МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ
ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития

Кафедра инфокоммуникаций

Дисциплина: «основы кроссплатформенного программирования»

ОТЧЕТ

по лабораторной работе №1

Тема: Исследование основных возможностей
Git и GitHub

Выполнил: студент 1 курса группы ИТС-21-1

Снадный Михаил Сергеевич

Проверил: к.ф.-м.н., доцент кафедры инфокоммуникаций

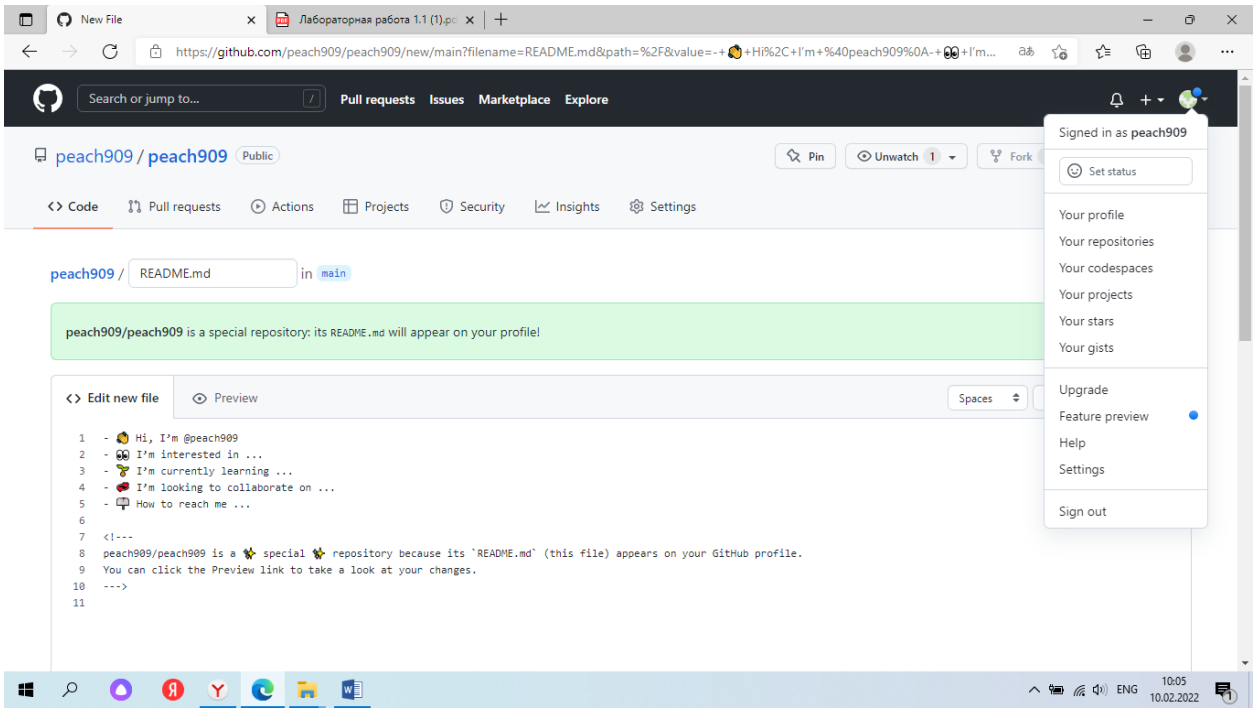
Воронкин Роман Александрович

Работа защищена с оценкой:_____

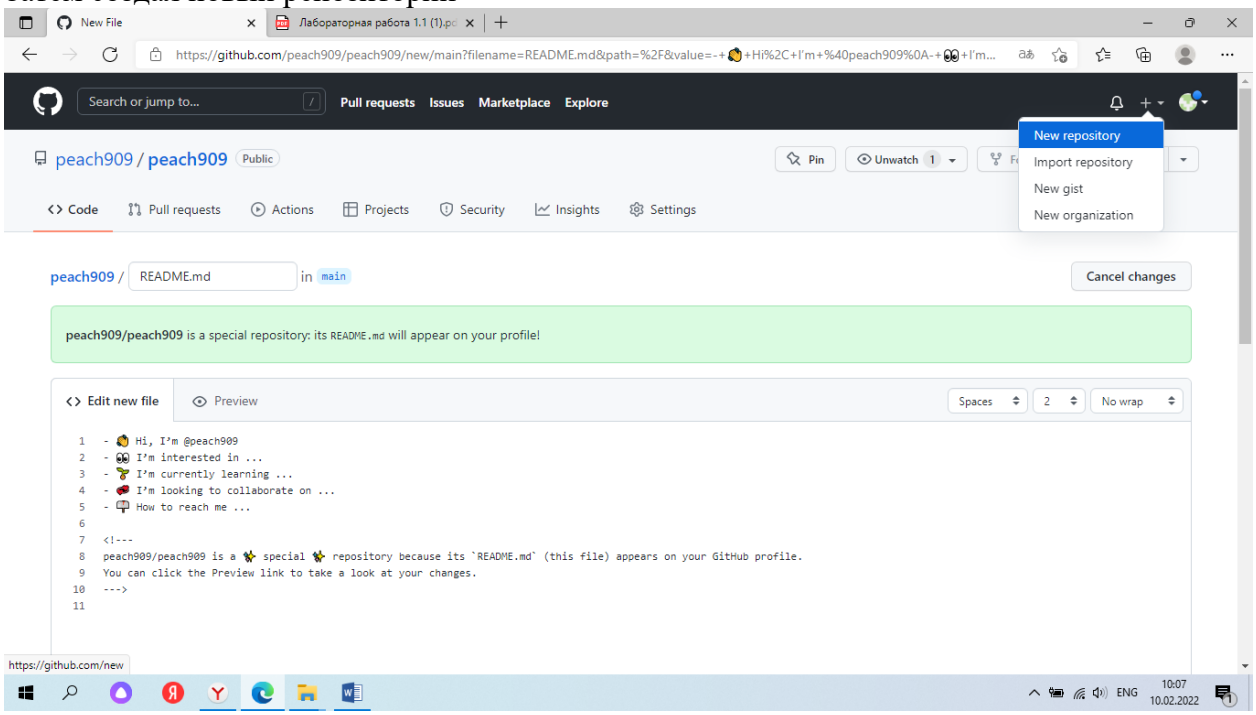
Ставрополь, 2021

Ход работы:

