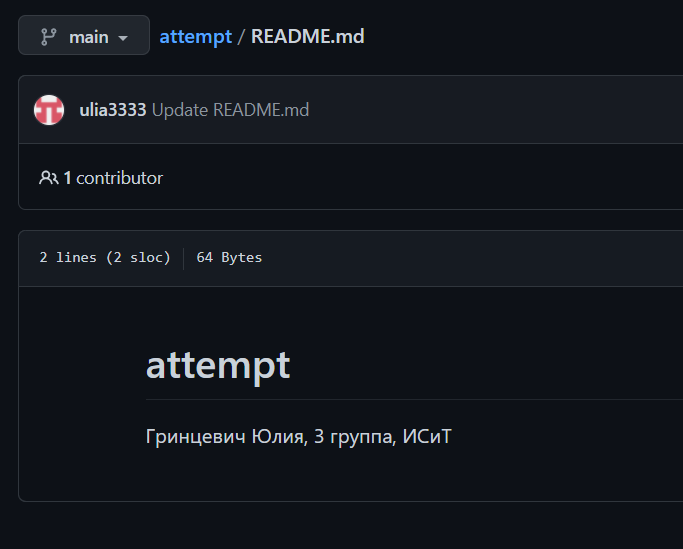
**Лабораторная работа №1**

**Системы контроля верий**

Задание №1

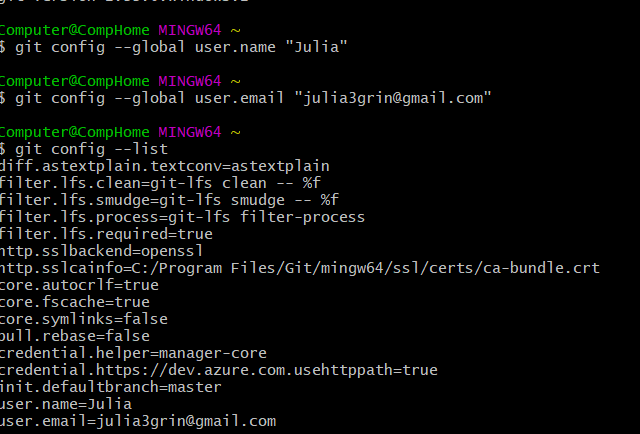


Задание №2 **Базовый уровень. Работа с клиентом GitBush**

Проверка версии

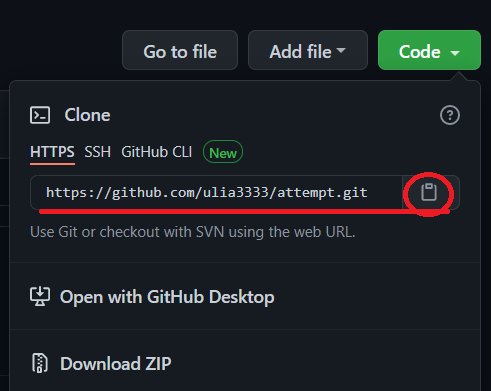


Выполним конфигурацию

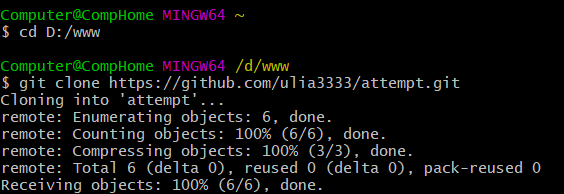


Выполним клонирование недавно созданного нами репозитория репозитория:

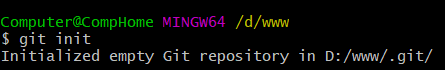
Скопируем ссылку

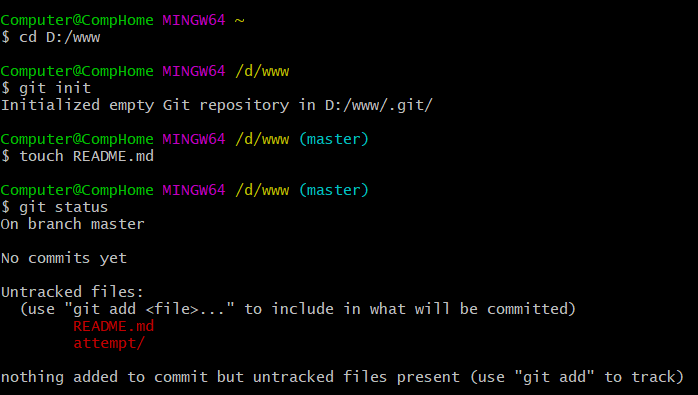


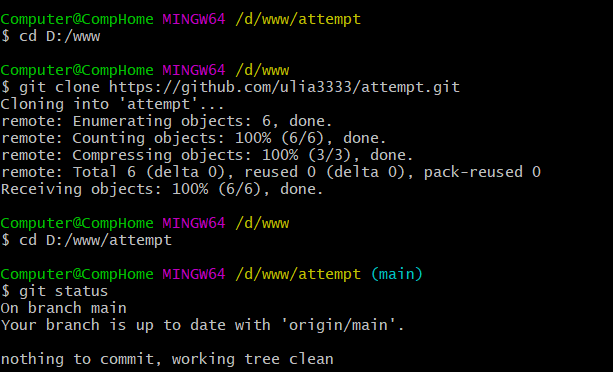
С помощью git clone склонируем в папку www:



С помощью команды git init создаем подкатолог с именем .git, содержащий все необходимые файлы репозитория — структуру Git репозитория. И просмотрим его состояние с ппомощью git status



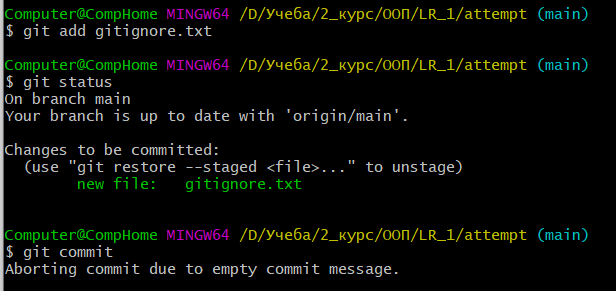


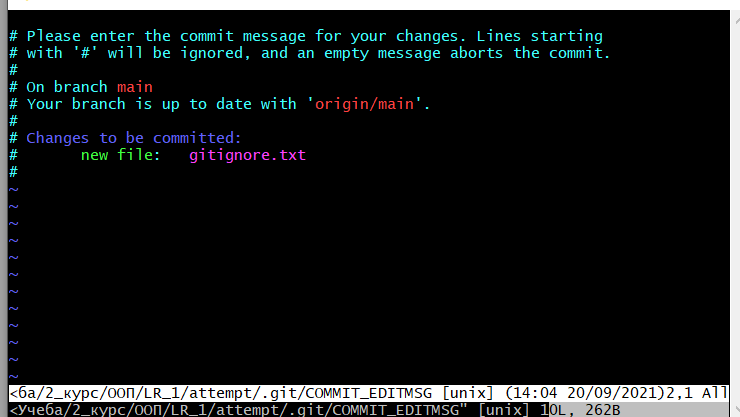


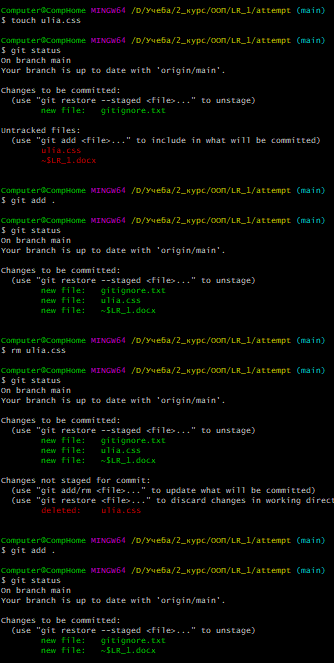
**Настройте gitignore файл для вашего с# репозитория.**

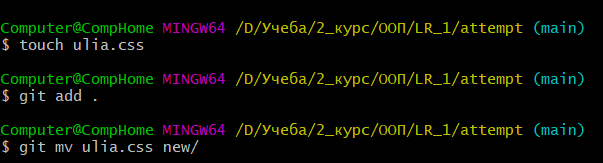
Коллекция шаблонов.gitignore Это коллекция шаблонов файлов .gitignore на GitHub. Мы используем этот список для заполнения шаблонов, доступных в интерфейсе GitHub.com при создании новых репозиториев и файлов..gitignore

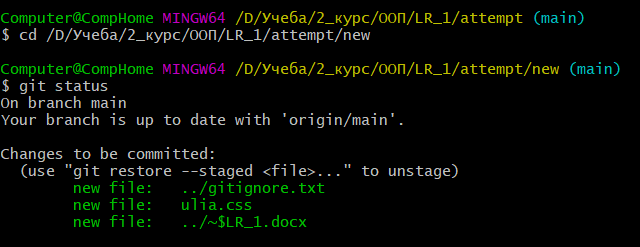
Шаблон должен содержать набор правил, помогающих репозиториям Git работать с определенным языком программирования, фреймворком, инструментом или средой.

