Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Дисциплина «Системы искусственного интеллекта»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Исследование основных возможностей Git и GitHub

Выполнил (а):

Сыроватская Вероника Вадимовна

(ФИО)

студентка 3 курса 2 группы

специальность ППЖ-б-о-20-1

очной формы обучения

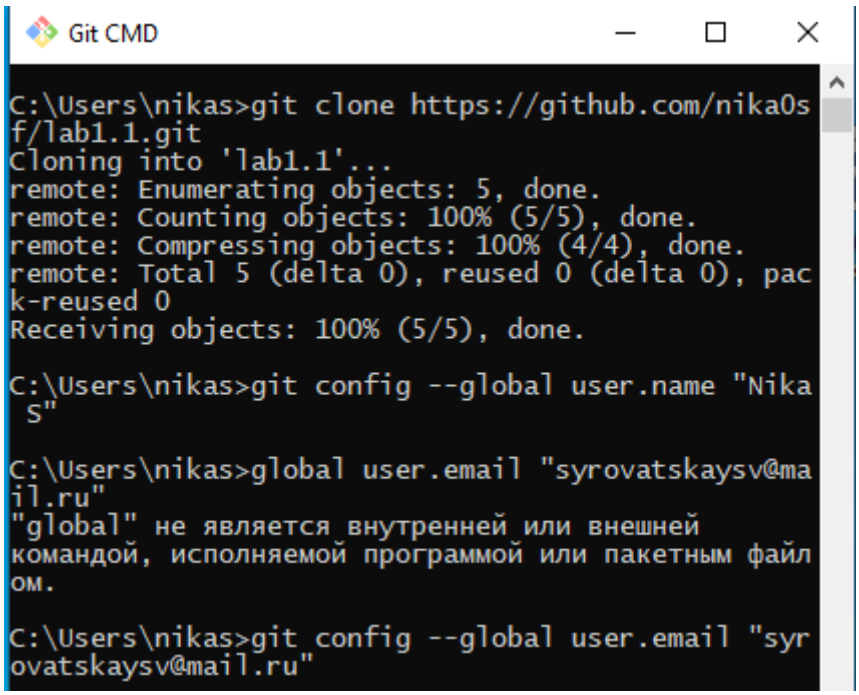
Проверил:

