

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 практической работе №1.1
дисциплины «Программирование на Python»

Выполнил:

Пустяков Андрей Сергеевич

2 курс, группа ИВТ-б-о-22-1,

09.03.01 «Информатика и
вычислительная техника (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных
систем», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р. А. кандидат технических
наук, доцент, доцент кафедры
инфокоммуникаций

(подпись)

Ставрополь, 2023 г.

Тема: Исследование основных возможностей Git и GitHub.

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

Ход работы:

Задание 2.

Создание общедоступного репозитория на GitHub с использованием лицензии MIT и выбранным языком программирования (Python) (рис. 1).

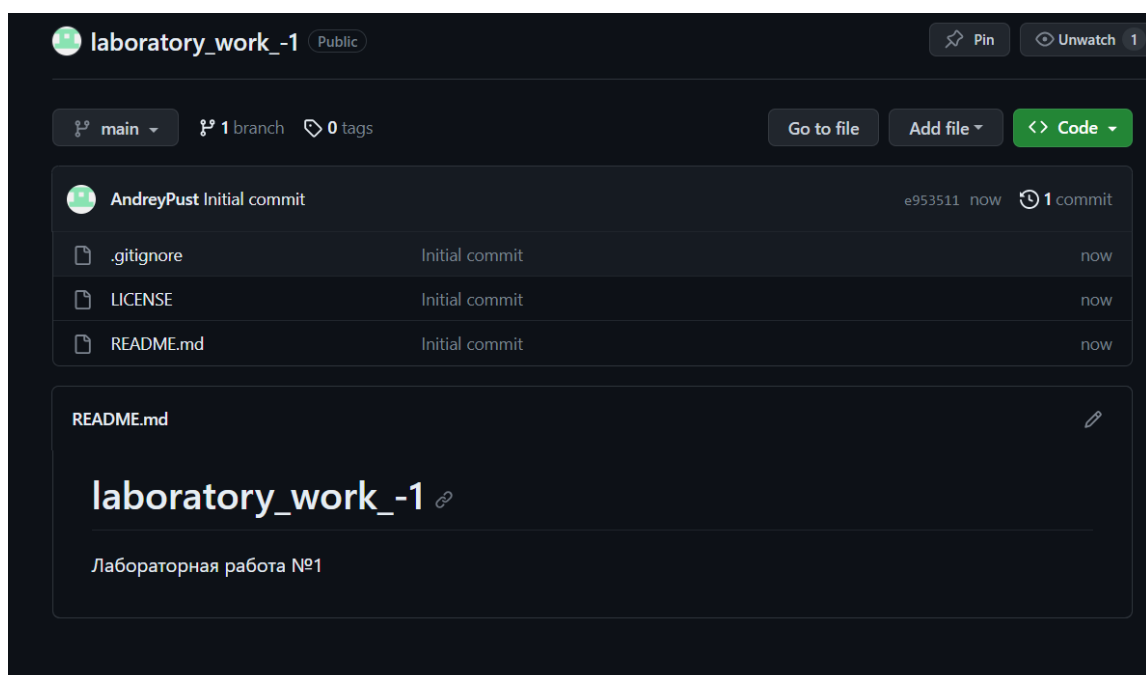
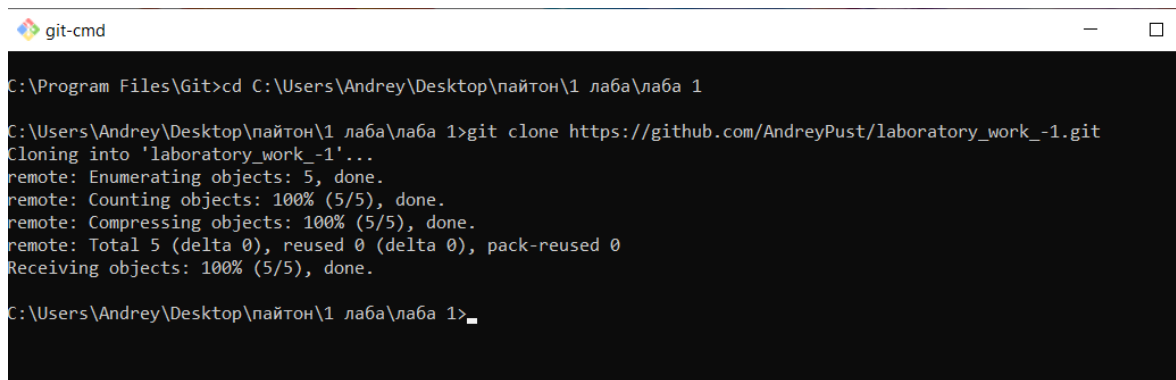


