**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт цифрового развития**

**Кафедра информационных систем и технологий**

Отчет по лабораторной работе №1.

Дисциплина: **«Основы программной инженерии»**

**Выполнил:**

Студент группы ПИЖ-б-о-22-1,
направление подготовки: 09.03.04
«Программная инженерия»

ФИО: Джараян Арег Александрович

**Проверил:**

Воронкин Роман Александрович

Доцент кафедры инфокомуникаций

Ставрополь 2023

Лабораторная работа №1

Исследование основных возможностей Git и GitHub

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

Выполнение работы:

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный язык программирования (в моем случае это Python).

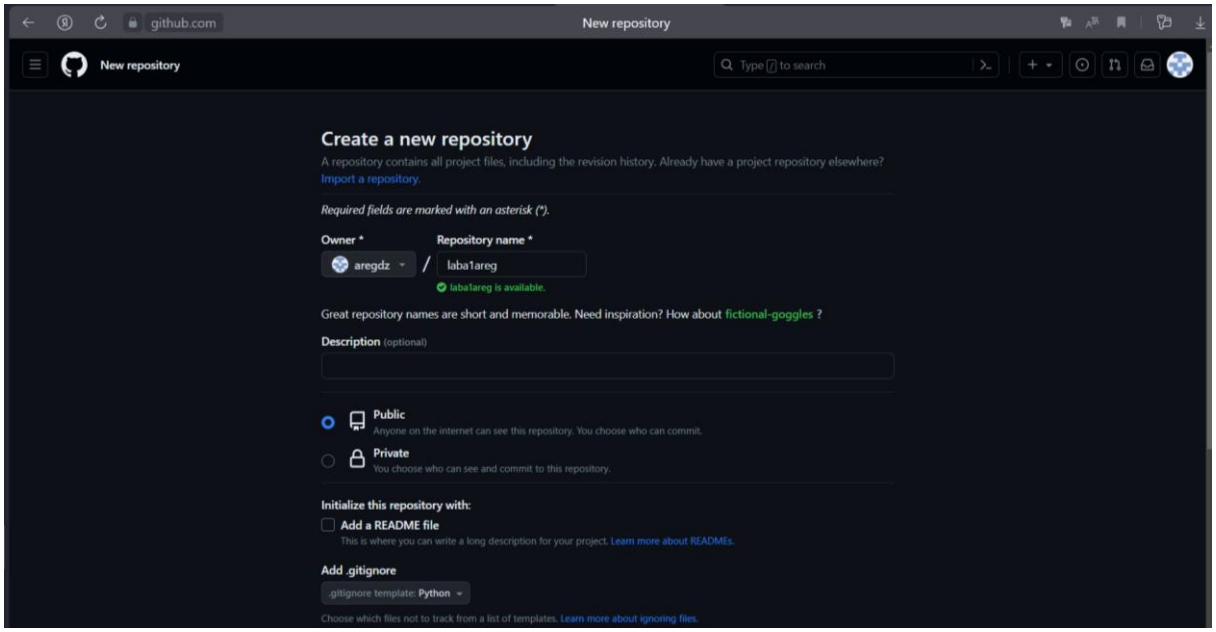


Рисунок 1.1 – Создание репозитория

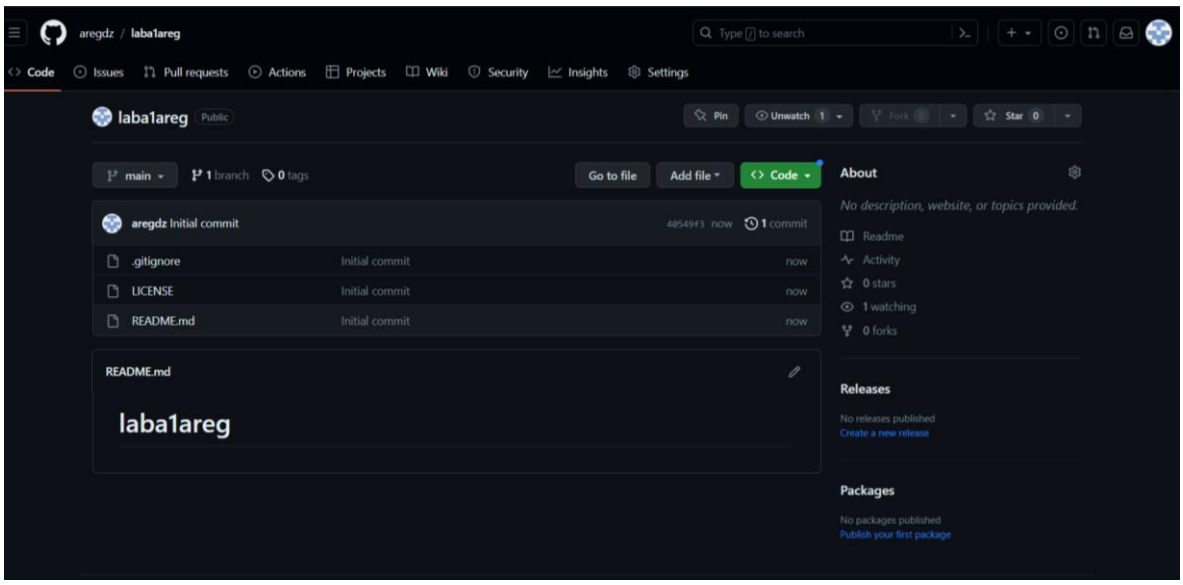


Рисунок 1.2 - Репозиторий

3. Добавил фамилию, имя и почту. Выполнил клонирование созданного репозитория на рабочий компьютер. Проверил состояние репозитория.