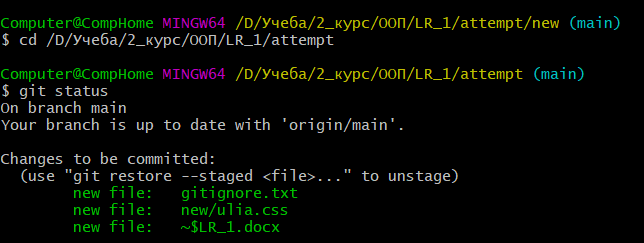






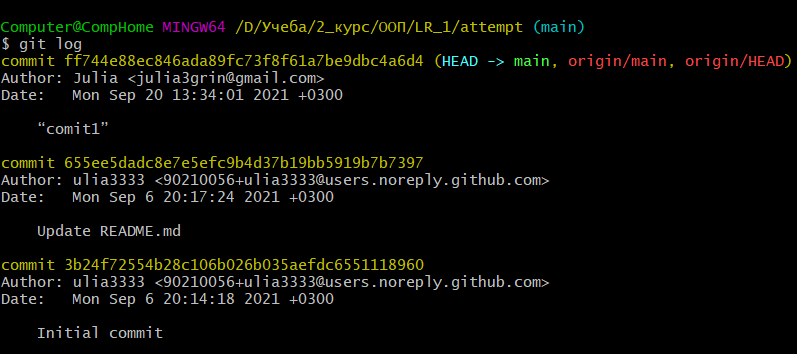






**Изучите команду git log.**

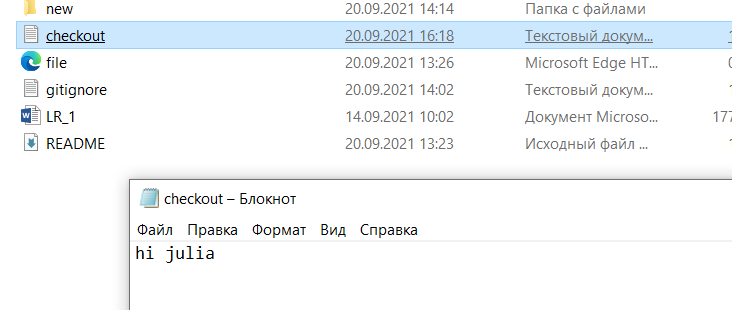
Команда git log выводит в обратном хронологическом порядке список сохраненных в данный репозиторий версий. То есть первыми показываются самые свежие коммиты. Как видите, рядом с каждым коммитом указывается его контрольная сумма SHA-1, имя и электронная почта автора, дата создания и сообщение о фиксации

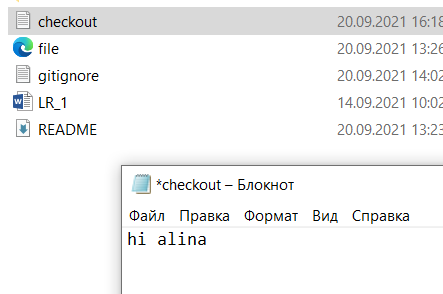


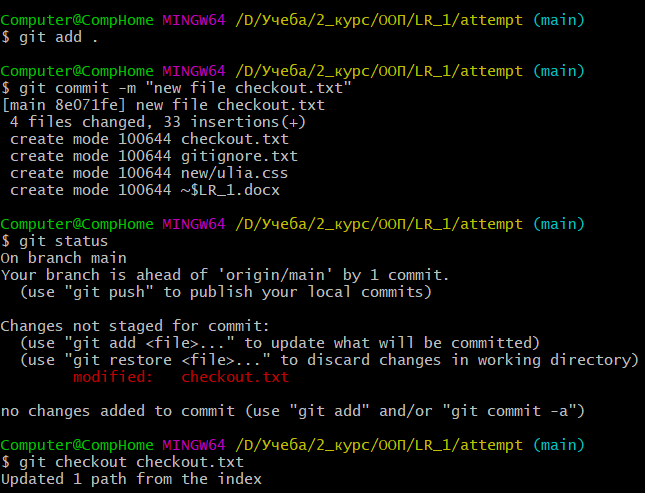
**Сделайте отмену внесенных в файл изменений.**