Рисунок 1 – Общедоступный репозиторий с использованием лицензии «MIT» и файлом «.gitignore» для языка программирования Python.

Задание 3.

Клонирование удаленного репозитория на рабочий компьютер в конкретную папку (рис. 2).

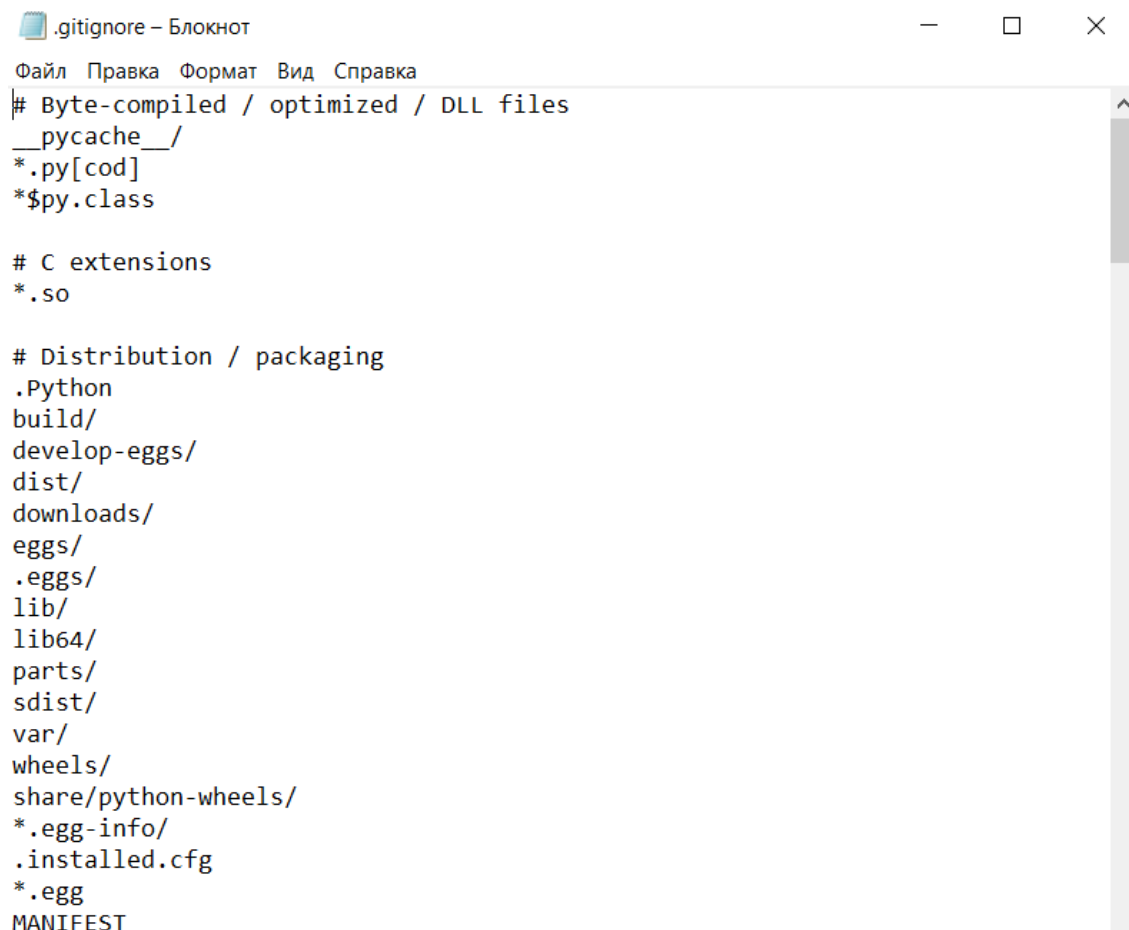


```
git-cmd
C:\Program Files\Git>cd C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1
C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1>git clone https://github.com/AndreyPust/laboratory_work_-1.git
Cloning into 'laboratory_work_-1'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1>_
```

Рисунок 2 – Клонирование репозитория на локальный ПК.

