**3. Работа с удаленными репозиториями и GitHub**

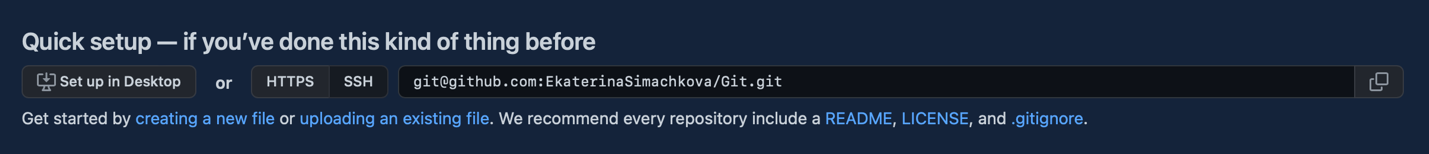
**Цель работы**

Освоить основные навыки работы с облачными и распределенными системами контроля версий, получить навыки работы с инструментальными средствами, обеспечивающими командную работу над разработкой ПО.

**Задания для выполнени**

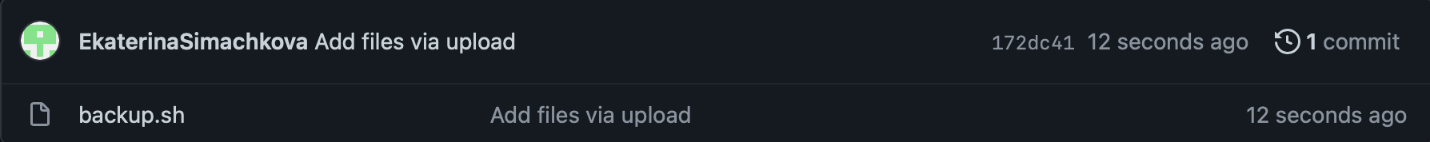
1. Зарегистрироваться на сайте github.com****
2. Установить на компьютере программу GitИзображение выглядит как текст

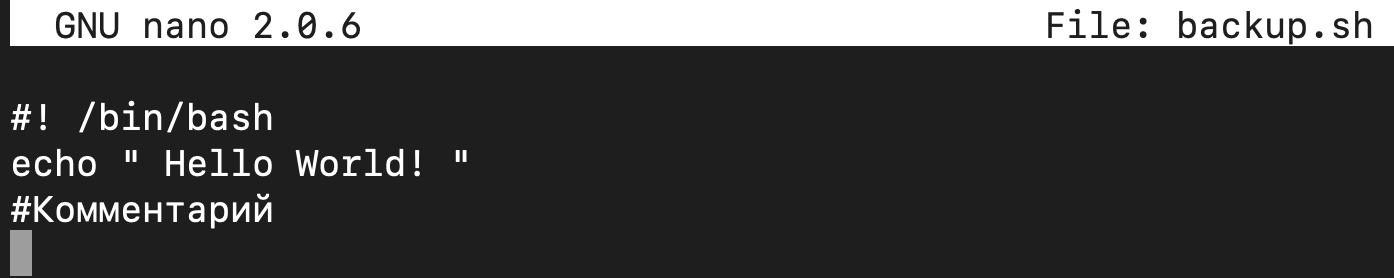
   Автоматически созданное описание
3. Форкнуть данный репозиторий в свой аккаунтИзображение выглядит как текст

   Автоматически созданное описание
4. Склонировать созданный удаленный репозиторий в директорию ~/git/test
5. На локальной машине пишем скрипт ~/git/test/backup.sh, с произвольным содержанием
6. Фиксируем скрипт в репозитории (делаем коммит)Изображение выглядит как текст

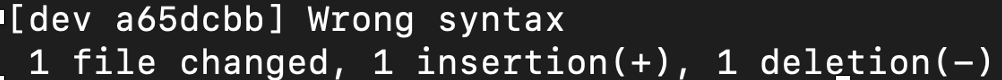
   Автоматически созданное описание
7. Обновляем удаленный репозиторий репозиторий (делаем пуш)Изображение выглядит как текст

   Автоматически созданное описание



1. Через текстовый редактор добавить любую новую строку с комментарием
2. Сделать коммит



1. Внести синтаксическую ошибку в скрип
2. Сделать коммит ошибочного скрипта
3. Откатываем до последней рабочей версии
4. Просмотреть историю коммитов



1. Добавить несколько коммитов произвольного содержимогоИзображение выглядит как текст

   Автоматически созданное описаниеИзображение выглядит как текст

   Автоматически созданное описание
2. Создать пулл реквест в данный репозиторий

**Контрольные вопросы**

1. Зачем нужен облачный хостинг репозиториев?

Для гибкой командной работы программистов.

1. Какими основными функциями обладает сайт github.com?

* Можно создавать приватные репозитории, которые будут видны только вам и выбранным вами людям. Раньше такая возможность была платной.
* Есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса.
* Код проектов можно не только скопировать через Git, но и скачать в виде обычных архивов с сайта.
* С помощью вкладки commits вы можете изучить историю разработки проекта: когда и кем были внесены изменения в какие файлы и строки, как развивался проект в целом. Коммиты (“commit” - фиксировать) - записи изменений в репозиторий.

1. Как организовать командную работу над открытым проектом?

Следует синхронизировать работу нескольких программистов в одном месте. Действует принцип “Распределённого репозитория”. Каждый пользователь может создавать репозитории, ветки и работать с ними. И у каждого из них есть связь с удалённым репозиторием. Таким образом, появляется возможность отправлять изменения на удалённый репозиторий и скачивать оттуда изменения.

**Дополнительные задания**

1. Дополнительно оценивается, если студент продемонстрирует работу с ветками в процессе написания более-сложного программного проекта (не менее трех файлов, двух веток, десяти коммитов, как минимум одно объединение).
2. Дополнительно оценивается демонстрация командной работы. Для этого нужно склонировать репозиторий другому члену команды и коммитить от своего имени. При отправке истории на удаленный сервер (push) на сайте будет отображаться общая история. При скачивании истории с сервера (pull) общая история будет отображаться на локальном компьютере.
3. Настройте работу с git вашей интегрированной среды разработки по выбору. Для работы с python рекомендуется использовать PyCharm. Выполните задания лабораторной работы в IDE используя встроенные средства работы с системами контроля версий.