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 №1.1
дисциплины «Основы кроссплатформенного программирования»

Выполнила:
Евдаков Евгений Владимирович
1 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные технологии и системы связи»,
направленность (профиль) «Инфокоммуникационные системы и сети», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

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:

Вариант №10

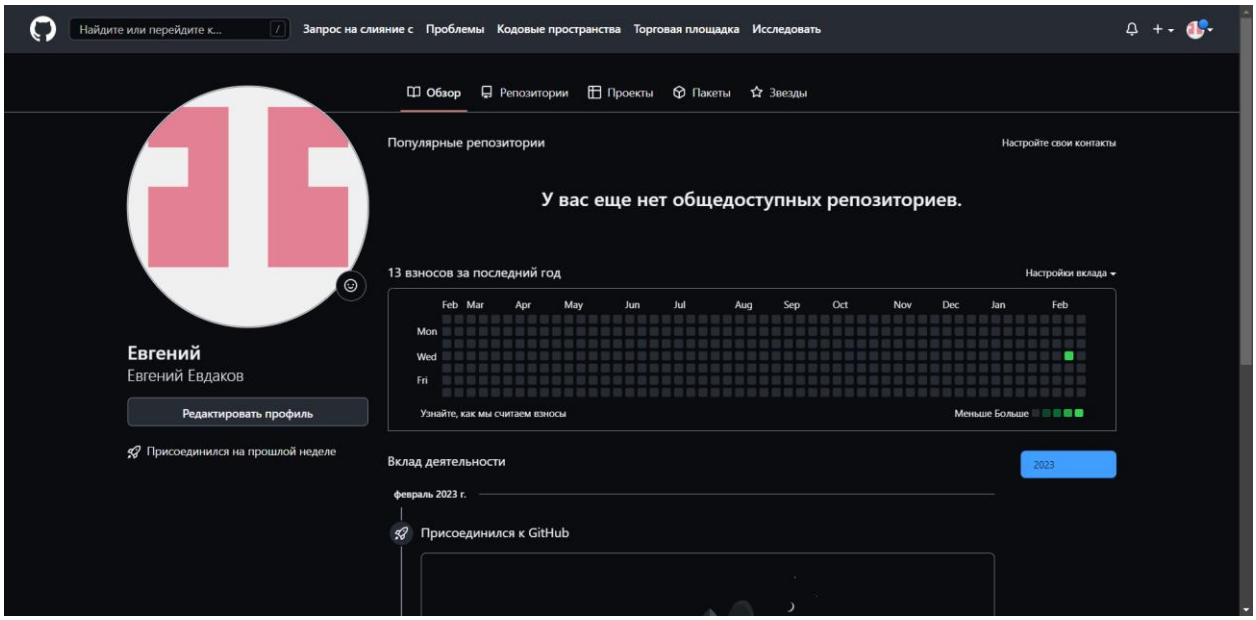

Рисунок 1. Создание аккаунта на GitHub

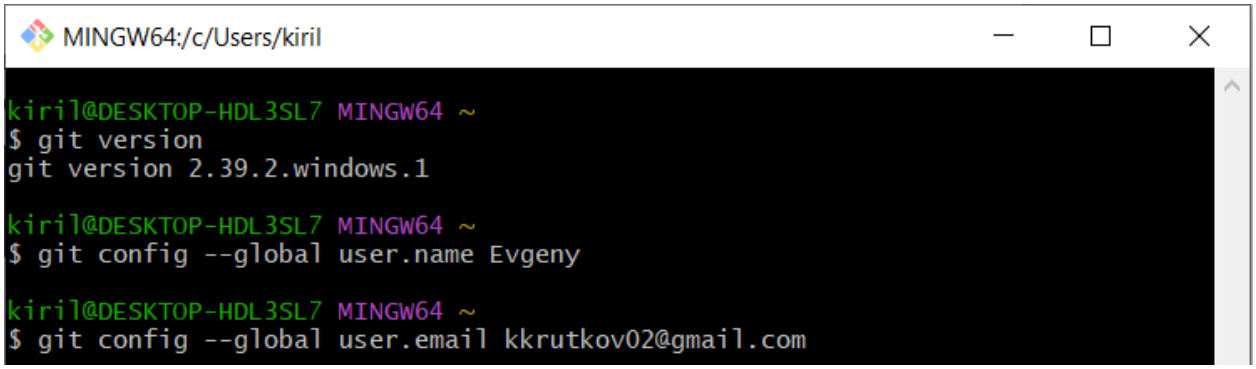

Рисунок 2. Версия Git и добавление имени и электронной почты

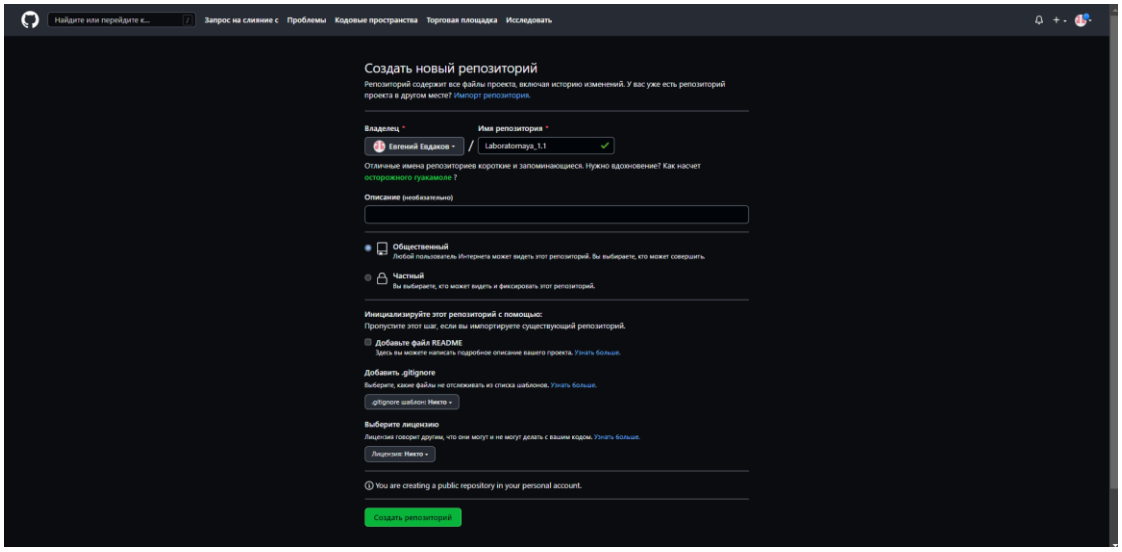

Рисунок 3. Создание репозитория GitHub

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~
$ git clone https://github.com/EvgenyEvdakov/EvdLaba.git
Cloning into 'EvdLaba'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

Рисунок 4. Клонирование репозитория

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the nex
t time Git touches it

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git add .

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Add information about local repository in readme file"
[main 8bd4e38] Add information about local repository in readme file
 1 file changed, 1 insertion(+)
```

Рисунок 5. Изменение файла README, добавление и коммит

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/EvgenyEvdakov/EvdLaba.git
   dfe4b43..8bd4e38  main -> main
```