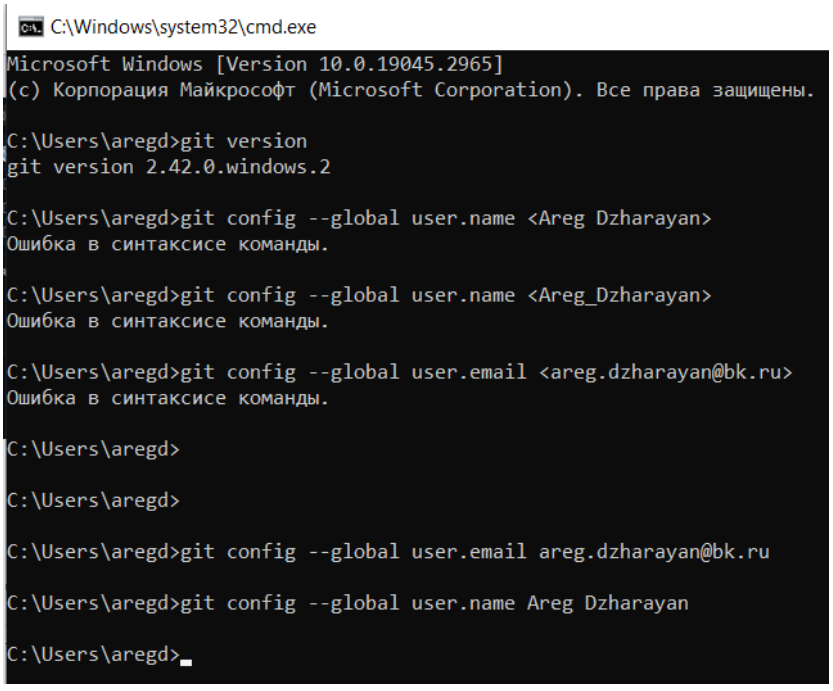


Рисунок 1.3 – Добавление ФИО И email

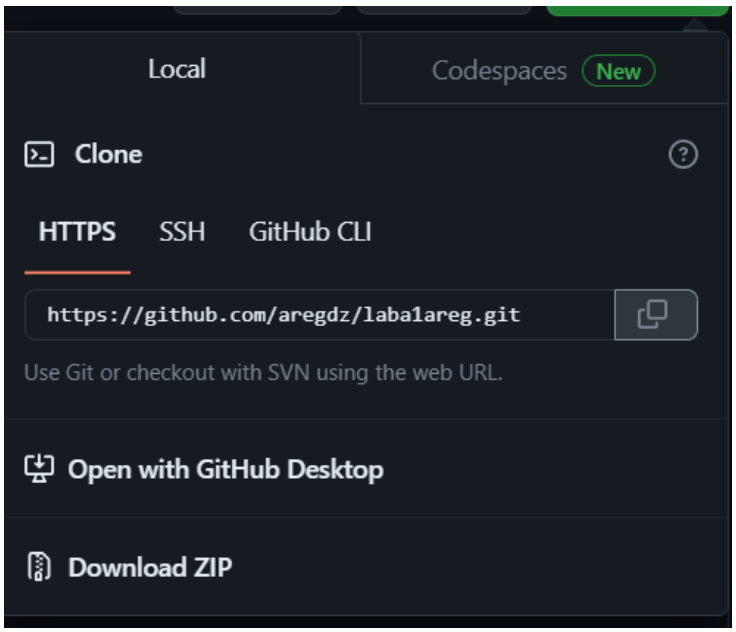


Рисунок 1.4 – Копирование ссылки на репозиторий

```
C:\Users\aregd\OneDrive\Рабочий стол\git>git clone https://github.com/aregdz/laba1areg.git
Cloning into 'laba1areg'...
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\aregd\OneDrive\Рабочий стол\git>
```

Рисунок 1.4 – Клонирование репозитория

```
C:\Users\aregd\OneDrive\Рабочий стол\git\laba1areg>git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

C:\Users\aregd\OneDrive\Рабочий стол\git\laba1areg>
```

Рисунок 1.5 – Проверка статуса

4. Дополнил файл. gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.


Рисунок 1.6 - Дополнение файла .gitignore

5. Добавил в файл README.md информацию о группе и ФИО студента, выполняющего лабораторную работу.