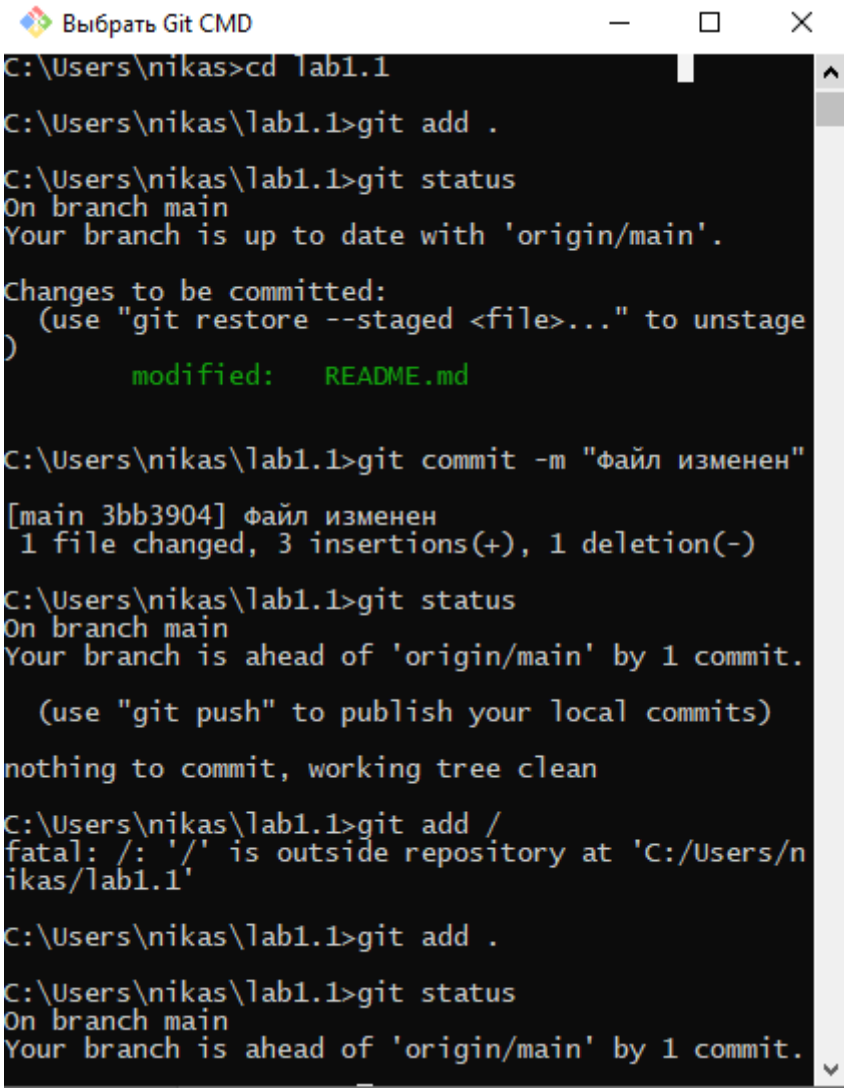
Кандидат технических наук

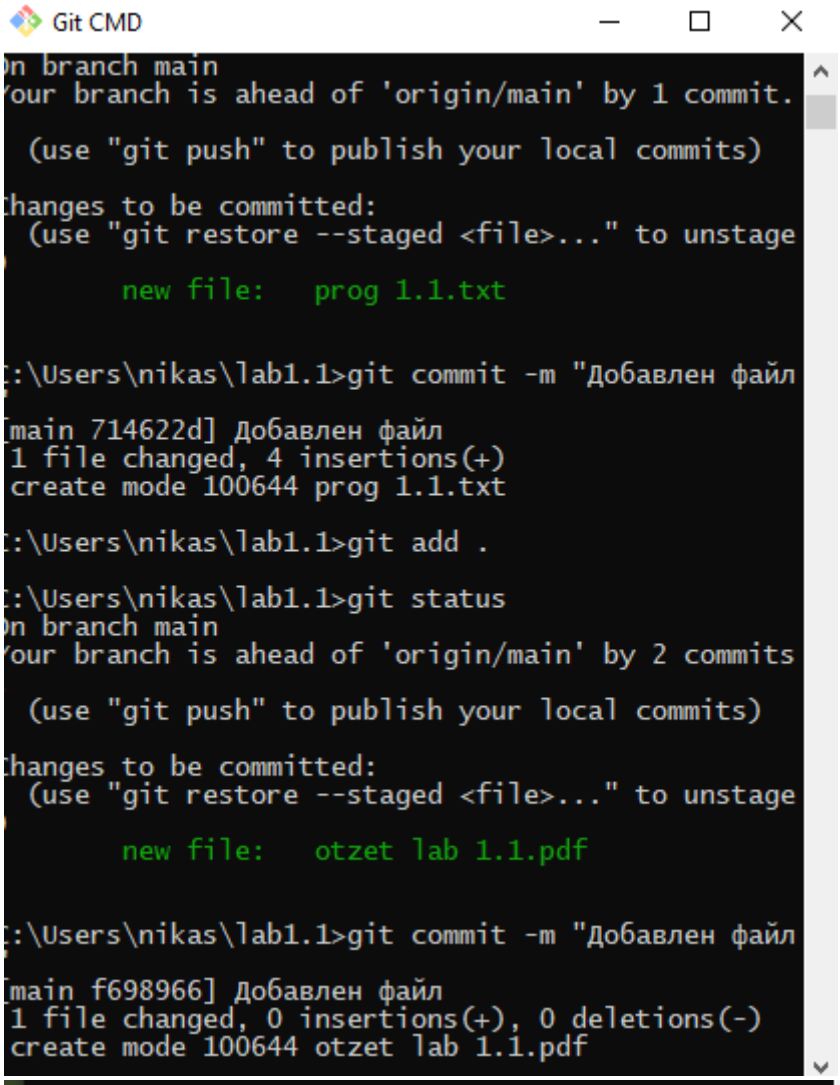
Доцент кафедры

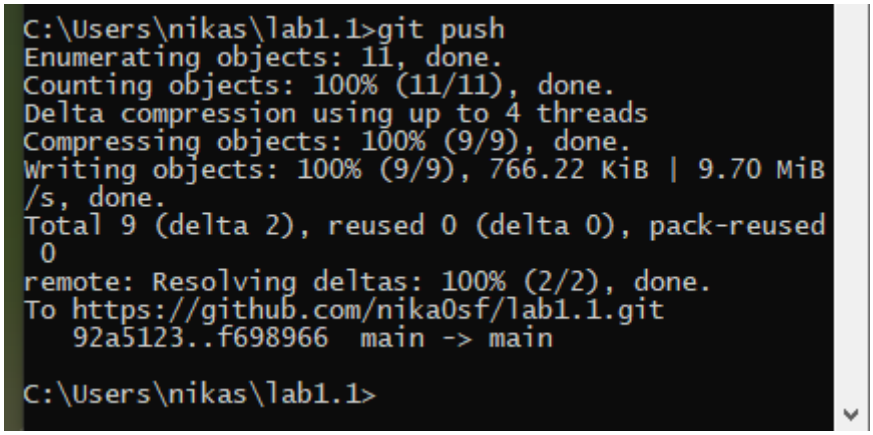
инфокоммуникаций

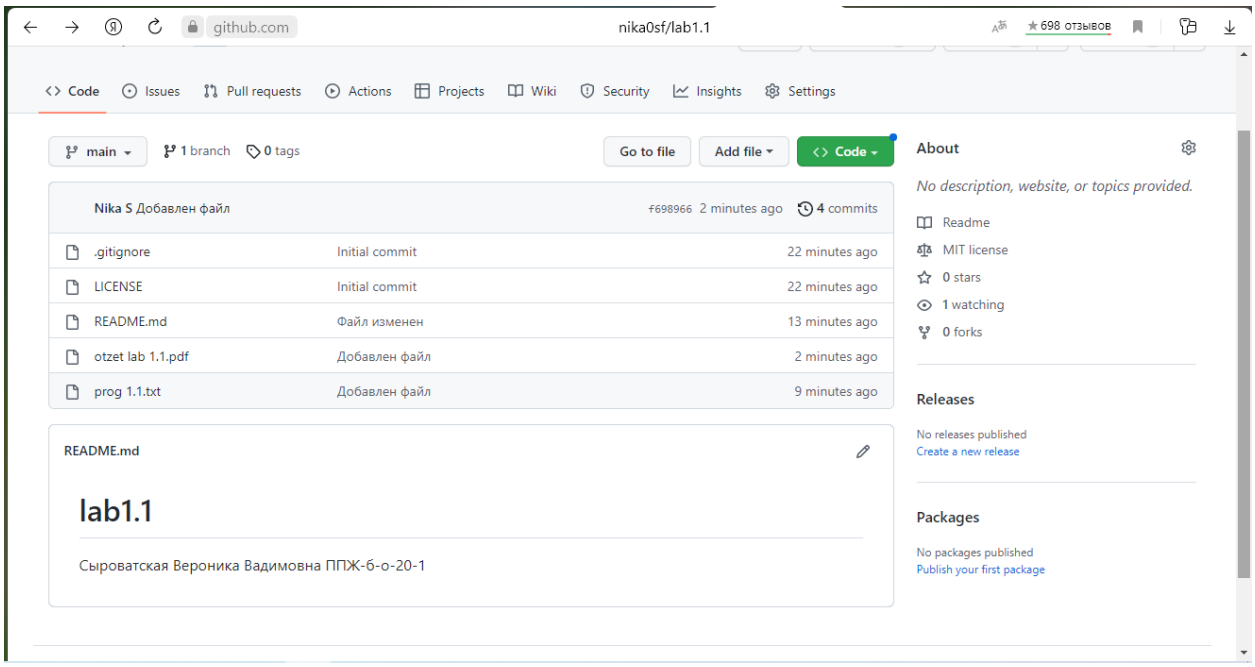
Воронкин Роман Александрович


Git CMD
C:\Users\nikas>git clone https://github.com/nika0sf/lab1.1.git
Cloning into 'lab1.1'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\nikas>git config --global user.name "Nika S"
C:\Users\nikas>global user.email "syrovatskaysv@mail.ru"
"global" не является внутренней или внешней командой, исполняемой программой или пакетным файлом.
C:\Users\nikas>git config --global user.email "syrovatskaysv@mail.ru"