Рисунок 6. Git push файла README

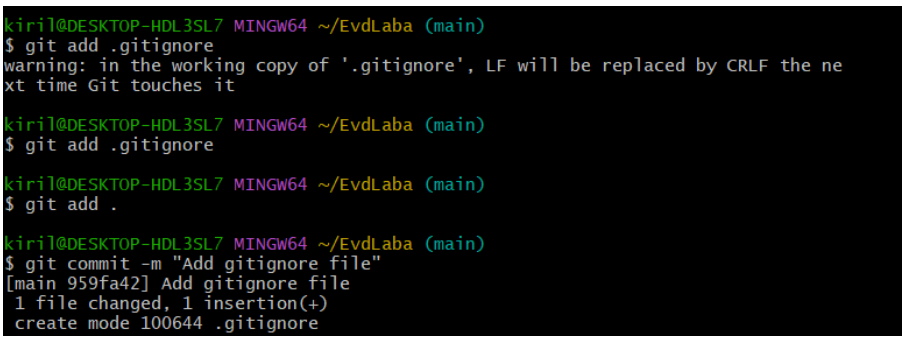

Рисунок 7. Локальное добавление .gitignore, commit

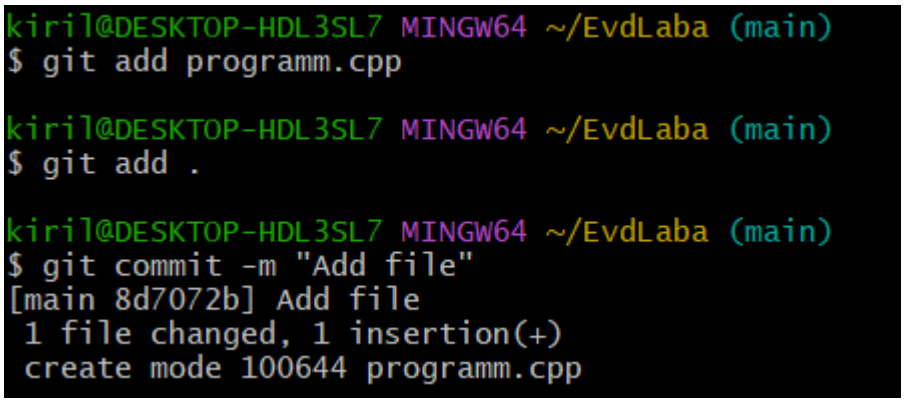

Рисунок 8. Добавление programm.cpp, commit 1 и push

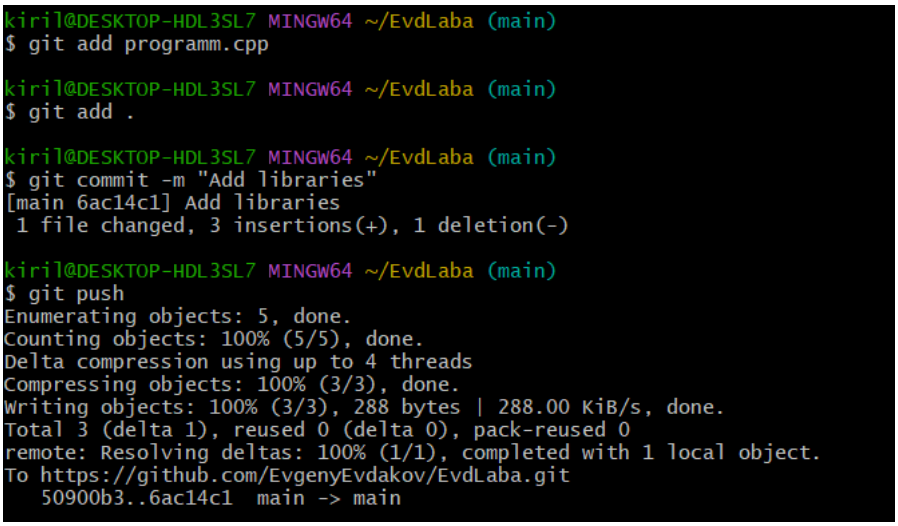

Рисунок 9. Commit 2 и push

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git  add programm.cpp

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git add .

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Add main and setlocale"
[main eee3088] Add main and setlocale
 1 file changed, 6 insertions(+), 1 deletion(-)

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 335 bytes | 335.00 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/EvgenyEvdakov/EvdLaba.git
   6ac14c1..eee3088  main -> main
```

Рисунок 10. Commit 3 и push

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Add float"
[main 33c05f1] Add float
 1 file changed, 1 insertion(+)

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 309 bytes | 309.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/EvgenyEvdakov/EvdLaba.git
   eee3088..33c05f1  main -> main
```

Рисунок 11. Commit 4 и push

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Add printf and scanf f:a, b, c"
[main 6a75531] Add printf and scanf f:a, b, c
 1 file changed, 6 insertions(+)

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 439 bytes | 439.00 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/EvgenyEvdakov/EvdLaba.git
   33c05f1..6a75531  main -> main
```

Рисунок 12. Commit 5 и push

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Add Beginning and End x"
[main e95215a] Add Beginning and End x
 1 file changed, 4 insertions(+)

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 385 bytes | 385.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/EvgenyEvdakov/EvdLaba.git
   6a75531..e95215a  main -> main
```

Рисунок 13. Commit 6 и push

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Add move x"
[main 91dd468] Add move x
 1 file changed, 3 insertions(+)

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 381 bytes | 381.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/EvgenyEvdakov/EvdLaba.git
   e95215a..91dd468  main -> main
```

Рисунок 14. Commit 7 и push

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Added a function"
[main 9980ee7] Added a function
 1 file changed, 19 insertions(+)

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 424 bytes | 424.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/EvgenyEvdakov/EvdLaba.git
   91dd468..9980ee7  main -> main
```

Рисунок 15. Commit 8 и push

```
kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git commit -m "Add print F"
[main 79d66b1] Add print F
 1 file changed, 1 insertion(+)

kiril@DESKTOP-HDL3SL7 MINGW64 ~/EvdLaba (main)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 330 bytes | 330.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/EvgenyEvdakov/EvdLaba.git
   9980ee7..79d66b1  main -> main
```

