

## № 1 Системы контроля версий

### Задание

1) **Заведите бесплатную учетную запись** на сайте <https://github.com/>.

Регистрация на сайте заключается в выборе имени пользователя и пароля. Заполните информацию о себе на странице вашего профиля.

Создайте репозиторий для вашего первого проекта (можно загрузить проект по ОАП). Для этого нужно перейти по <https://github.com/new>, выбрать название репозитория (произвольным образом), описание и режим доступа "Public".

Воспользуйтесь разделом "...or create a new repository on the command line" из подсказки, чтобы инициализировать репозиторий и создать в нем файл README.md

Модифицируйте файл README.md так, чтобы он содержал ваше имя, номер группы и специальность.

Ссылку на получившийся репозиторий нужно прислать преподавателю.

### 2) Базовый уровень

Скачайте <http://git-scm.com/download/> и установите консольного (обязательно!!!) клиента (gitbush). Проверьте версию.

Выполните конфигурацию (git config) (настройте имя пользователя и т.п.).

Инициализируйте репозиторий.

Научитесь клонировать репозитории (clone). Например, созданный ранее на <https://github.com>.

Научитесь получать информацию о статусе (status) индексированных и не индексированных файлов. Для этого добавьте в репозиторий новый файл и посмотрите его статус.

Настройте gitignore файл для вашего с# репозитория. Перейдите по ссылке <https://github.com/github/gitignore> и изучите структуру нужного файла.

Выполните команды управления и фиксации изменений (add , commit с добавлением сообщения).

Изучите команды удаления и перемещения файлов.

Изучите команду git log.

### 3) Работа с ветками

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

Выполните слияние (merge) ветвей.

По очереди отредактируйте файл test.json в ветках master и test. Выполните commit. Выполните слияние. Разрешите ситуацию конфликта слияний.

#### **4) Работа с удаленными репозиториями**

Выполните команды загрузки (push) и выгрузки (pull, fetch) в удаленный репозиторий. Добавьте collaborators к вашему репозиторию и попробуйте совместную работу.

**Используйте в папке Литература: Чакон С., Штрауб Б. - Git для профессионального программиста (Библиотека программиста)**

### **Вопросы**

1. Что такое система контроля версий, для чего ее используют?
2. Какой принцип хранения файлов использует Git?
3. В каких трех основных состояниях файлы могут находиться в Git ?
4. Что такое индексация файла? Какой командой она выполняется?
5. Что такое фиксация файла? Какой командой она выполняется?
6. Продемонстрируйте команду проверки выбранных настроек.
7. Как инициализировать репозиторий в существующей папке?
8. Как указать файлы, за которыми должна следить система?
9. Как выполнить фиксацию изменений?
10. Какой командой определить состояния файлов?
11. Для чего создается файл .gitignore? Поясните его структуру.
12. Как используется команда git diff?
13. Как используется команда git commit?
14. Как используется команда git log? Какие у нее есть параметры?
15. Как используется команда git commit –amend?
16. Как отобразить удаленные репозитории?
17. Как извлечь данные из удаленного репозитория?
18. Как отправить данные в удаленный репозиторий?
19. Продемонстрируйте создание новых веток в Git? Что такое ветвление? Что такое указатель HEAD?
20. Как используется команда git checkout?
21. Как выполнить включение изменений из одной ветки в другую?
22. Какие проблемы могут быть при слиянии и как они разрешаются?
23. Что такое GitLab?

