**Лабораторная работа №7 (сценарий)**

**Основы работы с системой контроля версий Git**

**Цель работы**: изучить базовые возможности систем управления версиями на примере управления репозиториями в Git.

**Основные понятия:** система контроля версий, локальный и удаленный репозиторий, поток снимков, фиксация изменений, ветвление, слияние.

**Время выполнения:** 2 часа.

**Теоретические сведения:**

Скачайте msysgit (инсталляционный пакет: <http://git-scm.com/download/win>) и запустите его. Все настройки в инсталляторе можете оставить по умолчанию.

Установите GUI клиент для Git (например, Tortoise Git: <https://tortoisegit.org/download/>). Этот пункт не является необходимым, но графический интерфейс может сильно облегчить управление системой контроля версий Git на платформе ОС Windows. Список доступных клиентов вы можете найти по адресу: <https://git-scm.com/download/gui/win>.

Подробное описание процесса работы с системой контроля версий Git можно найти в книге: <https://git-scm.com/book/ru/v2>. Краткое руководство по командам Git: <http://www.calculate-linux.ru/main/ru/git>. Хорошее руководство по работе с Git на английском: <http://opcode.org/peepcode-git.pdf>.

**Задание:**

1. Если система контроля версий Git не установлена, то установите ее (параметры оставить по умолчанию).
2. Запустите Git GUI или консоль, на ваше усмотрение. Далее подробно рассматривается работа с Git GUI. Создайте новый репозиторий (в папке по вашей фамилии). Для этого щелкните правой кнопки мышки на этой папке и выберете *Git GUI*. Так-как в папке пока что не содержится git хранилища, вам будет показан диалог создания. Выберите пункт *Create New Repository*, заполните путь к вашей новой директории и щёлкните *Create*. Далее вы увидите главный интерфейс Git GUI, который в дальнейшем будет открываться, когда вы будете делать правый щелчок на вашей папке и выбирать *Git GUI*.

Далее вам нужно сообщить системе кто вы такой, дабы в сообщении фиксации (commit message) был отмечен правильный автор. Чтобы сделать это, выберете *Edit → Options* (*Редактировать → Настойки*). В диалоге опций расположены два варианта на выбор. С левой стороны диалога находятся опции, которые влияют только на это хранилище, в то время как правая сторона содержит глобальные опции, применяемые ко всем хранилищам. Значения по умолчанию приемлемы, так что заполните только имя пользователя и email. Если у вас есть любимый шрифт, вы также можете выставить его.

1. Добавьте в папку репозитория файлы. Зафиксируйте состояние репозитория (выполните commit). Для этого нужно сделать следующее. Щелчок на кнопку *Rescan* (*Перечитать*) в Git GUI заставит систему искать новые, измененные и удалённые файлы в директории. Чтобы добавить ваш файл в фиксацию, щёлкните на иконке слева от имени файла. Файл будет перемещён с панели *Unstaged Changes* (*Измененено*) на панель *Staged Changes* (*Подготовлено*). Теперь вы можете добавить сообщение фиксации (commit message) и зафиксировать изменения кнопкой *Commit* (*Сохранить*).
2. Внесите изменения в файлы. Зафиксируйте новое состояние репозитория. Сначала щёлкните *Rescan* (*перечитать*) для поиска изменений. Изменения выделены красным (удаленные линии) или зеленым (добавленные линии). Далее щёлкните на иконке слева от файла чтобы подготовить (*stage*) изменения к фиксации. Файл будет перенесен в нижнее окно. Если вы не хотите фиксировать часть изменений – правый щелчок на нужной строке и выберите *Unstage Line From Commit* (*Убрать строку из подготовленного*). Теперь выбранная строка не будет подготовлена (unstaged) к фиксации, в отличие от остальных изменений. Остается заполнить сообщение фиксации и зафиксировать изменения щёлкнув по *Commit* (*Сохранить*).
3. Создайте новую ветку (например, с именем “testing”). Чтобы создать новую ветку в Git GUI выберете *Branch → Create* (*Ветвь → Создать*), введите имя и щёлкните *Create*. Внесите в ветвь изменения (необходимо добавить новый файл и изменить существующий файл: добавить, удалить и изменить строки) и зафиксируйте их.
4. Переключитесь на ветку мастера. Внесите в нее изменения (добавьте новый файл; измените существующие файлы: добавьте, удалите и измените строки первоначального файла) и зафиксируйте их.
5. Продемонстрируйте слияние веток. Чтобы выполнить слияние, используйте *Merge → Local Merge* (*Слияние → Локальное слияние*). Разрешите возникший конфликт, для этого вы можете использовать любой текстовый редактор (оставляете в тексте только нужный вариант). После разрешения конфликта, подготовьте изменения щёлкнув на иконке файла и зафиксируйте слияние щёлкнув по кнопке *Commit*: в редакторе внесите изменения в файл и сохранить его – *Перечитать → Подготовить → Сохранить*.
6. Покажите дерево изменений веток (историю). История хранилища может быть открыта с помощью *Repository → Visualize All Branch History*.
7. Продемонстрируйте откат изменений в ветке “testing”.