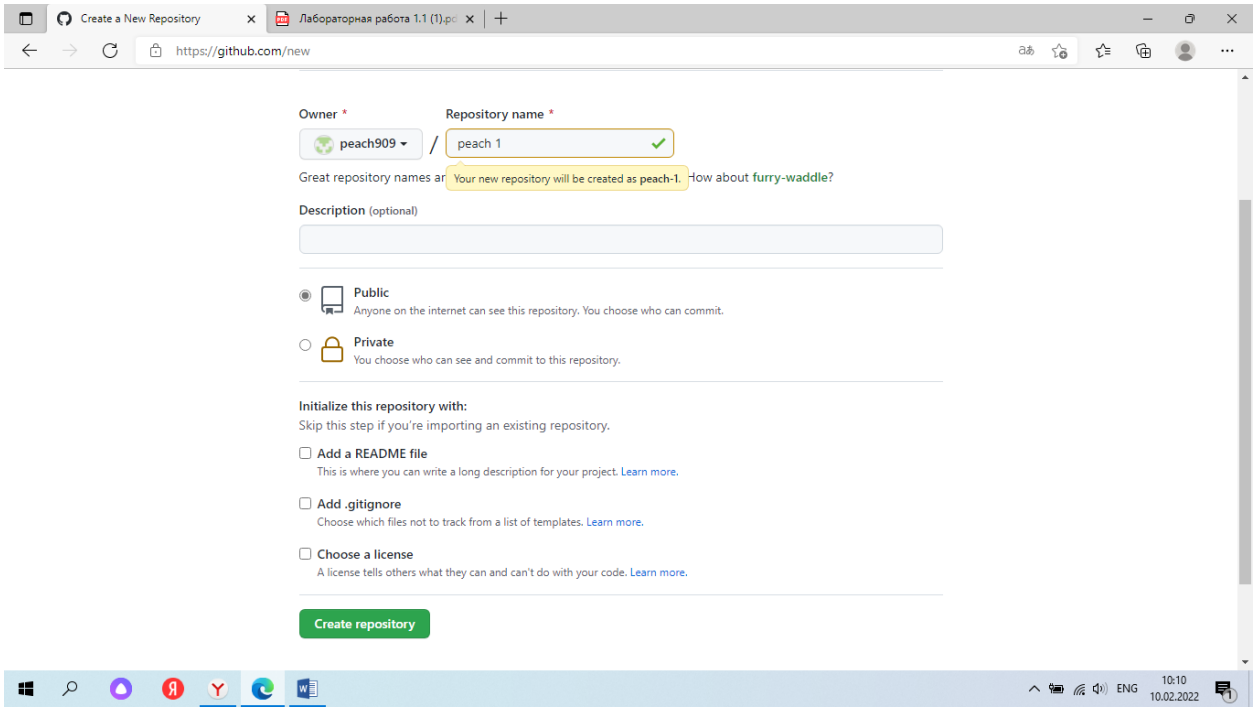
Зарегистрировался на сайте Git



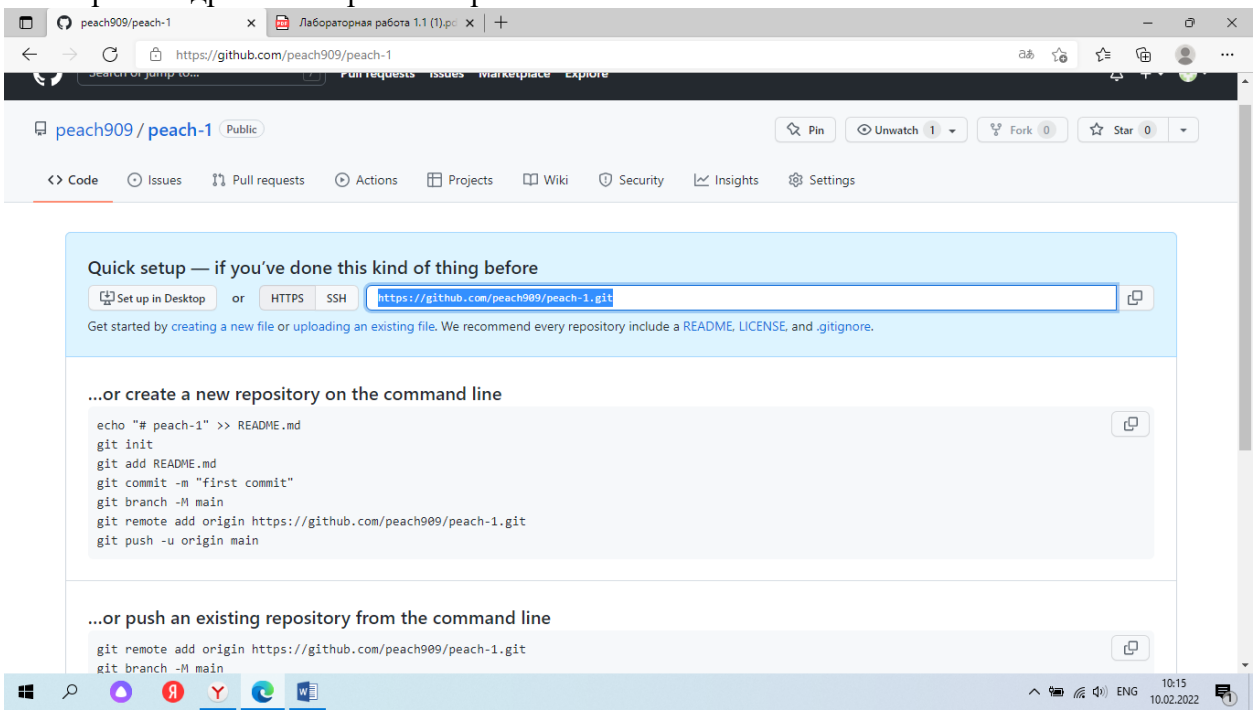