Рисунок 16. Commit 9 и push

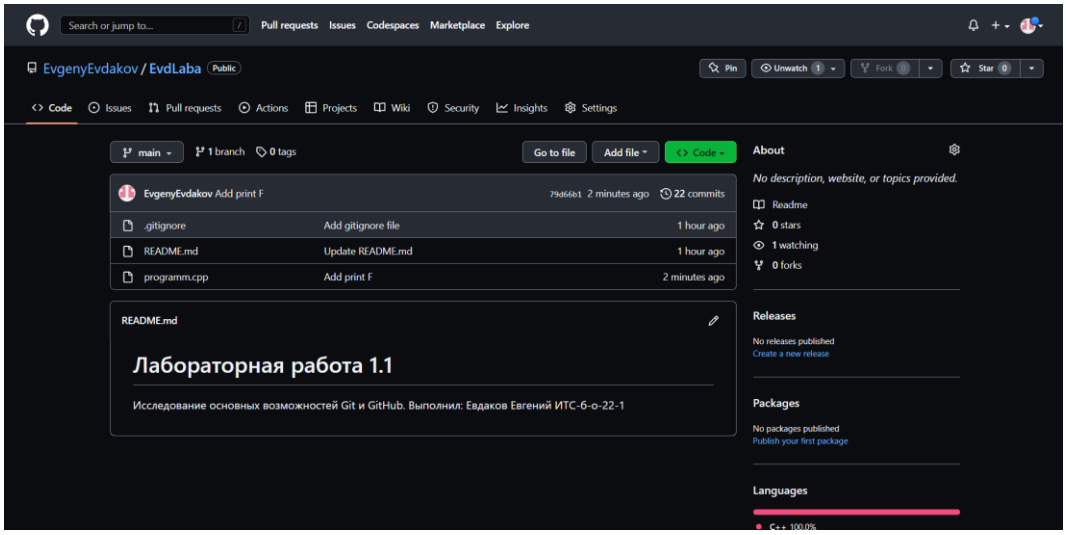

Рисунок 17. Обновленный удаленный репозиторий

Ссылка: https://github.com/EvgenyEvdakov/EvdLaba

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 этих файлов.

2. В чем недостатки локальных и централизованных СКВ?

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?

К распределённым системам контроля версий.

4. В чем концептуальное отличие Git от других СКВ?

Git не хранит и не обрабатывает данные таким же способом как другие СКВ.

5. Как обеспечивается целостность хранимых данных в Git?

В Git для всех данных вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме.

6. В каких состояниях могут находится файлы в Git? Как связаны эти состояния?

1) Зафиксированный значит, что файл уже сохранён в вашей

локальной базе;

2) К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы;

3) Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

7. Что такое профиль пользователя в GitHub?

Профиль - это наша публичная страница на GitHub, как и в социальных сетях. В нем другие пользователи могут посмотреть ваши работы и брать что-либо себе или участвовать в вашем проекте.

8. Какие бывают репозитории в GitHub?

Репозиторий — это хранилище всех версий кода. Он бывает трех видов:

1. Локальный — расположен на одном компьютере, и работать с ним может только один человек.

2. Централизованный — расположен на сервере, куда имеют доступ сразу несколько программистов.

3. Распределенный — самый удобный вариант с облачным хранилищем. Главный репозиторий хранится в облаке, а его локальные копии у разработчиков на компьютерах. Когда программист вносит правки в

локальную версию, ее можно синхронизировать с удаленной. Получается, что в облаке всегда актуальный код.

9. Укажите основные этапы модели работы с GitHub.

1) Регистрация;

2) Создание репозитория;

3) Клонирование репозитория;

4) Добавление новых файлов.

10. Как осуществляется первоначальная настройка Git после установки?

Необходимо убедиться, какой версии установленный Git используя команду: git version. Далее необходимо клонировать репозиторий командой: git clone ссылка. Далее в папке с локальным репозиторием копируем ссылку расположения папки и в git используем команду: cd /d <Расположения папки на компьютере>. Свяжем локальный репозиторий и удалённый командами: git config --global user.name <YOUR_NAME git config --global user.email <EMAIL>.

11. Опишите этапы создания репозитория в GitHub.

1) В правом верхнем углу, рядом с аватаром есть кнопка с плюсиком, нажимая которую мы переходим к созданию нового репозитория;

2) В результате будет выполнен переход на страницу создания репозитория. Наиболее важными на ней являются следующие поля: Имя репозитория. Описание (Description). Public/private. “Initialize this repository with a README” .gitignore и LICENSE.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

Microsoft Reciprocal License, The Code Project Open License (CPOL), 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).

13. Как осуществляется клонирование репозитория GitHub? Зачем нужно клонировать репозиторий?

После создания репозитория его необходимо клонировать на компьютер. Для этого на странице репозитория необходимо найти кнопку Clone или Code и щелкнуть по ней, чтобы отобразить адрес репозитория для клонирования.

Затем открыть командную строку или терминал и перейти в каталог, куда вы хотите скопировать хранилище. Затем написать git clone и ввести адрес.

Клонировать репозиторий нужно чтобы он был на компьютере со всеми созданными и измененными файлами.

14. Как проверить состояние локального репозитория Git?

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 ?

Файлы обновятся на репорзиории.

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.

1. git clone.

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

2) BitBucket — это служба хостинга репозиториев и управления версиями от Atlassian. Она тесно интегрирована с другими инструментами Atlassian — Jira, HipChat и Confluence.

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 совершенно бесплатное приложение с открытым исходным кодом, разработанное GitHub. С его помощью можно взаимодействовать с GitHub (что и не удивительно), а также с другими платформами (включая Bitbucket и GitLab).

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