МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОССУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРИТЕТ

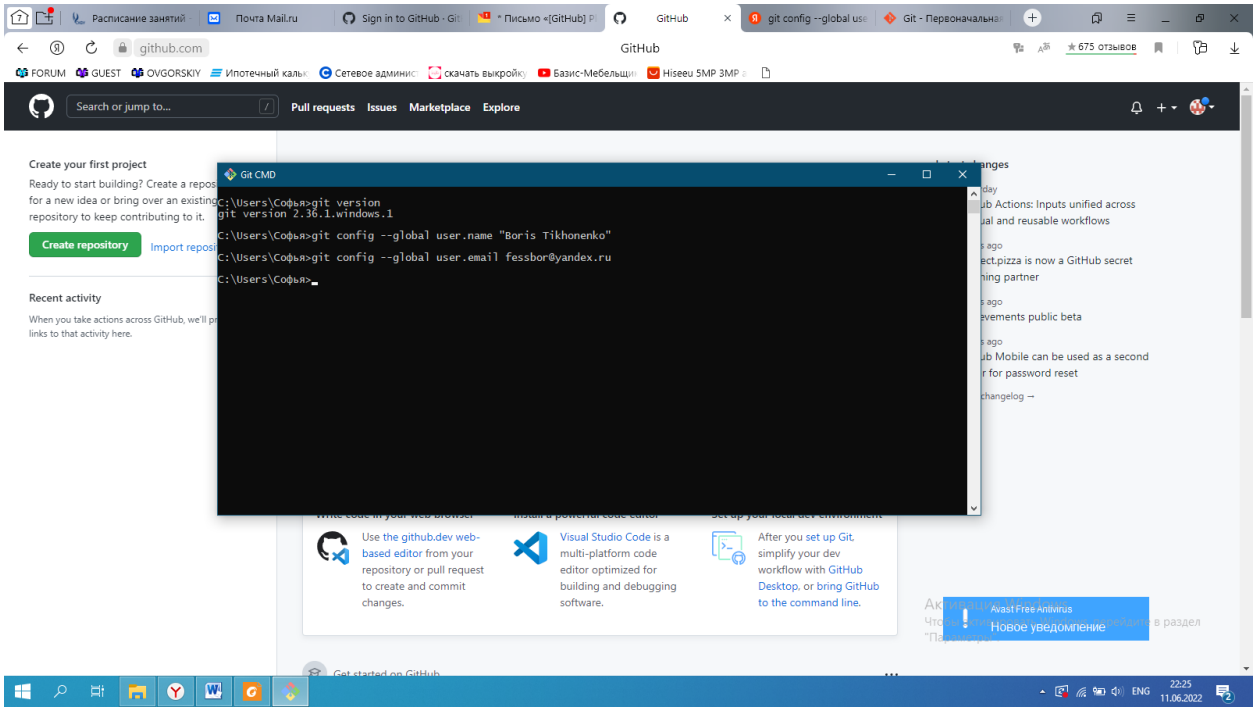
Лабораторная работа № 1.
«Исследование основных возможностей Git и GitHub»