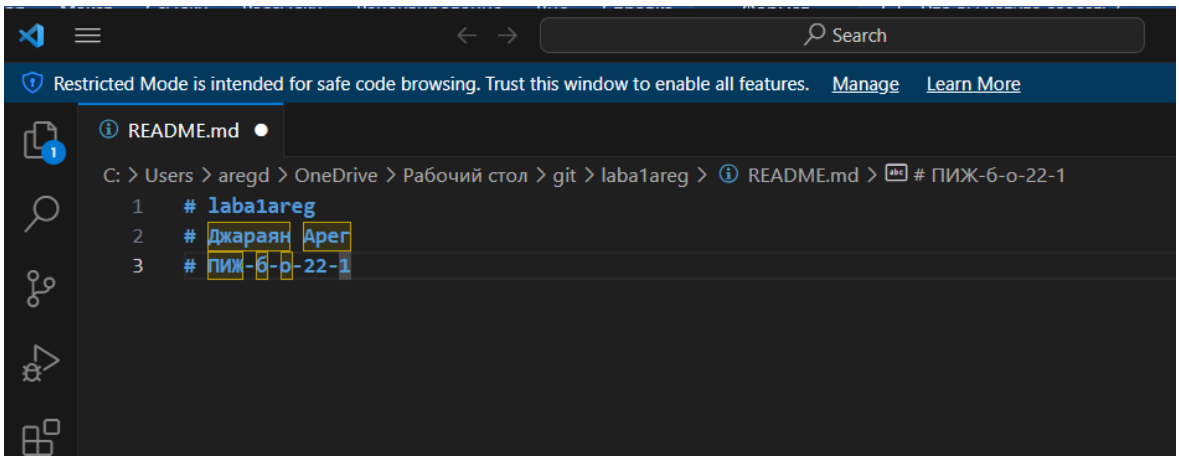


Рисунок 1.7 – Файл README.md

```
nothing to commit, working tree clean

C:\Users\aregd\OneDrive\Рабочий стол\git\laba1areg>git add .

C:\Users\aregd\OneDrive\Рабочий стол\git\laba1areg>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\aregd\OneDrive\Рабочий стол\git\laba1areg>git push
Everything up-to-date

C:\Users\aregd\OneDrive\Рабочий стол\git\laba1areg>git commit -m "change"
[main 02dc6d9] change
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\aregd\OneDrive\Рабочий стол\git\laba1areg>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/aregdz/laba1areg.git
   40549f3..02dc6d9  main -> main

C:\Users\aregd\OneDrive\Рабочий стол\git\laba1areg>
```

Рисунок 1.8 – Фиксирование изменений

6. Написал небольшую программу на выбранном языке программирования. Добавил 7 коммитов.

