

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования**
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №1

Выполнила студентка группы ИТС-б-о-
20-1 (2)

Маслова А.В. « » _____ 2021г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент
Кафедры инфокоммуникаций
Воронкин Р.А.

(подпись)

г. Ставрополь, 2021

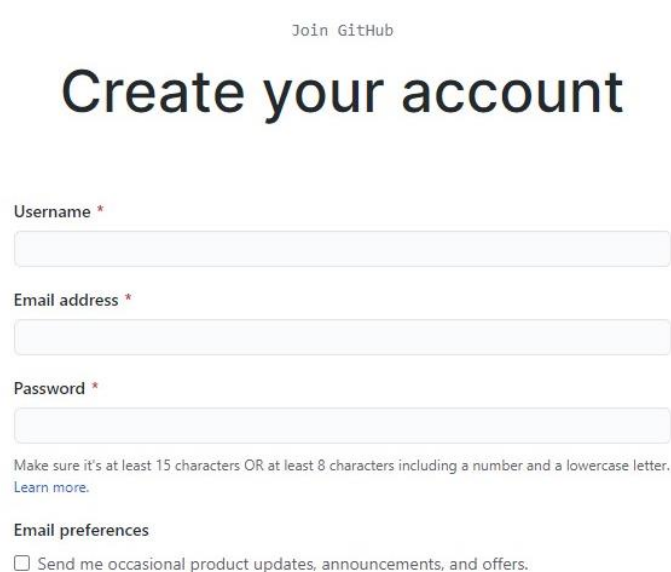
Лабораторная работа №1

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

Ход работы:

1. Создание аккаунта на GitHub.

1) регистрация на сайте GitHub;

The image shows the GitHub registration page. At the top, there is a link "Join GitHub". Below it, the heading "Create your account" is displayed in a large, bold font. The registration form consists of three input fields: "Username *", "Email address *", and "Password *". Below the password field, there is a note: "Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter." followed by a link "Learn more.". At the bottom of the form, there is a section titled "Email preferences" with a checkbox labeled "Send me occasional product updates, announcements, and offers.".

Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.
[Learn more.](#)

Email preferences

☐ Send me occasional product updates, announcements, and offers.

Рисунок 1 – регистрация на GitHub.

2) создание репозитория;

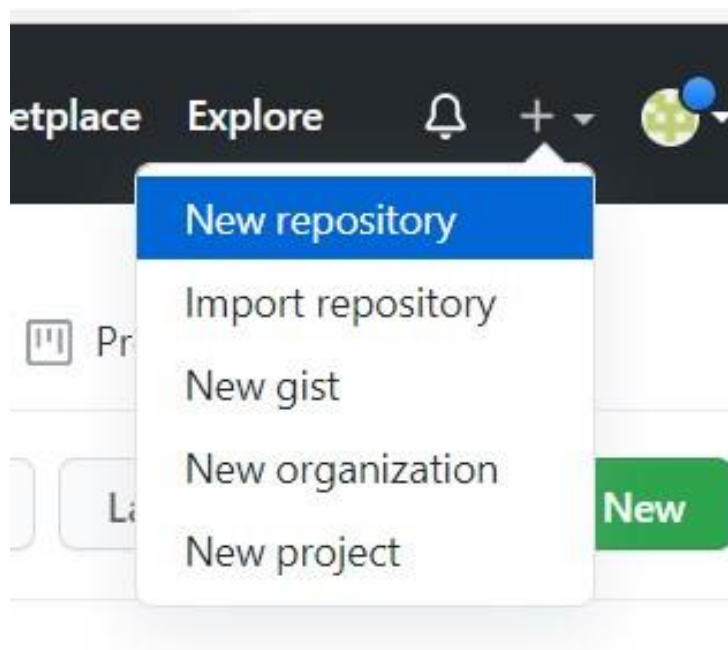


Рисунок 2 – создание нового репозитория.