Для отмены внесенных изменений до состояния последней фиксации: *Меню Состояния – Отменить изменения*.

Для отката к конкретной фиксации (reset): Меню *Репозиторий – История ветки* – (выбрать нужное состояние и в контекстном меню выбрать пункт *Установить для ветви это состояние*). Возможны варианты (мягкий – изменение только индекса или жесткий – изменения в индексе и на диске).

Для отката через новую фиксацию – создается новая фиксация, содержащая изменения, обратные зафиксированным (revert): Меню *Репозиторий* – *История ветки* – (выбрать нужное состояние и в контекстном меню выбрать *revert this commit*).

1. Зарегистрируйтесь на сайте <https://github.com/>. Создайте новый репозиторий: после входа на сайт выберите *Create new repository*, укажите его название, тип (публичный) и нажмите *Create repository*. После создания репозитория будет отображено его имя – адрес (в формате HTTPS) и команды для работы с ним в режиме командной строки (для желающих).

Далее из каталога репозитория на локальном ПК вызовите Git GUI, выберите меню *Внешние репозитории → Добавить*, в новом окне укажите псевдоним удаленного репозитория и его адрес. Адрес рекомендуется копировать с сайта.

1. Отправьте данные на удаленный репозиторий. Для этого нужно выбрать в меню *Внешние репозитории → Отправить*, указать псевдоним удаленного репозитория и отправляемую ветку, нажав *Отправить*. При запросе понадобится ввести логин и пароль, с которыми вы регистрировались на сайте.

Если данный этап прошел успешно, то на сайте вы сможете перейти в репозиторий и увидеть его содержимое. Также вы сможете просмотреть историю изменений репозитория, содержимое фиксаций, изменения.

Добавьте к удаленному репозиторию участников проекта. Для этого владелец проекта на сайте должен выбрать *New Collaborator* (сверху вверху), откроется страница участников проектов. На ней нужно указать имя добавляемого участника (он должен быть найден по имени или его части) и нажать *Add Collaborator*, после чего новый участник будет добавлен в список участников. Все участники проекта могут вносить изменения, используя адрес репозитория, свои данные логина и пароля.

1. Получите данные из удаленного репозитория вашего одногруппника. Для получения копии удаленного репозитория нужно открыть проводник, щелкнуть правой кнопкой мыши и из контекстного меню выбрать *Git GUI*. Вам будет предложен диалог создания репозитория, в нем нужно выбрать пункт *Clone* (Клонировать существующий репозиторий). Откроется диалог клонирования, в качестве источника укажите удаленный репозиторий (его адрес), в качестве приемника – новый каталог, нажмите кнопку *склонировать*. Если все успешно, то откроется среда Git GUI, в которой можно посмотреть содержимое файлов и историю изменений. Дальше можно работать над проектами независимо.

1. Измените полученные данные. Зафиксируйте изменения и отправьте их на удаленный репозиторий.
2. Получите обновленные данные из вашего удаленного репозитория. Покажите историю изменений.