

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**

Выполнила:
Кузнецова Алена Валерьевна
1 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

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

Выполнение работы:

1. Создали аккаунт на GitHub.

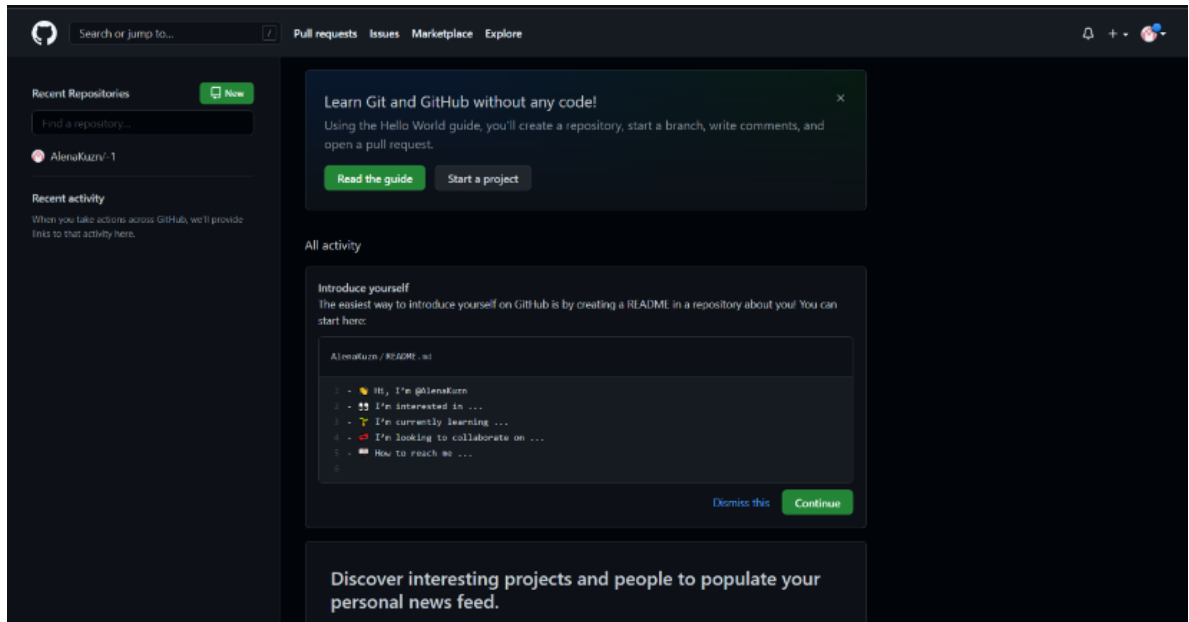


Рисунок 1 – Аккаунт на GitHub

2. Проверка успешной установки Git.

```
akuzn@LAPTOP-ARKAM4ED MINGW64 ~/github (main)
$ git version
git version 2.35.1.windows.2
```

Рисунок 2 – Git установлен

3. Создание репозитория.

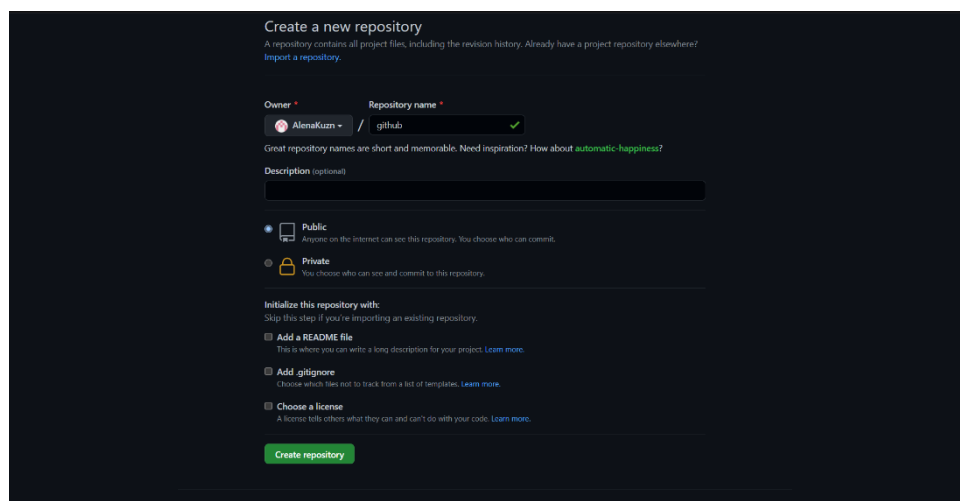


Рисунок 3 – Создание и настройка репозитория

4. Клонирование репозитория.

```
C:\Users\akuzn>git clone https://github.com/AlenaKuzn/github.git
Cloning into 'github'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\Users\akuzn>
```