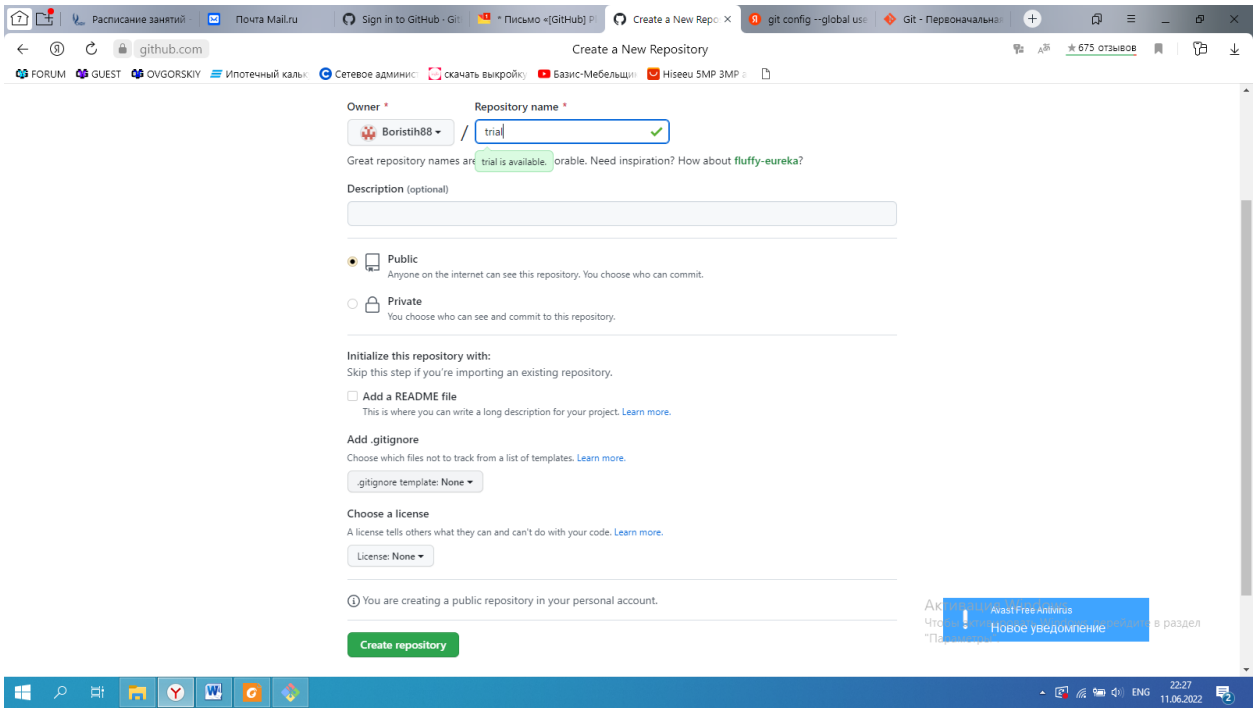
Выполнил
Тихоненко Борис Витальевич
1 курс, группа ИТС-б-з-21-1,
11.03.02
«Инфокоммуникационные
системы и сети», заочная
форма обучения.
«Основы
кроссплатформенного
программирования»
Преподаватель: Воронкин
Роман Александрович

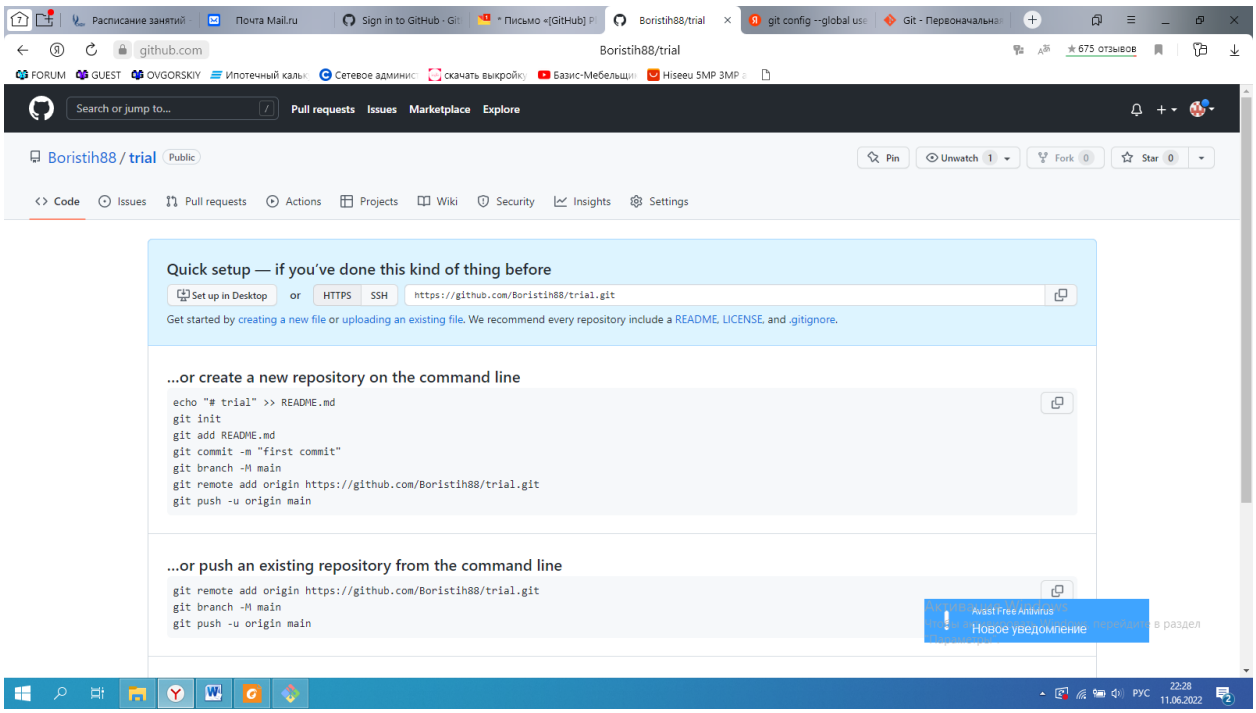
Ставрополь, 2022 г.