Затем создал новый репозиторий



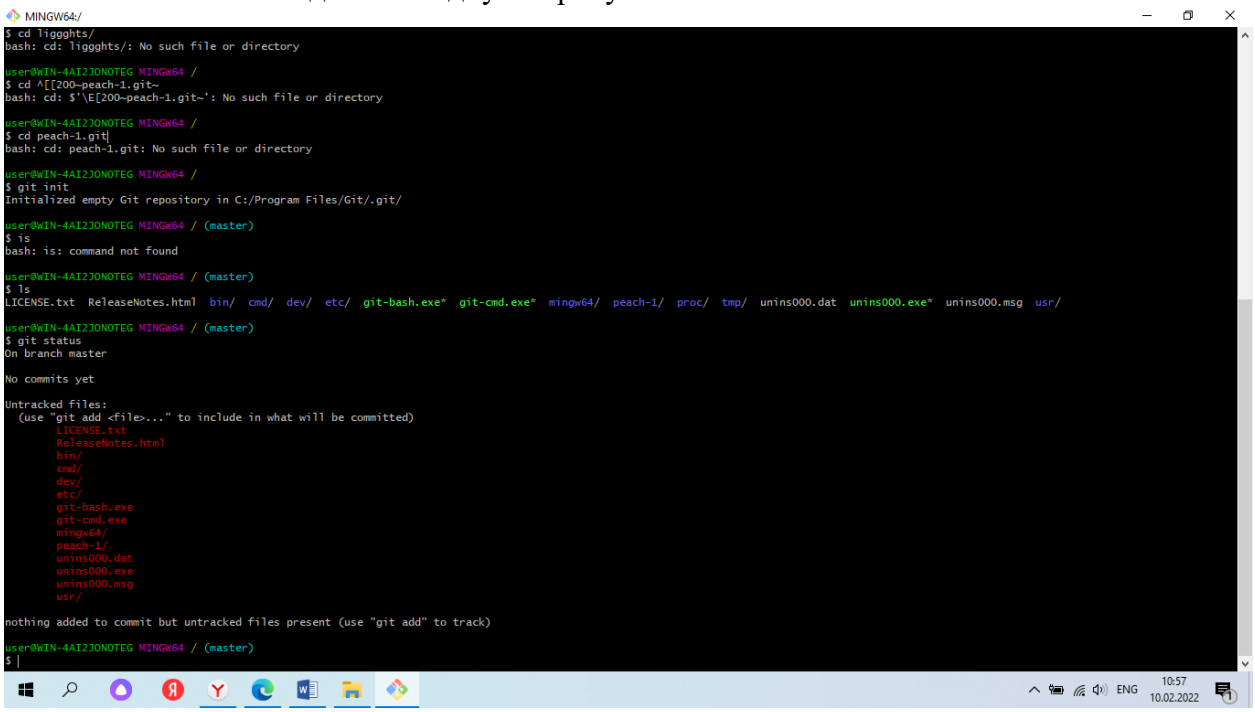
Заполнил настройки моего репозитория:



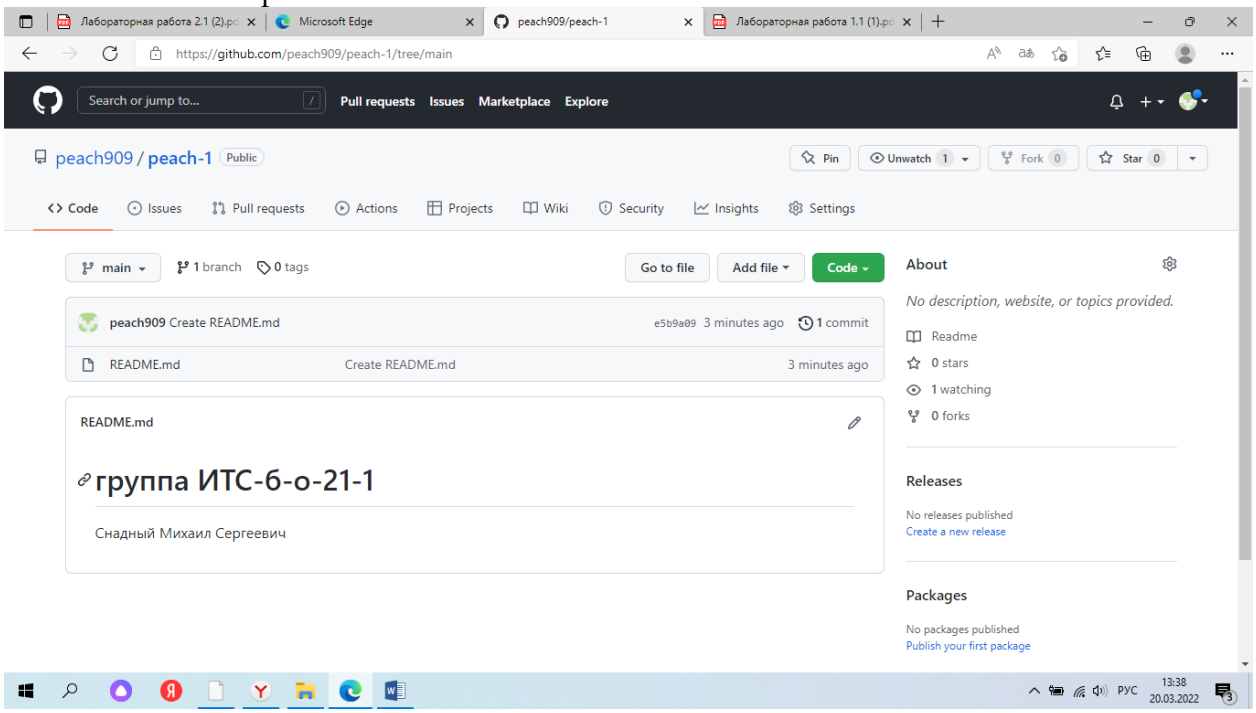
Скопировал адрес моего репозитория



Ввёл несколько команд в командную строку



Внёс изменения в файл README



Выполнил клонирование репозитория

Ответы на контрольные вопросы:

1. Что такое СКВ и каково ее назначение?

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?

Многие люди в качестве метода контроля версий применяют копирование файлов в отдельную директорию (возможно даже, директорию с отметкой по времени, если они достаточно сообразительны). Данный подход очень распространён из-за его простоты, однако он невероятно сильно подвержен появлению ошибок. Следующая серьёзная проблема, с которой сталкиваются люди, — это необходимость взаимодействовать с другими разработчиками.

3. К какой СКВ относится Git?

GIT относится к распределенным системам контроля версий

4. В чем концептуальное отличие Git от других СКВ?

Основное отличие Git от любой другой СКВ (включая Subversion и её собратьев) — это подход к работе со своими данными.

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 находится 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).

7. Что такое профиль пользователя в GitHub?

Профиль - это ваша публичная страница на GitHub

8. Какие бывают репозитории в GitHub?

CSS Protips 30 Seconds of Code A List of Useful Resources for Front-End Developers WTFJS

9. Укажите основные этапы модели работы с GitHub.

Вы изменяете файлы в вашей рабочей директории. 2. Вы выборочно добавляете в индекс только те изменения, которые должны попасть в следующий коммит, добавляя тем самым снимки только этих изменений в область подготовленных файлов. 3. Когда вы делаете коммит, используются файлы из индекса как есть, и этот снимок сохраняется в вашу Git-директорию.

10. Как осуществляется первоначальная настройка Git после установки?

еперь, когда Git установлен в вашей системе, самое время настроить среду для работы с Git под себя. Это нужно сделать только один раз — при обновлении версии Git настройки сохранятся. Но, при необходимости, вы можете поменять их в любой момент, выполнив те же команды снова.