Рисунок 4 - Клонирование репозитория

5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки.

```
1
2 # Created by https://www.toptal.com/developers/gitignore/api/c++,visualstudio
3 # Edit at https://www.toptal.com/developers/gitignore?templates=c++,visualstudio
4
5 ### C++ ###
6 # Prerequisites
7 *.d
8
9 # Compiled Object files
10 *.slo
11 *.lo
12 *.o
13 *.obj
14
15 # Precompiled Headers
16 *.gch
17 *.pch
18
19 # Compiled Dynamic libraries
20 *.so
21 *.dylib
22 *.dll
23
24 # Fortran module files
25 *.mod
26 *.smod
27
28 # Compiled Static libraries
29 *.lai
30 *.la
31 *.a
32 *.lib
33
34 # Executables
35 *.exe
36 *.out
37 *.app
38
```

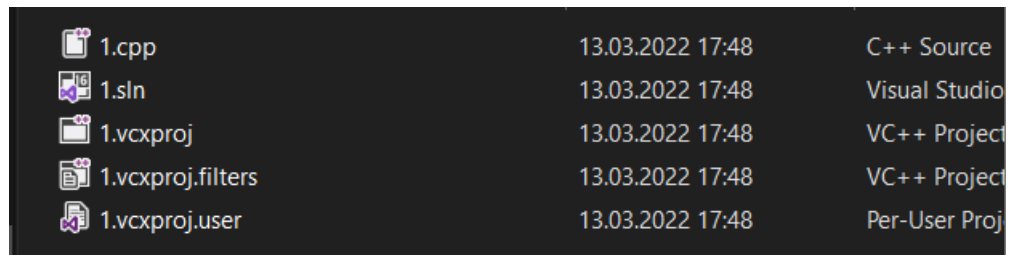
Рисунок 5 – Вид измененного файла .gitignore

6. Добавьте в файл README.md информацию о группе и ФИО студента, выполняющего лабораторную работу.

```
# github
Кузнецова Алена Валерьевна
ИВТ-6-о-21-1(2)
```

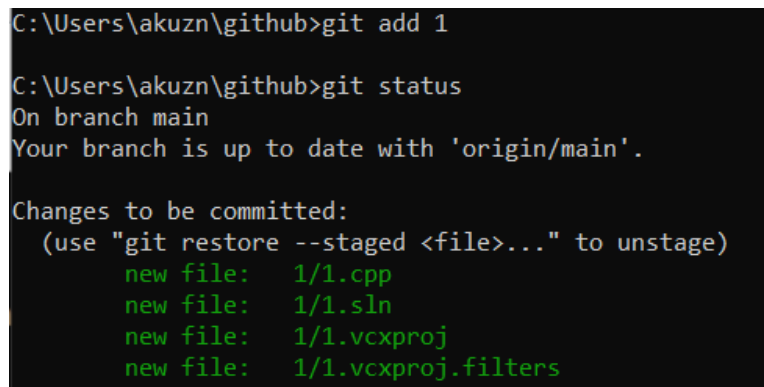
Рисунок 6 – Измененный файл README

7. Напишите небольшую программу на выбранном Вами языке программирования.



1.cpp	13.03.2022 17:48	C++ Source
1.sln	13.03.2022 17:48	Visual Studio
1.vcxproj	13.03.2022 17:48	VC++ Project
1.vcxproj.filters	13.03.2022 17:48	VC++ Project
1.vcxproj.user	13.03.2022 17:48	Per-User Proj

Рисунок 7 – Файлы с программой на языке C++

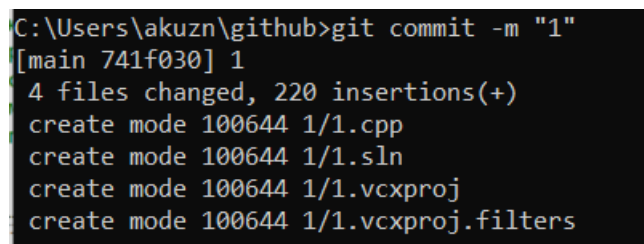


```
C:\Users\akuzn\github>git add 1

C:\Users\akuzn\github>git status
On branch main
Your branch is up to date with 'origin/main'.

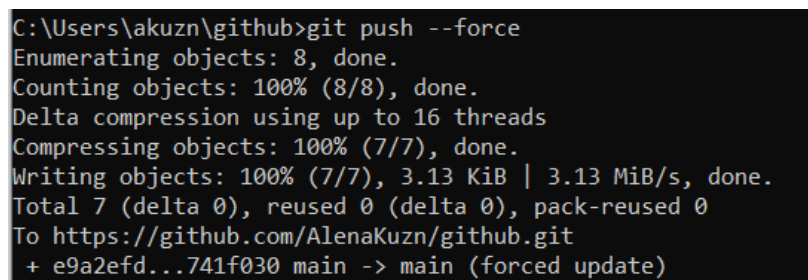
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   1/1.cpp
    new file:   1/1.sln
    new file:   1/1.vcxproj
    new file:   1/1.vcxproj.filters
```