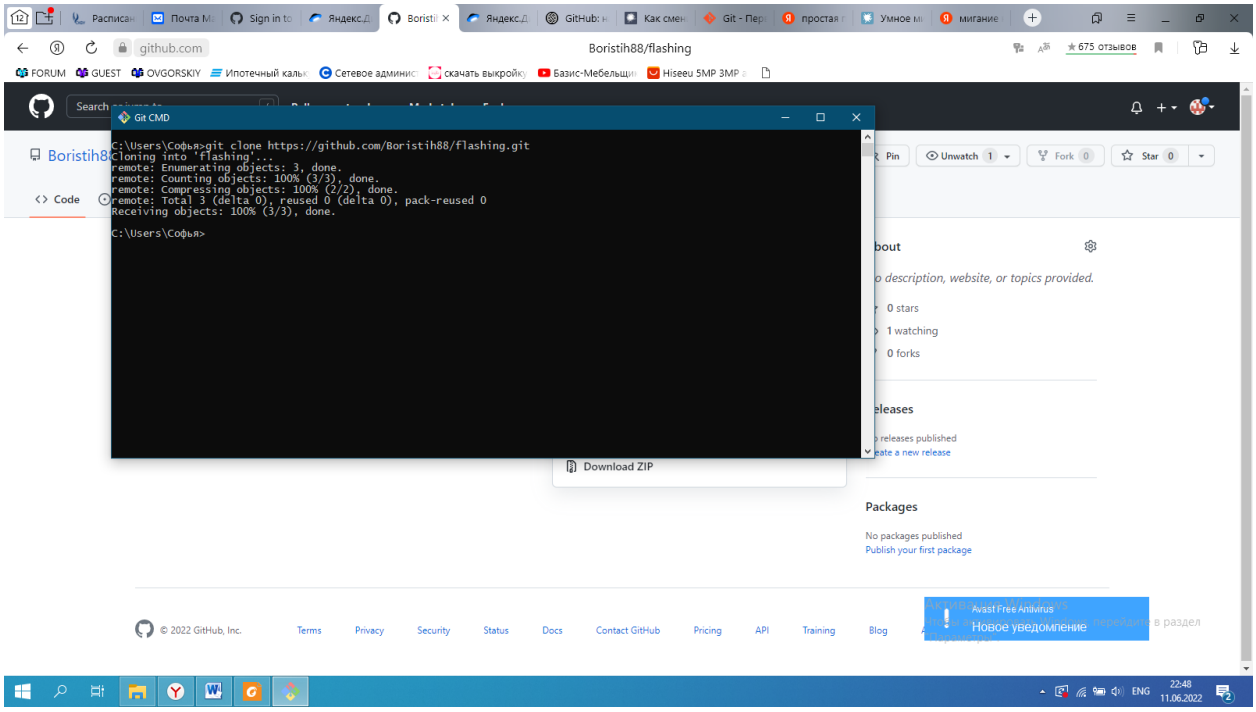
Цель работы: исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