В состав Git входит утилита git config, которая позволяет просматривать и настраивать параметры, контролирующие все аспекты работы Git, а также его внешний вид. Эти параметры могут быть сохранены в трёх местах:

Файл [path]/etc/gitconfig содержит значения, общие для всех пользователей системы и для всех их репозиториев. Если при запуске git config указать параметр --system, то параметры будут читаться и сохраняться именно в этот файл. Так как этот файл является системным, то вам потребуются права суперпользователя для внесения изменений в него.

Файл ~/gitconfig или ~/.config/git/config хранит настройки конкретного пользователя. Этот файл используется при указании параметра --global и применяется ко всем репозиториям, с которыми вы работаете в текущей системе.

Файл config в каталоге Git (т. е. .git/config) репозитория, который вы используете в данный момент, хранит настройки конкретного репозитория. Вы можете заставить Git читать и писать в этот файл с помощью параметра --local, но на самом деле это значение по умолчанию. Неудивительно, что вам нужно находиться где-то в репозитории Git, чтобы эта опция работала правильно.

11. Опишите этапы создания репозитория в GitHub.

для создания Git-репозитория существуют два основных подхода. Первый подход — импорт в Git уже существующего проекта или каталога. Второй — клонирование уже существующего репозитория с
сервера. Создание репозитория в существующем каталоге. Если вы собираетесь начать использовать Git для существующего проекта, то вам необходимо перейти в проектный каталог и в командной строке ввести.

12. Какие типы лицензий поддерживаются GitHub при создании репозитория?

MIT Open Source

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?

Проверить статус можно командой:> **git** status. Результат команды **git** status, если все коммиты проведены и нет новых файлов: # On branch master # Your branch is ahead of 'origin/master' by 2 commits. # nothing to commit (working directory clean). Возможный результат команды **git** status, если имеются изменения.

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 ?

никак

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 .

Перенос репозитория на другой компьютер

После того, как репозиторий был создан на Github, его можно скопировать на любой другой компьютер. Для этого применяется команда:

```
git clone https://github.com/myuser/project.git
```

Результатом выполнения этой команды будет создание папки project в текущем каталоге. Эта папка также будет содержать локальный репозиторий (то есть папку .git).

Так же можно добавить название папки, в которой вы хотите разместить локальный репозиторий.

```
git clone https://github.com/myuser/project.git
```

Работа с одним репозиторием с разных компьютеров

С одним репозиторием с разных компьютеров может работать несколько разработчиков или вы сами, если например работаете над одним и тем же проектом дома и на работе.

Для получения обновлений с удаленного репозитория воспользуйтесь командой:

```
git pull
```

Если вы изменили ваши локальные файлы, то команда git pull выдаст ошибку. Если вы уверены, что хотите перезаписать локальные файлы, файлами из удаленного репозитория то выполните команды:

```
Gitfetch--allgit reset --hard github/master
```

Вместо github подставьте название вашего удаленного репозитория, которое вы зарегистрировали командой `git push -u`.

Как мы уже знаем, для того чтобы изменения выложить на удаленный репозиторий используется команда:

```
git push
```

В случае, если в удаленном репозитории лежат файлы с версией более новой, чем у вас в локальном, то команда git push выдаст ошибку. Если вы уверены, что хотите перезаписать файлы в удаленном репозитории несмотря на конфликт версий, то воспользуйтесь командой:

```
git push -f
```

Иногда возникает необходимость отложить ваши текущие изменения и поработать над файлами, которые находятся в удаленном репозитории. Для этого отложите текущие изменения командой:

```
git stash
```

После выполнения этой команды ваша локальная директория будет содержать файлы такие же, как и при последнем коммите. Вы можете загрузить новые файлы из удаленного репозитория командой git pull и после этого вернуть ваши изменения которые вы отложили командой:

```
git stash pop
```

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

Github Bitbucket Beanstalk Launchpad

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 Fork Tower

Вывод:

Исследовал основные возможности Git и GitHub.