git checkout -- [file] — опасная команда. Любые изменения соответствующего файла пропадают — вы просто копируете поверх него другой файл. Ни в коем случае не используйте эту команду, если вы не убеждены, что файл вам не нужен.

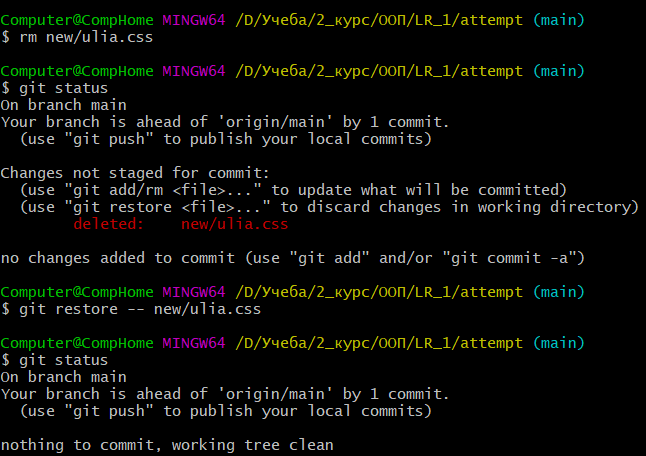
Все, что зафиксировано коммитом в Git, почти всегда можно восстановить. Но все, что вы потеряете, не сделав коммит, скорее всего, вам больше не увидеть

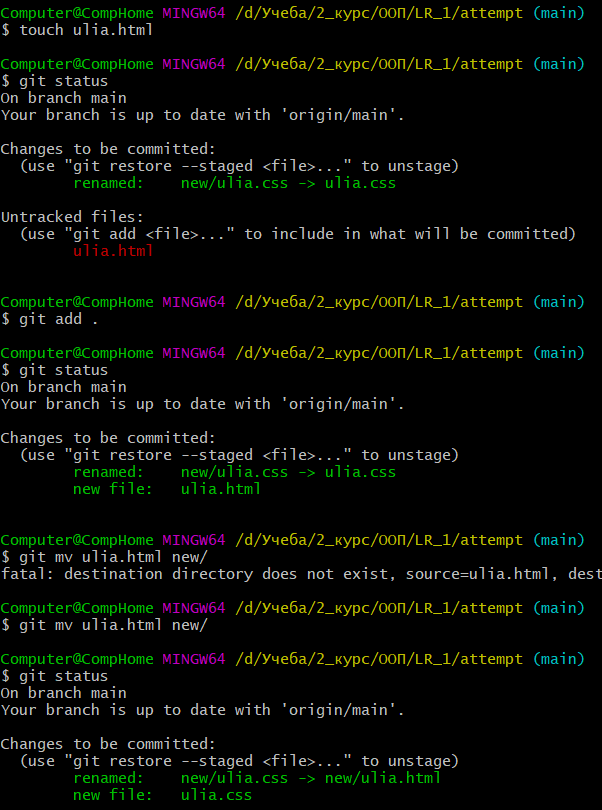






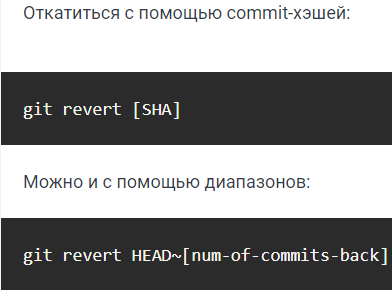
**Изучите команды удаления и пермещения файлов**