Задание 4.

Редактирование файла «.gitignore» для выбранного языка программирования (Python) и среды (PyCharm) (рис. 3).



```
.gitignore – Блокнот
Файл  Правка  Формат  Вид  Справка
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
```

Рисунок 3 – Добавление пунктов в файл «gitignore».

Задание 5.

Добавление в файл «README» информации об имени и группе (рис. 4).

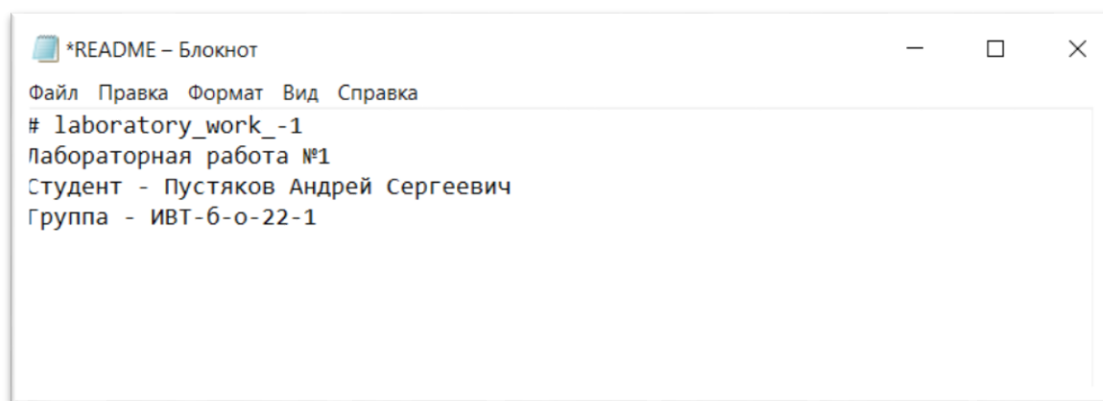


Рисунок 4 – Редактирование файла «README».

Создание простенькой программы на языке программирования «Python» и добавление ее в папку репозитория, создание 7-ми коммитов (рис. 5, 6).