Выбрать Git CMD
C:\Users\nikas>cd lab1.1
C:\Users\nikas\lab1.1>git add .
C:\Users\nikas\lab1.1>git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified: README.md
C:\Users\nikas\lab1.1>git commit -m "Файл изменен"
[main 3bb3904] Файл изменен
1 file changed, 3 insertions(+), 1 deletion(-)
C:\Users\nikas\lab1.1>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
nothing to commit, working tree clean
C:\Users\nikas\lab1.1>git add /
fatal: /: '/' is outside repository at 'C:/Users/nikas/lab1.1'
C:\Users\nikas\lab1.1>git add .
C:\Users\nikas\lab1.1>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.


Git CMD
n branch main
our branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
hanges to be committed:
(use "git restore --staged <file>..." to unstage
new file: prog 1.1.txt
:\Users\nikas\lab1.1>git commit -m "Добавлен файл
main 714622d] Добавлен файл
1 file changed, 4 insertions(+)
create mode 100644 prog 1.1.txt
:\Users\nikas\lab1.1>git add .
:\Users\nikas\lab1.1>git status
n branch main
our branch is ahead of 'origin/main' by 2 commits
(use "git push" to publish your local commits)
hanges to be committed:
(use "git restore --staged <file>..." to unstage
new file: otzet lab 1.1.pdf
:\Users\nikas\lab1.1>git commit -m "Добавлен файл
main f698966] Добавлен файл
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 otzet lab 1.1.pdf


C:\Users\nikas\lab1.1>git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 766.22 KiB | 9.70 MiB
/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused
0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/nika0sf/lab1.1.git
92a5123..f698966 main -> main
C:\Users\nikas\lab1.1>


github.com
nika0sf/lab1.1
★ 698 отзывов
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
Nika S Добавлен файл
f698966 2 minutes ago
4 commits
.gitignore Initial commit 22 minutes ago
LICENSE Initial commit 22 minutes ago
README.md Файл изменен 13 minutes ago
otzet lab 1.1.pdf Добавлен файл 2 minutes ago
prog 1.1.txt Добавлен файл 9 minutes ago
README.md
lab1.1
Сыроватская Вероника Вадимовна ППЖ-б-о-20-1
About
No description, website, or topics provided.
Readme
MIT license
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages
No packages published
Publish your first package

Вопросы для защиты работы

1. Что такое СКВ и каково ее назначение?

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

2. В чем недостатки локальных и централизованных СКВ?

Самый очевидный минус централизованных СКВ — это единая точка отказа, представленная централизованным сервером. Если этот сервер выйдет из строя на час, то в течение этого времени никто не сможет использовать контроль версий для сохранения изменений, над которыми работает, а также никто не сможет обмениваться этими изменениями с другими разработчиками. Если жёсткий диск, на котором хранится центральная БД, повреждён, а своевременные бэкапы отсутствуют, вы потеряете всё — всю историю проекта, не считая единичных снимков репозитория, которые сохранились на локальных машинах разработчиков. Локальные СКВ страдают от той же самой проблемы: когда вся история проекта хранится в одном месте, вы рискуете потерять всё.

3. К какой СКВ относится Git?

К распределённым СКВ.

4. В чем концептуальное отличие Git от других СКВ?

Основное отличие Git от любой другой СКВ (включая Subversion и её собратьев) — это подход к работе со своими данными. Концептуально, большинство других систем хранят информацию в виде списка изменений в файлах. Эти системы (CVS, Subversion, Perforce, Bazaar и т. д.) представляют хранимую информацию в виде набора файлов и изменений, сделанных в каждом файле, по времени. Git не хранит и не обрабатывает данные таким способом. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда вы делаете коммит, то есть сохраняете состояние своего проекта в Git, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Для увеличения эффективности, если файлы не были изменены, Git не запоминает эти файлы вновь, а только создаёт ссылку на предыдущую версию идентичного файла, который уже сохранён. Git представляет свои данные как, скажем, поток снимков

5. Как обеспечивается целостность хранимых данных в Git?