2. Выполнение клонирования репозитория на рабочий компьютер:

- для этого скачаем и установим Git;

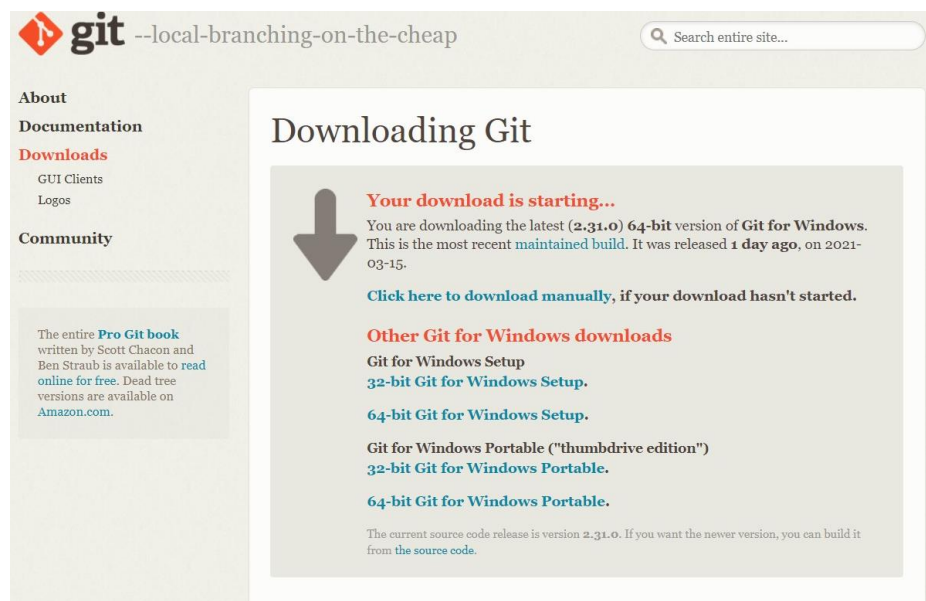


Рисунок 3 – скачивание Git

3. Клонирование репозитория на свою компьютер:

```
D:\git1>git clone https://github.com/anya0230/anya.git
Cloning into 'an'
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.

D:\git1>
```

Рисунок 4 – Выполненные команды для клонирования репозитория.

4. Записать небольшую программу на выбранном языке программирования.

1) создание тестового файла;

Имя	Дата изменения	Тип
 .gitignore	17.03.2021 20:21	Текстовый файл
 test	17.03.2021 20:24	Python File