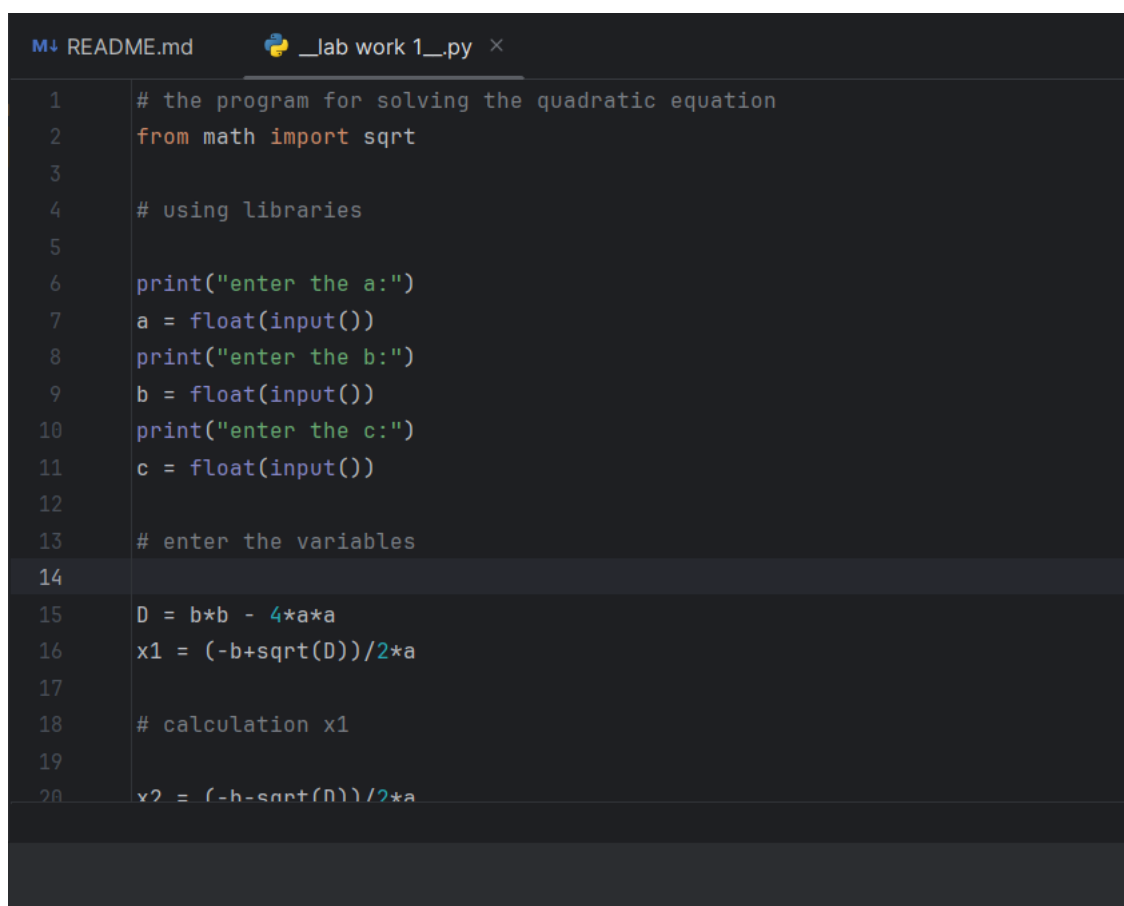


Рисунок 5 – Создание небольшой программы вычисления корней квадратного уравнения.

```

create mode 100644 .idea/modules.xml
create mode 100644 .idea/vcs.xml
create mode 100644 __lab work 1__.py

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git add .

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git commit -m "add (print)"
[main 4fa72f5] add (print)
1 file changed, 1 insertion(+)

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git add .

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git commit -m "add beginning programm"
[main e174630] add beginning programm
1 file changed, 6 insertions(+), 1 deletion(-)

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>gid add .
"gid" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git add .

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git commit -m "add programm"
[main 185f1da] add programm
1 file changed, 5 insertions(+), 1 deletion(-)

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git add .

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git commit -m "add a small modifications"
[main 1092153] add a small modifications
1 file changed, 3 insertions(+), 1 deletion(-)

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git add .

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git commit -m "adding the comments"
[main 68d3f79] adding the comments
1 file changed, 4 insertions(+), 4 deletions(-)

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git add .

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git commit -m "editing a hearth for the pep8 standard"
[main 3fa72b0] editing a hearth for the pep8 standard
1 file changed, 16 insertions(+), 4 deletions(-)

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>

```

Рисунок 6 – Создание 7-ми коммитов.

Задание 7.

Добавление в репозиторий файла «README» и фиксация изменений (рис. 7).

```

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git add README.md

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>git commit -m "adding file README"
[main c4e7b5e] adding file README
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_1>

```

Рисунок 7 – Добавление файла «README».

Задание 8.

Добавление в репозиторий отчета в формате «PDF» в папку «doc» и фиксация полученных изменений (рис. 8).

```

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_-1>git add .

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_-1>git commit -m "addind pdf"
[main b8f350a] addind pdf
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 "doc/\320\236\321\202\321\207\320\265\321\202 \320\277\320\276 \320\237\
\321\201\321\202\321\217\320\272\320\276\320\262.pdf"

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_-1>

```

Рисунок 8 – Добавление отчета в формате «pdf».

Задание 9.