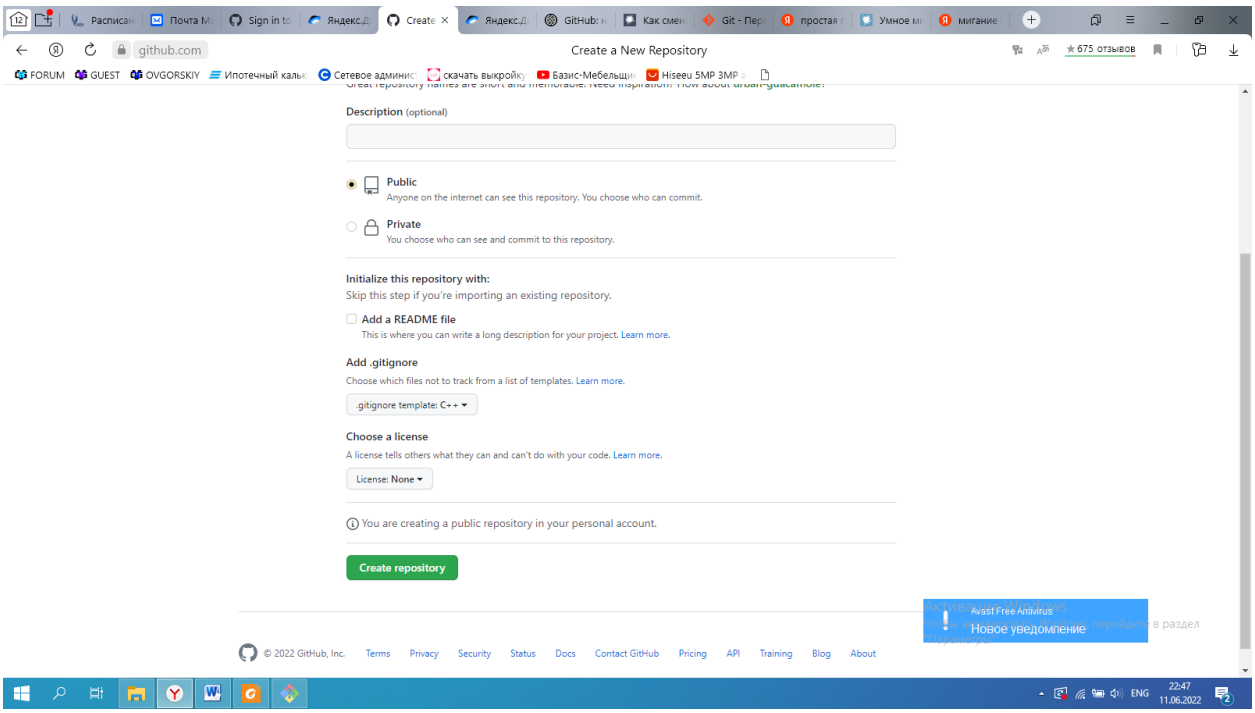
Согласно методике и порядку выполнения работ, были сделаны следующие скриншоты, подтверждающие выполнение лабораторной работы.