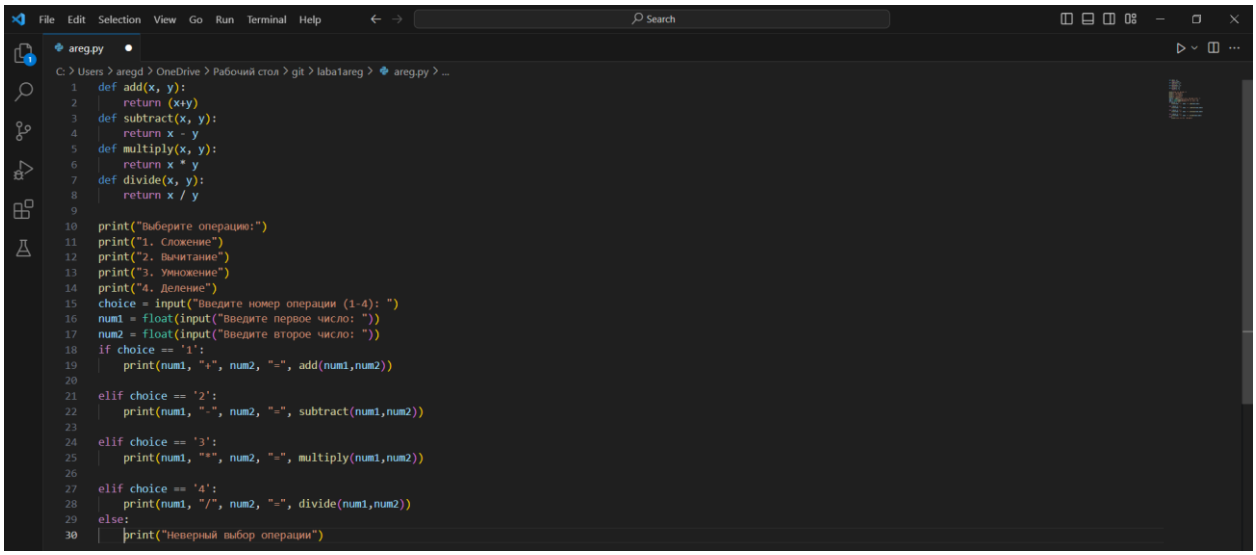


Рисунок 1.9 – Код программы

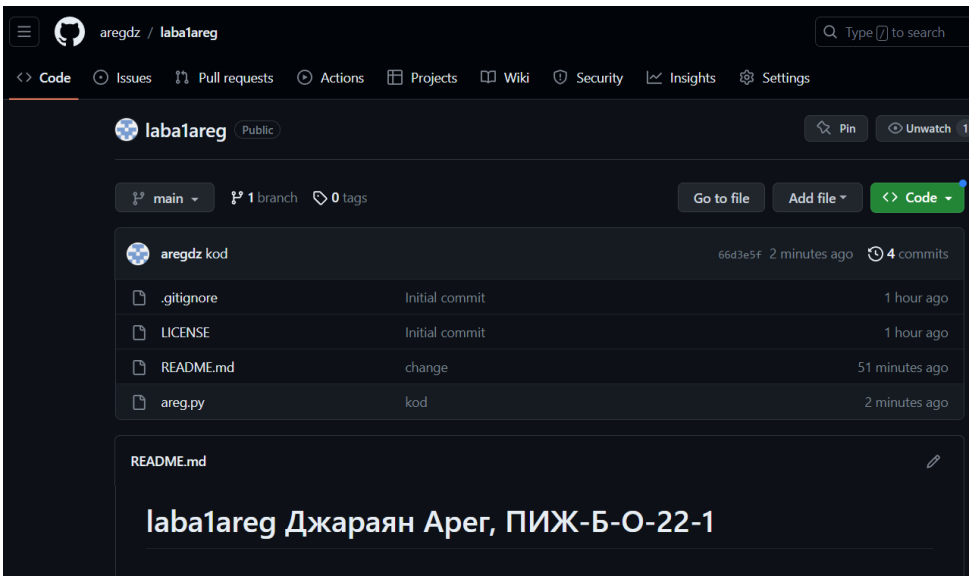


Рисунок 1.10 – Репозиторий

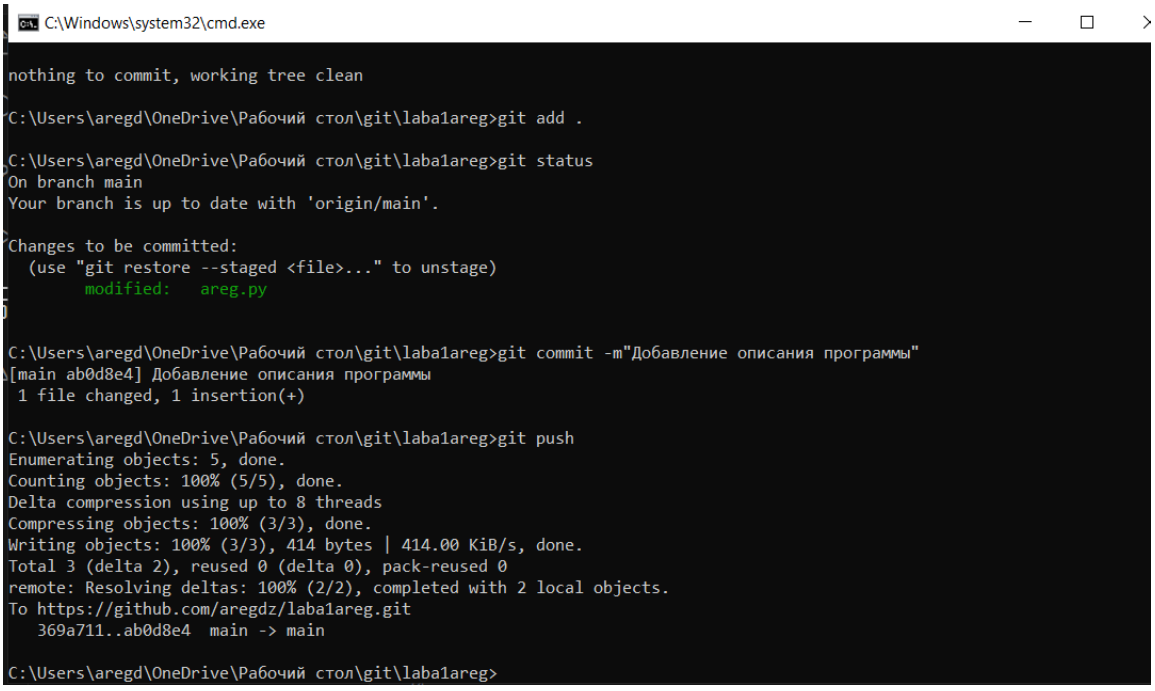