Отправка изменений в локальном репозитории на удаленный репозиторий «GitHub» (рис. 9).

```

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_-1>git push --set-upstream origin main
info: please complete authentication in your browser...
Enumerating objects: 39, done.
Counting objects: 100% (39/39), done.
Delta compression using up to 8 threads
Compressing objects: 100% (33/33), done.
Writing objects: 100% (37/37), 588.13 KiB | 25.57 MiB/s, done.
Total 37 (delta 13), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (13/13), done.
To https://github.com/AndreyPust/laboratory_work_-1.git
 e953511..b8f350a main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\Andrey\Desktop\пайтон\1 лаба\лаба 1\laboratory_work_-1>

```

Рисунок 9 – Отправка изменений в удаленный репозиторий.


Задание 10.

Проверка удаленного репозитория на «GitHub» на наличие изменений (рис. 10).

laboratory_work_-1

Add file


...









AndreyPust

addind pdf

b8f350a · 36 minutes ago

 History

Name	Last commit message	Last commit date
 .idea	add file programm	1 hour ago
 doc	addind pdf	36 minutes ago
 .gitignore	Initial commit	last week
 LICENSE	Initial commit	last week
 README.md	adding file README	44 minutes ago
 _lab work 1_.py	editing a hearth for the pep8 standard	52 minutes ago

README.md




Рисунок 10 – Просмотр удаленного репозитория.

Ссылка на репозиторий: https://github.com/AndreyPust/laboratory_work_1.git

Ответы на контрольные вопросы:

1. СКВ – это система контроля версий, ее назначение заключается в версионном контроле файлов.

2. Недостатками локальных СКВ можно назвать невозможность взаимодействия с другими людьми и другими разработчиками, а также риск потери данных и всех версий при возникновении проблем с локальным компьютером. Недостатками централизованных СКВ можно назвать единую точку отказа, при которой проблемы на централизованном сервере ведут к невозможности использовать СКВ, при повреждении центрального сервера утрачиваются все версии, кроме тех, которые находились на локальных компьютерах.

3. Git относится к распределенной системе контроля версий, так как имеет как локальные хранилища (репозитории), так и хранилища на сервере.

4. Git отличается от других систем контроля версий тем, что у данной системы свой подход к работе с данными, а именно представляет данные как поток сделанных в разное время снимков.

5. Целостность данных в git обеспечивается благодаря принципу вычисления хеш-суммы содержимого по механизму SHA-1, таким образом git отслеживает все изменения и защищает файлы от повреждения и изменения, все объекты в Git хранятся не по именам, а по хеш-сумме.

6. Файлы в Git могут храниться в трех состояниях: зафиксированном (сохраненном на локальной базе), измененном (те файлы, которые поменялись, но еще не были зафиксированы), подготовленном (измененные файлы, которые были отмечены в следующий коммит для фиксации). Связаны они следующим образом – в рабочей директории файлы изменяются пользователем и становятся измененными, добавляются файлы в коммит

(помечаются для коммита) и становятся подготовленными, а после коммита файлы в директории фиксируются и становятся зафиксированными.

7. Профиль пользователя на GitHub – это публичная страница пользователя разработчика, на которой размещена какая-либо информация о нем и ссылки на репозитории проектов, созданных пользователем.

8. В GitHub в зависимости от доступа могут быть общедоступными (публичными), приватными (доступными только для соавторов проекта).

9. Для работы с GitHub необходимо зарегистрироваться на сервисе для создания профиля на GitHub. Для работы с сервисом можно создавать репозитории, указывая доступ к ним, загружать на них свои проекты и другие пользователи сервиса могут их просматривать или даже участвовать в проекте.

10. После установки Git необходимо убедиться в том, что он успешно установлен написав в терминале команду «git version», после чего необходимо указать фамилию и адрес электронной почты (связанные с профилем на GitHub) с помощью команд «git config --global user.name <YOUR_NAME>» и «git config --global user.email <EMAIL>». Для начала работы с проектом необходимо создать удаленный репозиторий. Создать директорию для нового репозитория, перейти в нее, скопировать туда удаленный репозиторий, внести изменения и отправить их на удаленный репозиторий создав удаленную ветку.