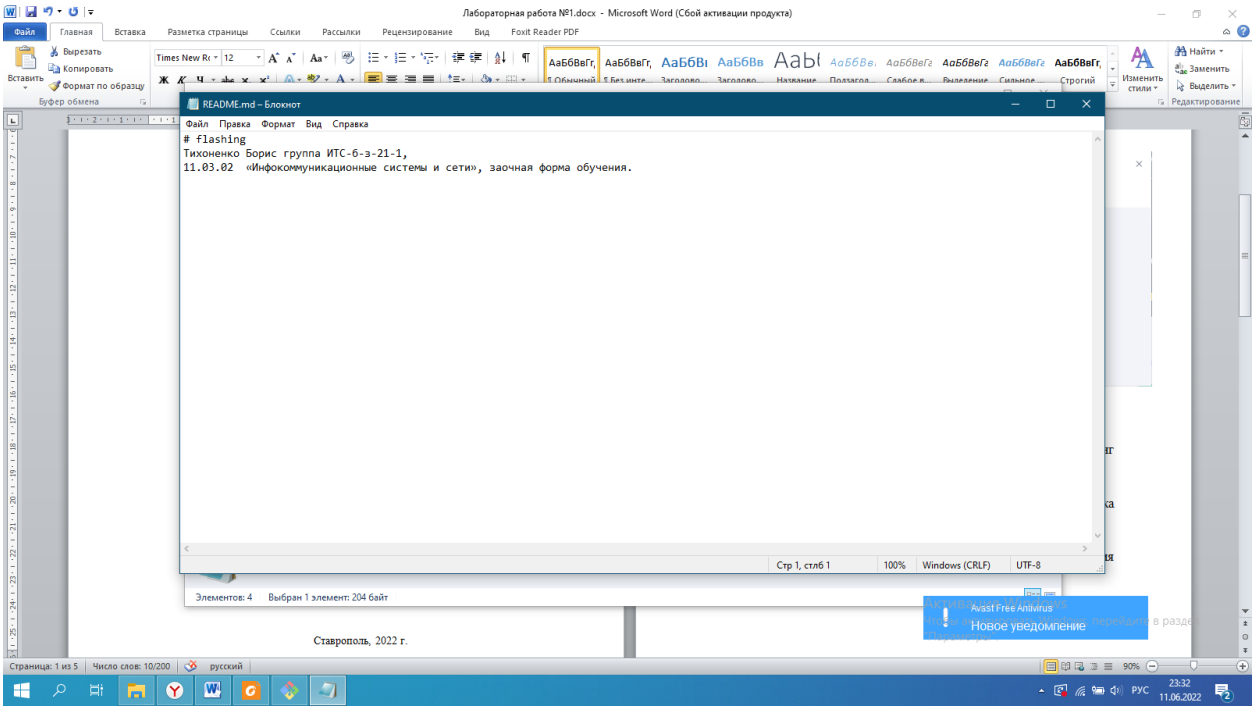


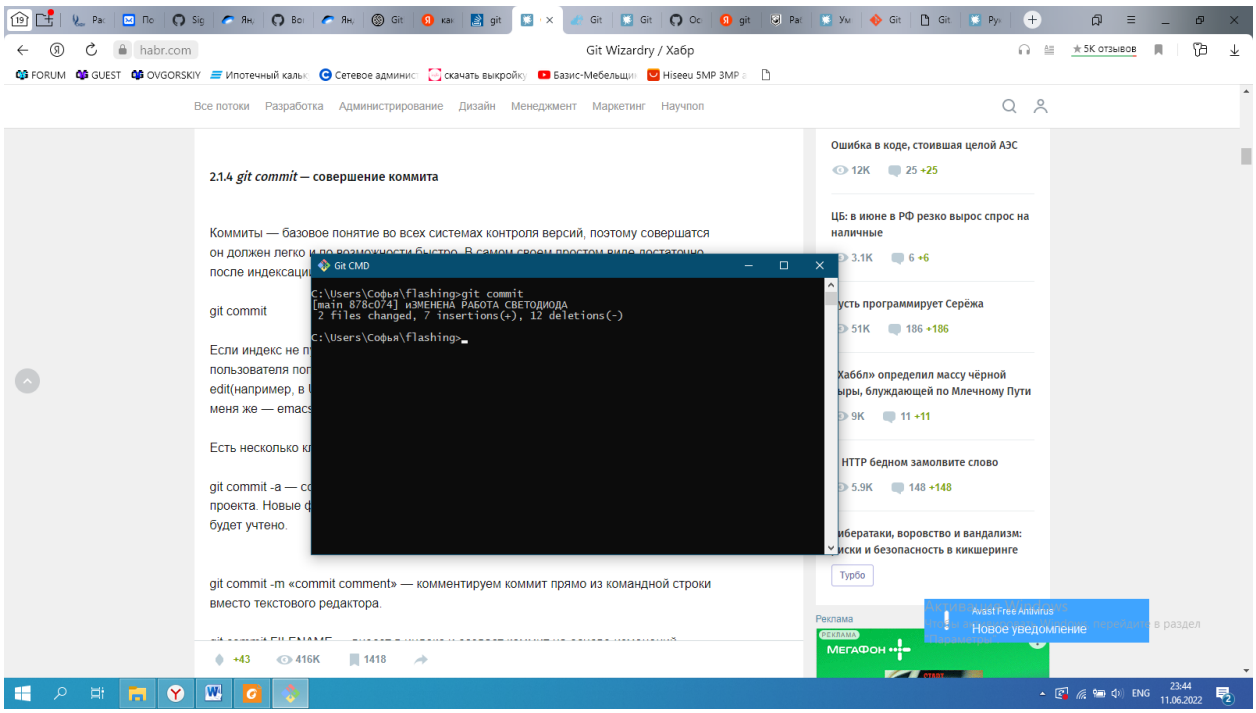


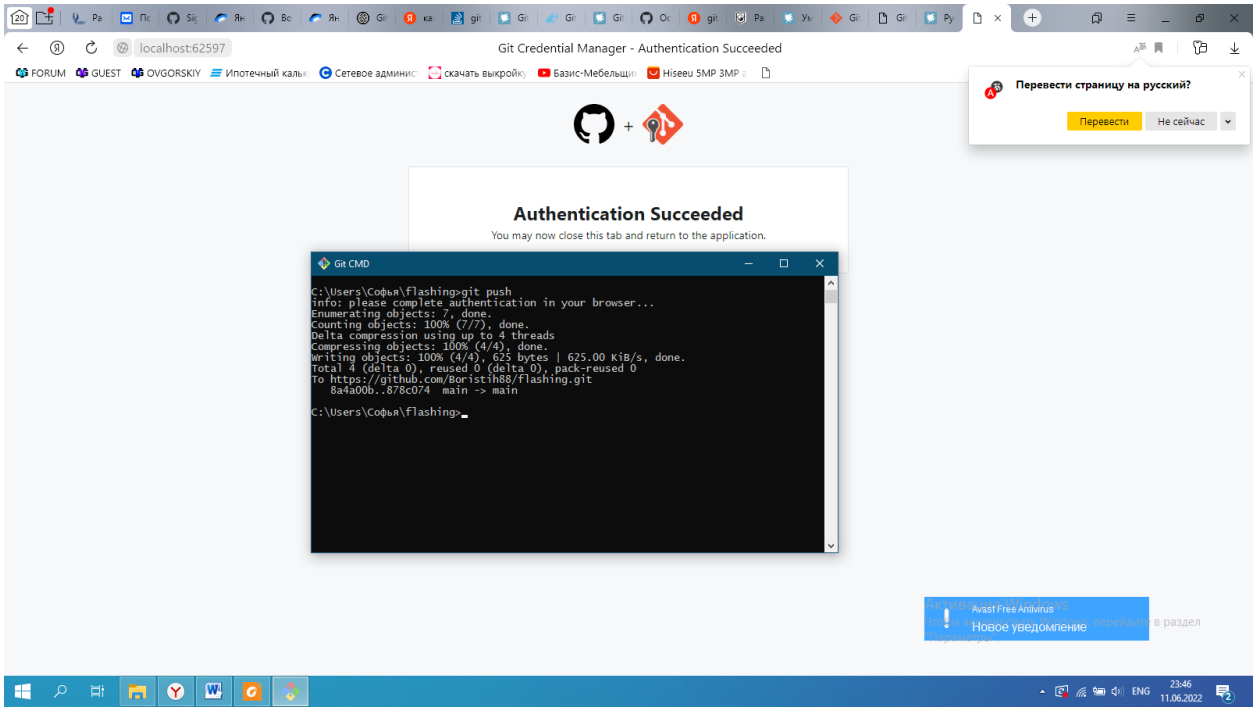

Boristih88/trial
Search or jump to...
Pull requests
Issues
Marketplace
Explore
Boristih88 / trial
Public
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/Boristih88/trial.git
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# trial" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Boristih88/trial.git
git push -u origin main
...or push an existing repository from the command line
git remote add origin https://github.com/Boristih88/trial.git
git branch -M main
git push -u origin main
Avast Free Antivirus
Новое уведомление


Boristih88/flashing
Git CMD
C:\Users\Софья>git clone https://github.com/Boristih88/flashing.git
Cloning into 'flashing'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
C:\Users\Софья>
Code
Download ZIP
Packages
No packages published
Publish your first package
Avast Free Antivirus
Новое уведомление


Create a New Repository
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Skip this step if you're importing an existing repository.
Add a README file
This is where you can write a long description for your project. Learn more.
Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: C++
Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None
