и обновление репозитория

10. Первоначальная настройка Git включает в себя:

- Использование git config для установки имени пользователя и электронной почты GitHub

- Настройка глобальных параметров Git

- Создание SSH-ключа для безопасной аутентификации на GitHub

11. Создание репозитория происходит так:

- Вход в учетную запись GitHub

- В разделе репозиториев надо нажать на New для создания нового репозитория

- Заполняется имя репозитория, настраиваются описание, .gitignore, доступ к репозиторию и лицензия

- Создается репозиторий после нажатия Create repository

12. Существует множество типов лицензий в GitHub – MIT, Apache, GPL и т.д. Чтобы выбрать лицензию, надо при создании репозитория выбрать пункт выбор лицензии

13. Клонирование репозитория происходит с помощью команды git clone [ссылка на репозиторий]. Это необходимо для локального изменения файлов репозитория и последующего их коммита

14. Для проверки текущего состояния локального репозитория использую команду git status

15. Состояние репозитория изменяется следующим образом:

- Добавление/изменение файла – репозиторий меняет статус файлов на «Измененные»

- Добавление/изменение файла под версионный контроль (git add) – репозиторий меняет статус файлов на «Подготовленные»

- Фиксация изменений (git commit) – репозиторий изменяет статус файлов на «Зафиксированные»

- Отправка изменений на сервер (git push) – репозиторий на сервере и локальный репозиторий синхронизируются – на удалённом изменяются файлы в соответствии с изменёнными файлами в локальном

16. 1) Каждый из компьютеров выполняет клонирование репозитория (git clone)

2) Один из компьютеров делает изменения в нем, фиксирует изменения и делает коммит. Затем, выполняется команда git push

3) Другой компьютер выполняет команду git pull и получает изменения, сделанные первым компьютером

17. Кроме GitHub существует и несколько альтернативных ему сервисов. Вот примеры некоторых из них:

- GitLab – имеет свои собственные серверы, отличающиеся от GitHub, и дает множество инструментов DevOps

- Launchpad – платформа, созданная в основном для работы с репозиториями Ubuntu

- Bitbucket – позволяет работать с такими инструментами, как Jira, HipChat и Confluence

18. На данный момент мне известен только одно программное средство для работы с Git – GitHub Desktop

GitHub Desktop – официальный клиент для работы с Git от создателей GitHub. Главный плюс данного программного средства – наличие удобного графического интерфейса для работы с репозиториями

Вывод: Я изучил базовый функционал Git и научился создавать и редактировать репозитории с помощью Git и GitHub