Рисунок 1.11 – Создание коммитов

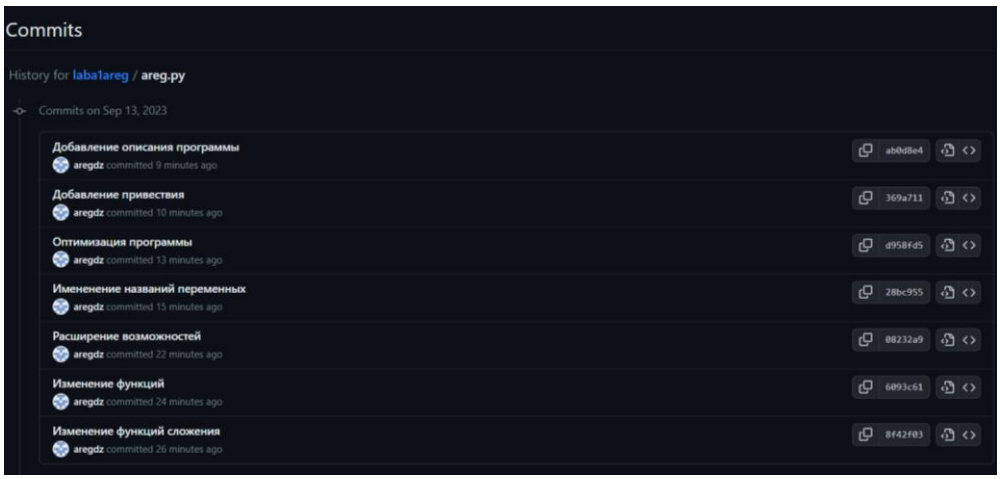


Рисунок 1.12 – коммиты

7. Изменил файл README и зафиксировал изменения.