В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме. Это значит, что невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом. Данная функциональность встроена в Git на низком уровне и является неотъемлемой частью его философии. Вы не потеряете информацию во время её передачи и не получите повреждённый файл без ведома Git.

6. В каких состояниях могут находится файлы в Git? Как связаны эти состояния?

У Git есть три основных состояния, в которых могут находиться ваши файлы: зафиксированное (committed), изменённое (modified) и подготовленное (staged). Зафиксированный значит, что файл уже сохранён в вашей локальной базе. К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы. Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль - это ваша публичная страница на GitHub, как и в социальных сетях. Когда вы ищете работу в качестве программиста, работодатели могут посмотреть ваш профиль GitHub и принять его во внимание, когда будут решать, брать вас на работу или нет.

8. Какие бывают репозитории в GitHub?

Public/private. (ОТКРЫТЫЙ/ЗАКРЫТЫЙ)

9. Укажите основные этапы модели работы с GitHub.
Сначала рассмотрим область GitHub. В нем есть два хранилища: upstream - это оригинальный репозиторий проекта, который вы скопировали, origin - ваш fork (копия) на GitHub, к которому у вас есть полный доступ. Чтобы перенести изменения с вашей копии в исходному репозиторий проекта, вам нужно сделать запрос на извлечение. Если вы хотите внести небольшие изменения в свою копию (fork), вы можете использовать вебинтерфейс GitHub. Однако такой подход не удобен при разработке программ, поскольку вам часто приходится запускать и отлаживать их локально. Стандартный способ - создать локальный клон удаленного репозитория и работать с ним локально, периодически внося изменения в удаленный репозиторий.

10. Как осуществляется первоначальная настройка Git после установки?

Чтобы убедиться, что Git был успешно установлен, введите команду ниже в терминале, чтобы отобразить текущую версию вашего Git: git version

11. Опишите этапы создания репозитория в GitHub.

Имя репозитория. Оно может быть любое, необязательно уникальное во всем github, потому что привязано к вашему аккаунту, но уникальное в рамках тех репозиториев, которые вы создавали. Описание (Description). Можно оставить пустым. Public/private. Выбираем открытый (Public), НЕ ставим галочку “Initialize this repository with a README” (В README потом будет лежать какая-то основная информация, что же такое ваш проект и как с ним работать). .gitignore и LICENSE можно сейчас не выбирать.

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

После создания репозитория его необходимо клонировать на ваш компьютер. Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования

14. Как проверить состояние локального репозитория Git?

Перейдите в каталог хранилища и посмотрите на содержимое. Локальный репозиторий включает в себя все те же файлы, ветки и историю коммитов, как и удаленный репозиторий. Введите эту команду, чтобы проверить состояние вашего репозитория: git status

15. Как изменяется состояние локального репозитория Git после выполнения следующих операций: добавления/изменения файла в локальный репозиторий Git; добавления нового/ измененного файла под версионный контроль с помощью команды git add ; фиксации (коммита) изменений с помощью команды git commit и отправки изменений на сервер с помощью команды git push ?

16. У Вас имеется репозиторий на GitHub и два рабочих компьютера, с помощью которых Вы можете осуществлять работу над некоторым проектом с использованием этого репозитория. Опишите последовательность команд, с помощью которых оба локальных репозитория,

связанных с репозиторием GitHub будут находиться в синхронизированном состоянии. Примечание: описание необходимо начать с команды git clone .

17. GitHub является не единственным сервисом, работающим с Git. Какие сервисы еще Вам известны? Приведите сравнительный анализ одного из таких сервисов с GitHub.

18. Интерфейс командной строки является не единственным и далеко не самым удобным способом работы с Git. Какие Вам известны программные средства с графическим интерфейсом пользователя для работы с Git? Приведите как реализуются описанные в лабораторной работе операции Git с помощью одного из таких программных средств.