Рисунок 5 – создание файла test

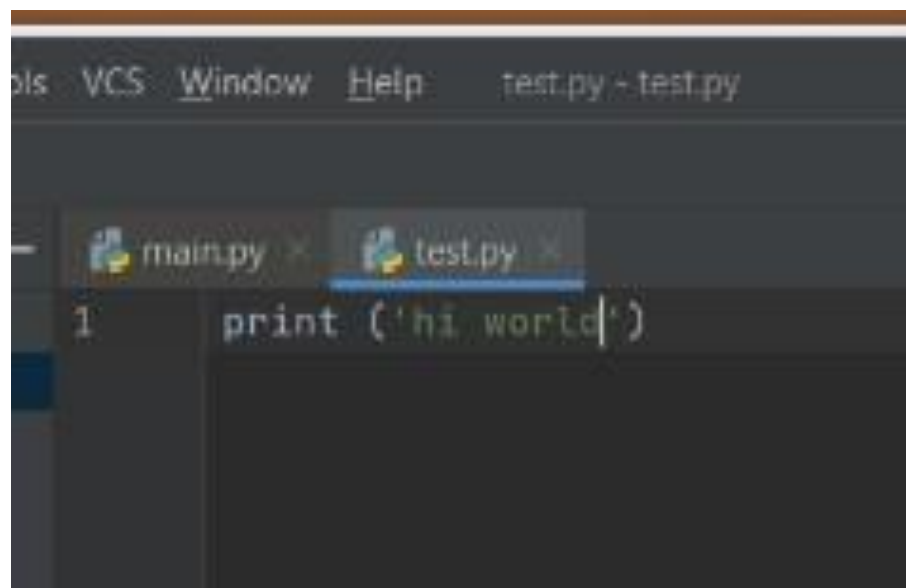


Рисунок 6 – запись программы в Python

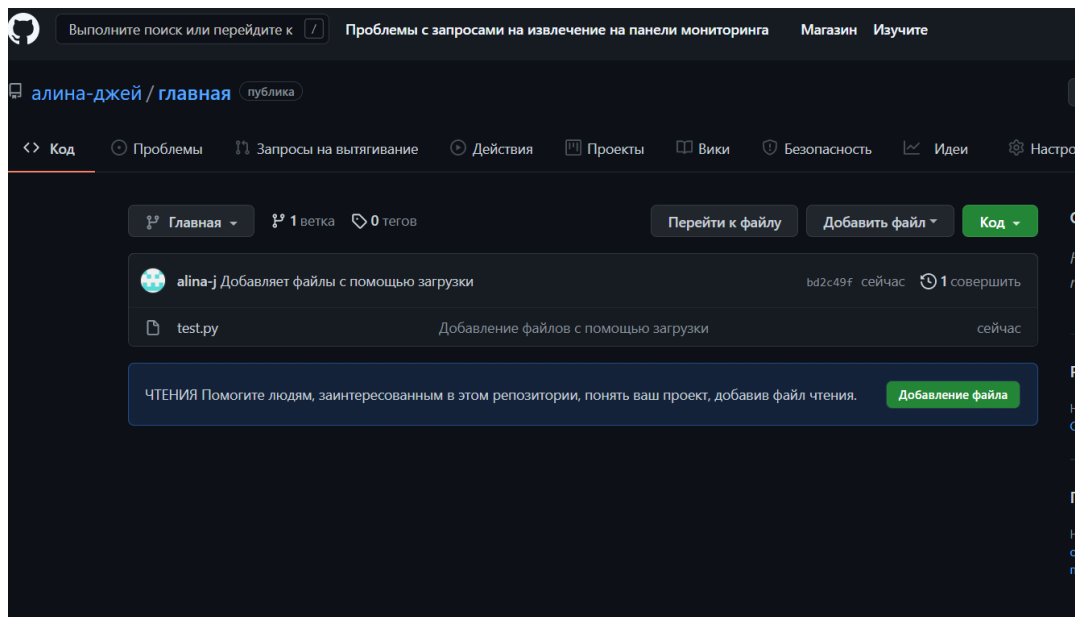


Рисунок 7 – добавление файла test.py

Ответы на контрольные вопросы:

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.

СКВ даёт возможность возвращать отдельные файлы к прежнему виду, возвращать к прежнему состоянию весь проект, просматривать происходящие со временем изменения, определять, кто последним вносил изменения во внезапно переставший работать модуль, кто и когда внёс в код какую-то ошибку, и многое другое.

2. В чем недостатки локальных и централизованных СКВ?

В локальных: сильно подвержен появлению ошибок. Можно легко забыть, в какой директории вы находитесь, и случайно изменить не тот файл или скопировать не те файлы, которые вы хотели.

В централизованных: 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?

относится к распределённым системам контроля версий (РСКВ).

4. В чём концептуальное отличие Git от других СКВ?

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.

В Git переосмысливает практически все аспекты контроля версий, которые были скопированы из предыдущего поколения большинством других систем. Это делает Git больше похожим на миниатюрную файловую систему с удивительно мощными утилитами, настроенными над ней, нежели просто на СКВ

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.

6. В каких состояниях могут находиться 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.

Если определённая версия файла есть в Git-директории, эта версия считается зафиксированной. Если версия файла изменена и добавлена в индекс, значит, она подготовлена. И если файл был изменён с момента последнего распаковывания из репозитория, но не был добавлен в индекс, он считается изменённым

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?

репозиторий бывает разветвленный и оригинальный, локальный и удаленный.

Локальный репозиторий имеет полную историю версий проекта, которая может быть полезна при разработке без подключения к интернету. После того, как вы что-то изменили в локальном, вы можете отправить свои изменения в удаленный репозиторий, чтобы сделать их видимыми для других разработчиков.

9. Укажите основные этапы модели работы с GitHub.

Этапы модели работы с GitHub:

- регистрация;
- создание нового репозитория; – клонирование репозитория.