You are creating a public repository in your personal account.
Create repository
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About
Avast Free Antivirus
Новое уведомление


Лабораторная работа №1.docx - Microsoft Word (Сбой активации продукта)
README.md – Блокнот
Файл Правка Формат Вид Справка
flashing
Тихоненко Борис группа ИТС-б-з-21-1,
11.03.02 «Инфокоммуникационные системы и сети», заочная форма обучения.
Стр 1, стлб 1
100%
Windows (CRLF)
UTF-8
Элементов: 4
Выбран 1 элемент: 204 байт
Ставрополь, 2022 г.
Avast Free Antivirus
Новое уведомление





Ответы на контрольные вопросы:

1. 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. Недостаток локальной системы контроля версий заключается в невозможности проверить изменения другими пользователями, не подключенными непосредственно к рабочей станции исполнителя. Недостаток же централизованной системы контроля версий

заключается в единой точке отказа, представленной 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.

3. GIT относится к распределенным системам контроля версий.
4. Основное отличие Git от любой другой СКВ (включая Subversion и её собратьев) — это подход к работе со своими данными. 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.
5. 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.
6. 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. Профиль - это ваша публичная страница на GitHub, как и в социальных сетях. В нем размещается вся информация о проектах, кодах, активности.
8. Каждый проект размещается в своем собственном контейнере, который называется репозиторием. В нем можно хранить код, конфигурации, наборы данных, изображения и другие файлы, включенные в ваш проект. Любые изменения файлов в репозитории будут отслеживаться с помощью контроля версий. Соответственно, видов репозиториев может быть огромное количество, в зависимости от людей, которые их создают.

9. Установить программу git на вашей системе. Настроить программу и проверить её работоспособность локально. Зарегистрировать ваш аккаунт на GitHub. Создать локальный репозиторий или копировать репозиторий существующего проекта. Написать файл README.MD. В случае, если вы начинаете проект, создать удаленный репозиторий. Фиксировать изменения локально. Отправлять изменения на GitHub. Зарегистрировать аккаунты разработчиков вашего проекта. Выдать им ссылку на проект.
10. Чтобы убедиться, что Git был успешно установлен, введите команду ниже в терминале, чтобы отобразить текущую версию вашего Git:

git version

Если она сработала, необходимо добавить в настройки Git ваше имя, фамилию и адрес электронной почты, связанный с вашей учетной записью GitHub:

git config --global user.name <YOUR_NAME>

git config --global user.email <EMAIL>

11. В правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория (рис 1.12).В результате будет выполнен переход на страницу создания репозитория. Наиболее важными наней являются следующие поля: 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), .gitignore и LICENSE. После заполнения этих полей нажимаем кнопку Create repository. Отлично, ваш репозиторий готов!
12. MIT, BSD, GPL, APACHE, LGPL, ECLIPSE, AFFERO.
13. 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 (рис. 1.13). Откройте командную строку или терминал и перейдите в каталог, куда вы хотите скопировать хранилище. Затем напишите git clone и введите скопированный адрес. Копировать репозиторий требуется для того, чтобы была возможность работать с ним локально. А потом уже вносить изменения в удаленный репозиторий, не мешая другим пользователям.

14. Командой git status.
15. Локальный репозиторий в процессе выполнения данных команд не изменяется.
16. После того, как репозиторий был создан на Github, его можно скопировать на любой другой компьютер. Для этого применяется команда:

git clone https://github.com/myuser/project.git

Результатом выполнения этой команды будет создание папки project в текущем каталоге. Эта папка также будет содержать локальный репозиторий (то есть папку .git). Так же можно добавить название папки, в которой вы хотите разместить локальный репозиторий.

git clone https://github.com/myuser/project.git

С одним репозиторием с разных компьютеров может работать несколько разработчиков или вы сами, если например работаете над одним и тем же проектом дома и на работе. Для получения обновлений с удаленного репозитория воспользуйтесь командой:

git pull

Если вы изменили ваши локальные файлы, то команда git pull выдаст ошибку. Если вы уверены, что хотите перезаписать локальные файлы, файлами из удаленного репозитория то выполните команды:

git fetch --all

git reset --hard github/master

Вместо github подставьте название вашего удаленного репозитория, которое вы зарегистрировали командой git push -u.

Как мы уже знаем, для того чтобы изменения выложить на удаленный репозиторий используется команда:

git push

В случае, если в удаленном репозитории лежат файлы с версией более новой, чем у вас в локальном, то команда git push выдаст ошибку. Если вы уверены, что хотите перезаписать файлы в удаленном репозитории несмотря на конфликт версий, то воспользуйтесь командой:

git push –f

17. Gitlab, Bitbucket, Gogs, SourceForge. Остановимся на Gitlab. Среди преимуществ Gitlab — «чистый» и близкий к GitHub интерфейс,

собственная wiki-система, система отслеживания ошибок, возможность назначить пользователям роли в проекте, встроенные возможности непрерывной интеграции и развёртывания (CI/CD), неограниченное число личных проектов по бесплатной подписке. Платформа также предлагает встроенные инструменты для постановки и отслеживания задач и интеграцию с Google Cloud для развёртывания проектов. Кроме того, Gitlab проект с частично открытым исходным кодом, и пользователи могут самостоятельно собрать собственную версию инструмента. Разница между GitLab и GitHub. GitHub делает упор на высокую доступность и производительность своей инфраструктуры и делегирует другие сложные функции сторонним инструментам. GitLab, наоборот, фокусируется на включении всех функций на одной проверенной и хорошо интегрированной платформе; он обеспечивает все для полного жизненного цикла DevOps под одной крышей. Что касается популярности, GitHub определенно превосходит GitLab. В GitLab меньше разработчиков внедряют на платформу открытые исходные коды. Кроме того, что касается цен, GitHub стоит дороже, что делает его неподходящим для пользователей с небольшим бюджетом.

18. Рассмотрим Git клиент Tower. После его установки, в открышемся окне клиента по центру размещена табличка, предлагающая нам добавить репозиторий или же клонировать из Github. При нажатии на кнопку clone, необходимо вставить в поле URL нашего удаленного репозитория и нажать кнопку clone. После этого в панеле задач, расположенной слева, появится папка с названием нашего репозитория. Дальнейшая работа с ним не представляет особого труда, так как все CMD команды здесь представлены в виде графически представленных кнопок или же содержаться в выпадающих меню.

Вывод: Выполняя данную лабораторную работу я исследовал базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.