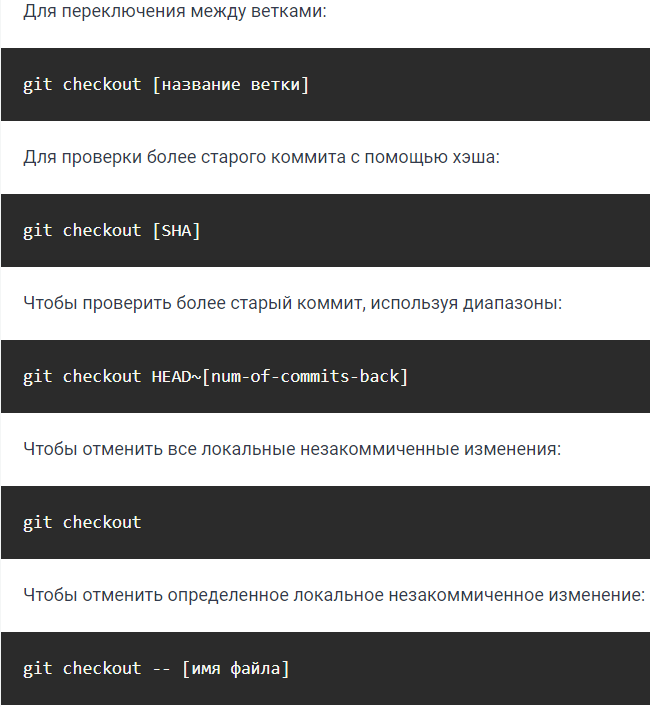


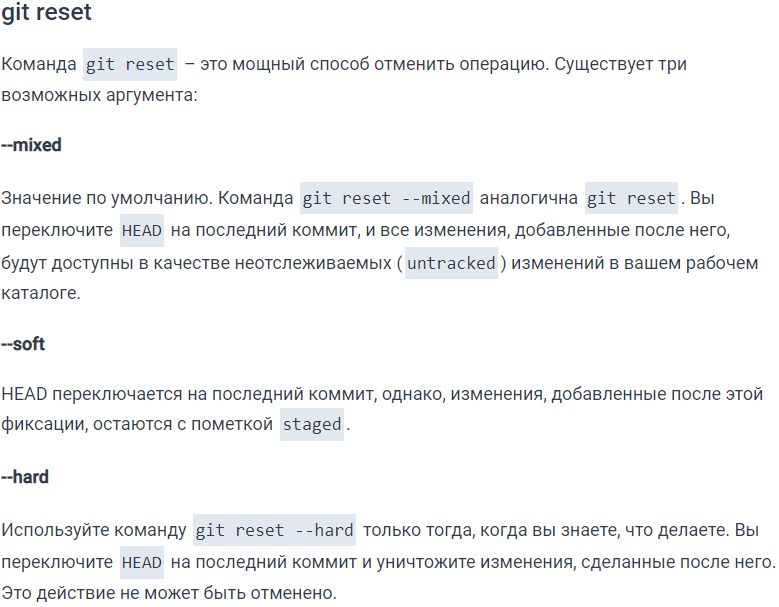
Команды revert, checkout, reset позволяют отменять изменения в репозитории и управлять ими.

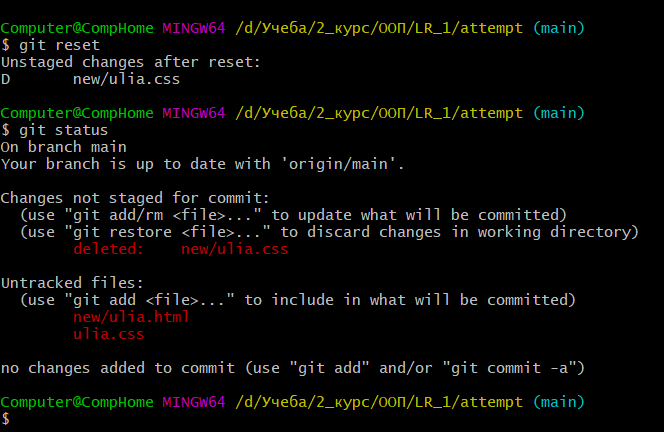
Команда git revert – безопасный способ отменить операцию без потери истории коммитов.



Универсальный инструмент git checkout позволяет переключаться между ветками, проверять старые коммиты и отменять локальные незакоммиченные изменения, переключая HEAD и изменяя рабочий каталог.







**3) Работа с ветками**

Создайте (branch) новую ветку, например, test и переключитесь (checkout) в нее. Добавье новый файл test.json и выполните commit.

Выполните слияние (merge) ветвей.

По очереди отредактируйте файл test.json в ветках master и test.