10. Как осуществляется первоначальная настройка Git после установки?

- для начала необходимо убедиться, что Git установлен: команда «git version»;
- при помощи командной строки вводим « cd/d<файл> переходим в папку с локальным репозиторием;
- для того, чтобы проверить, что команда сработала, добавим в настройки Git имя и фамилию и адрес электронной почты, связанный с нашей учетной записью GitHub(команды: `git config --global user.name <YOUR_NAME` `git config --global user.email <EMAIL>`)

11. Опишите этапы создания репозитория в GitHub.

Этапы создания репозитория в GitHub:

- в правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория;
- в открывшемся окне даем имя репозиторию, оставляя пустым «описание»;
- выбираем открытый , не ставим галочку “Initialize this repository with a README”;
- после заполнения полей нажимаем на кнопку Create repository.

12.Какие типы лицензий поддерживаются GitHub при создании репозитория?

Apache, GPL, MIT

13.Как осуществляется клонирование репозиторияGitHub?

Зачем нужно клонировать репозиторий?

- на странице репозитория необходимо найти кнопку Clone или Code;
- щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования;
- открываем командную строку или терминал и перейдите в каталог, куда вы хотите скопировать хранилище;
- записываем `git clone` и вводим адрес.

14. Как проверить состояние локального репозитория Git?

для того чтобы проверить состояние репозитория, необходимо ввести в командную строку команду `git status`.

15. Как изменяется состояние локального репозиторияGit после выполнения следующих операций: добавления/изменения файла в локальный репозиторийGit; 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` ?

файл будет обновлен на удаленном репозитории в GitHub.

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 репозиториемGitHub будут находиться в синхронизированном состоянии. Примечание: описание необходимо начать с команды `git clone`.

- для начала необходимо клонировать репозиторий на наш компьютер, введя в командную строку «`git clone`»;

– чтобы сделать синхронизацию изменений, необходимо ввести в командную строку команду «git pull»

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

1) GitLab– предоставляет не только веб-сервис для совместной работы, но и ПО с открытым исходным кодом.

2) BitBucket — это служба хостинга репозитория и управления версиями от Atlassian. Сервис бесплатен для проектов, в команде которых не больше пяти участников

Сравнение GitLab и GitHub:

GitLab в отличие от GitHub предлагает пользователям бесплатные репозитории для проектов. Также GitLab имеет собственный модуль для мониторинга, анализа времени и планирования.

Сходства: GitLab и GitHub предлагают пользователям расширенные возможности для отслеживания проблем (изменять статус, назначать ответственных, классификация при помощи меток). У обоих ресурсов сформировалось большое сообщество разработчиков, которые помогут начинающим в решении их проблем. Для описания проблем или запросов разработчикам доступна коллекция шаблонов, помогающая уменьшить время на заполнение. Для каждого проекта разработчики могут создать отдельную документацию. Оба ресурса позволяют сторонним разработчикам и пользователям вносить предложения и совместно работать над проектом.

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.

1) Fork – это продвинутый GUI-клиент для macOS и Windows. Преимущество его состоит в скорости, дружелюбности к пользователю и эффективности.

Особенности:

- красивый вид;
- кнопки быстрого доступа;
- встроенная система разрешения конфликтов слияния; – менеджер репозитория и уведомления GitHub.

2) GitHub Desktop – бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub, а также с другими платформами (включая Bitbucket и GitLab). Приложение просматривать различия в версиях изображений и блоков кода. При этом элементы для дальнейшего управления можно добавлять в приложение даже путем перетаскивания.

3) SmartGit – это Git-клиент для Mac, Linux и Windows. Имеет богатый функционал. Поддерживает пул-реквесты для SVN, GitHub и Bitbucket. В нем можно найти CLI для Git, графическое отображение слияний и истории коммитов, SSH-клиент, Git-Flow, программу для разрешения конфликтов слияния.

Вывод: в ходе выполнения данной лабораторной работы были исследованы базовые возможности системы контроля версий Git и веб-сервиса для хостинга Итпоектов GitHub, а также было проведено ознакомление с регистрацией репозитория и новыми командами.