11. Для создания репозитория на GitHub необходимо выбрать кнопку создания нового репозитория. После перехода на страницу создания нового репозитория необходимо заполнить поля имени репозитория (оно должно быть уникальным только для всех проектов внутри одного профиля на GitHub), поле описания проекта (поле не является обязательным), выбрать вид репозитория (публичный или приватный в зависимости от того, кому и как будет дан доступ к будущему проекту), выбрать лицензию и файл «.gitignore» и после чего создать репозиторий. Данный репозиторий можно клонировать локально и работать с ним.

12. При создании репозитория на GitHub поддерживается множество лицензий со своими особенностями и требованиями, среди них существует популярная лицензия «MIT», которая позволяет использовать код другим разработчикам свободно, но только при указании первоисточника. В основном, лицензии на GitHub характеризуются свободой использования кода других пользователей в своих проектах или гибкостью коммерческого использования, а также требованием к сохранению авторских прав на код в проекте.

13. Клонирование репозитория с сервиса GitHub осуществляется после настройки Git (согласно профилю на GitHub) на локальном компьютере. Для клонирования репозитория необходимо получить его адрес, сделать это можно с профиля на GitHub нажав кнопку «clone». После получения адреса репозитория его можно скопировать, выполнив команду в терминале «git clone ...» написав соответствующий адрес репозитория, предварительно перейдя в терминале в каталог, в который мы хотим скопировать репозиторий. Клонировать репозиторий необходимо для того, чтобы получить копию локального хранилища проекта и вносить в нее изменения локально.

14. Для того, чтобы проверить состояние локального репозитория необходимо ввести команду в терминал Git «git status», заранее перейдя в каталог, где хранится репозиторий.

15. Состояние данных локального репозитория после внесения в него изменений изменится на состояние измененных, даже при добавлении нового файла в локальный репозиторий. После добавления измененного или нового файла под версионный контроль, данные в репозитории изменятся на состояние зафиксированных. После команды «git commit» произойдет фиксация новой версии данных в репозитории, а после отправки изменений на сервер с помощью команды «git push», изменения отправятся и на удаленный репозиторий GitHub.

16. Для того, чтобы осуществлять работу на двух локальных компьютерах над одним репозиторием на одной странице GitHub и чтобы

локальные репозитории находились в синхронизированном состоянии, необходимо на обоих компьютерах выполнить команду в терминале Git «git clone», тем самым получив копию репозитория на сервере. Для того, чтобы вносить изменения в репозиторий на сервере, во избежание конфликтов необходимо выполнять команду «git pull» с псевдонимом репозитория и названия ветки (чаще всего это основная ветка «main») для извлечения истории изменений на локальные репозитории, причем делать это постоянно при внесении изменений на обоих компьютерах. Отправлять изменения при помощи команды «git push» можно только после того, как мы в очередной раз получили на локальный компьютер историю изменений.

17. С Git помимо GitHub также могут работать такие сервисы как Fork, Gitlab (сервис очень схожий с GitHub), Bitbucket, Gogs. «Fork» по сравнению с «GitHub», может быть быстрее, проявляется дружелюбность к пользователю и эффективность использования. К особенностям «Fork» можно отнести привлекательный вид, наличие кнопок быстрого доступа, встроенная система разрешения конфликтов слияния, менеджер репозитория. Основная особенность «Fork» - скорость и простота использования.

18. Интерфейс командной строки является не единственным способом взаимодействовать и работать с «Git». Существуют и другие программные средства с графическими интерфейсами, например «Git GUI», который предназначен для того, чтобы работать с системой контроля версий в приятном интерфейсе. Все операции, которые осуществлялись в командной строке теперь реализуются при помощи удобного интерфейса и соответствующих кнопок. С помощью соответствующих кнопок можно создать новый репозиторий, клонировать репозиторий. Кнопка «этап изменен» добавляет файлы в «Git» под индекс.

Вывод: в ходе лабораторной работы были изучены основные особенности работы с «Git» и «GitHub». Были изучены особенности и назначения СКВ, их виды. Была проведена начальная настройка «Git», был создан аккаунт на «GitHub» и новый репозиторий с небольшим проектом,

были изучены принципы настройки файла «.gitignore». Были изучены особенности веток в СКВ как локальных, так и удаленных.