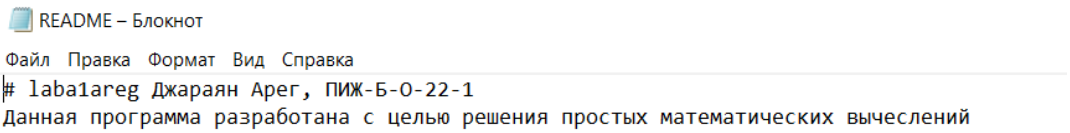


Рисунок 1.13 – файл README

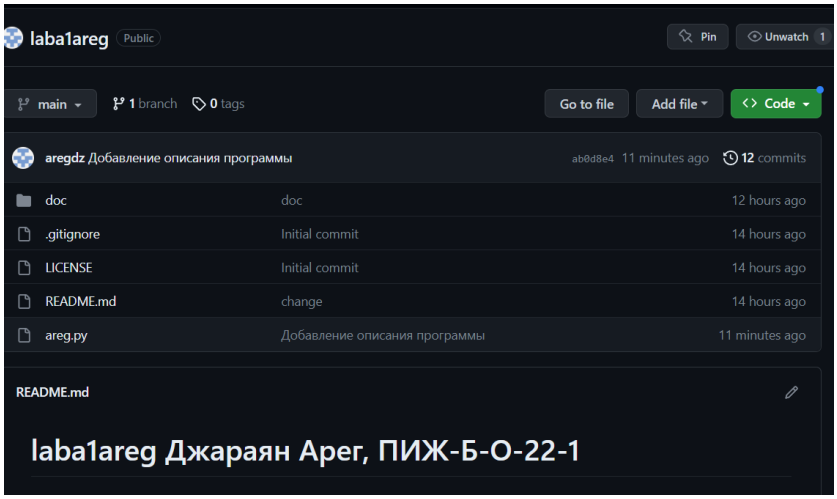


Рисунок 1.14 - Репозиторий

Ответы на 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.

2. В чем недостатки локальных и централизованных СКВ?

Недостаток локальных СКВ в том, что он невероятно 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.

Самый очевидный минус ЦСКВ — 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.

3. К какой СКВ относится Git?

Распределенные СКВ.

4. В чем концептуальное отличие Git от других СКВ?

Простое ветвление. В других СКВ создание веток— утомительная и трудоёмкая задача, так как весь код копируется в новую ветку. В Git управление ветками реализовано гораздо проще и эффективнее.

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 находится файлы в Git? Как связаны эти состояния?

У Git есть три основных состояния, в которых могут находиться ваши файлы: **изменён** (modified), **индексирован** (staged)
и **зафиксирован** (committed):

7. Что такое профиль пользователя в GitHub?

На странице пользователя отображаются сведения о работе через репозитории, вклад, который был внесен, и беседы.

8. Какие бывают репозитории в GitHub?

Репозиторий Git бывает локальный и удалённый. Локальный репозиторий — это подкаталог .git, создаётся (в пустом виде) командой git init и (в непустом виде с немедленным копированием содержимого родительского удалённого репозитория и простановкой ссылки на родителя) командой git clone.

9. Укажите основные этапы модели работы с GitHub.

GitHub - это платформа для размещения кода. Иными словами, это место, где разработчики могут хранить свои проекты и работать вместе. Таким образом контролировать версии программ и сотрудничать становится гораздо проще. GitHub основан на популярной системе управления версиями под названием Git и предоставляет некоторые дополнительные функции, такие как веб-интерфейс, инструменты совместной работы, средство отслеживания ошибок, статистика проекта и многое другое.

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:7

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

11. Опишите 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 результате будет выполнен переход на страницу создания репозитория, на ней будут поля

после заполнения которых, нажимаем кнопку Create repository. Отлично, ваш репозиторий готов!

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

В нашем случае используется MIT License.

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. Откройте командную строку или терминал и перейдите в каталог, куда вы хотите скопировать хранилище. Затем напишите git clone и введите адрес.

14. Как проверить состояние локального репозитория Git?

Использовать команду «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 ?

На локальном репозитории создается или изменяется файл. С помощью команды git commit происходит сохранение изменений. С помощью команды git push изменения отправляются в удаленный репозиторий.

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.

Использовать команду git push.

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

Bitbucket поддерживает функцию pull запроса, которая помогает загрузить проект из платформы. GitHub также поддерживает функцию pull запроса и помогает пользователю получить проект с платформы. В GitLab такая функция pull запроса отсутствует, и вместо нее в платформе GitLab поддерживается merge запрос.

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.

SmartGit также является кроссплатформенным, мощным, популярным Git-клиентом с графическим интерфейсом для Linux, Mac OS X и Windows. Он называется Git для профессионалов. Он позволяет пользователям справляться с ежедневными задачами Git и повышает их производительность за счет эффективных рабочих процессов.