Рисунок 8 – Добавили файл с программой



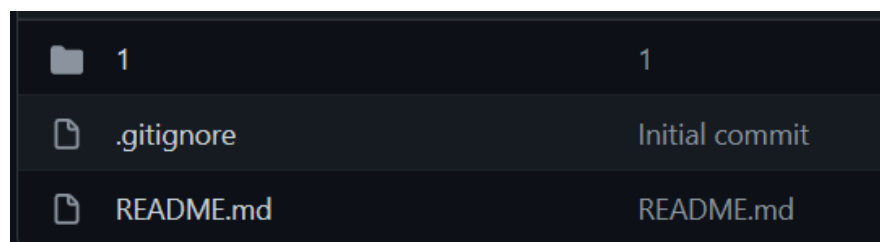
```
C:\Users\akuzn\github>git commit -m "1"
[main 741f030] 1
4 files changed, 220 insertions(+)
create mode 100644 1/1.cpp
create mode 100644 1/1.sln
create mode 100644 1/1.vcxproj
create mode 100644 1/1.vcxproj.filters
```

Рисунок 9 – Добавление коммита



```
C:\Users\akuzn\github>git push --force
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 3.13 KiB | 3.13 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AlenaKuzn/github.git
+ e9a2efd...741f030 main -> main (forced update)
```

Рисунок 10 – Отправка на удаленный репозиторий



1	1
.gitignore	Initial commit
README.md	README.md

Рисунок 11 – Папка с программой успешно добавлена

Вопросы:

1. Что такое СКВ и каково ее назначение?

Система контроля версий — 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?

Недостатки локальных СКВ в том, что 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

Недостатки централизованной СКВ в том, что 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.

3. К какой СКВ относится Git?

Распределённой

4. В чем концептуальное отличие Git от других СКВ?

Это очень важное отличие между Git и почти любой другой СКВ. 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. 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 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.

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?

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`. Практически все обычные операции с системой контроля версий, такие, как коммит и слияние, производятся только с локальным репозиторием. Удалённый доступ к репозиториям Git обеспечивается `git-daemon`, SSH или HTTP-сервером. TCP-сервис `git-daemon` входит в дистрибутив Git и является наряду с SSH наиболее

распространённым и надёжным методом доступа. Удалённый репозиторий можно только синхронизировать с локальным как «вверх» (push), так и «вниз» (pull).

9. Укажите основные этапы модели работы с GitHub.

- 1) регистрация;
- 2) создание репозитория;
- 3) клонирование репозитория.

10. Как осуществляется первоначальная настройка Git после установки?

- 1) убедимся, что Git установлен используя команду: `git version`;
- 2) перейдём в папку с локальным репозиторием, используя команду: `cd /d`;
- 3) свяжем локальный репозиторий и удалённый.

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. Наиболее важными на ней 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 репозиториях, 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.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Microsoft Reciprocal License, The Code Project Open License (CPO), The Common Development and Distribution License (CDDL), The Microsoft Public

License (Ms-PL), The Mozilla Public License 1.1 (MPL 1.1), The Common Public License Version 1.0 (CPL), The Eclipse Public License 1.0, The MIT License, The BSD License, The Apache License, Version 2.0, The Creative Commons Attribution-ShareAlike 2.5 License, The zlib/libpng License, A Public Domain dedication, The Creative Commons Attribution 3.0 Unported License, The Creative Commons Attribution-Share Alike 3.0 Unported License, The Creative Commons Attribution-NoDerivatives 3.0 Unported, The GNU Lesser General Public License (LGPLv3), The GNU General Public License (GPLv3).

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?

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`?

Коммит добавляет изменения только в ваш локальный репозиторий. Если вы хотите распространить их в исходный репозиторий на GitHub, вам нужно использовать `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`.

Примечание: описание необходимо начать с команды `git clone`.

1) Клонировать репозиторий на каждый из компьютеров, используя команду `git clone` и ссылку.

2) Для синхронизации изменений используем 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.

GitLab — альтернатива GitHub номер один. GitLab предоставляет не только веб-сервис для совместной работы, но и программное обеспечение с открытым исходным кодом.

Launchpad — платформа для совместной работы над программным обеспечением от Canonical, компании-разработчика Ubuntu. На ней размещены PPA-репозитории Ubuntu, откуда пользователи загружают приложения и обновления.

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.

GitHub Desktop это бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub, а также с другими платформами (включая GitLab).

Вывод: мы исследовали базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.