**Тема: Использование GitHub Desktop для выполнения основных команд Git**

**1. Введение**

Git — система контроля версий, которая позволяет вести коллективную разработку и отслеживать изменения в программных проектах.  
GitHub Desktop — это официальный графический клиент от GitHub, делающий работу с Git проще: вместо командной строки все действия выполняются интуитивно через интерфейс.

**2. Установка и настройка**

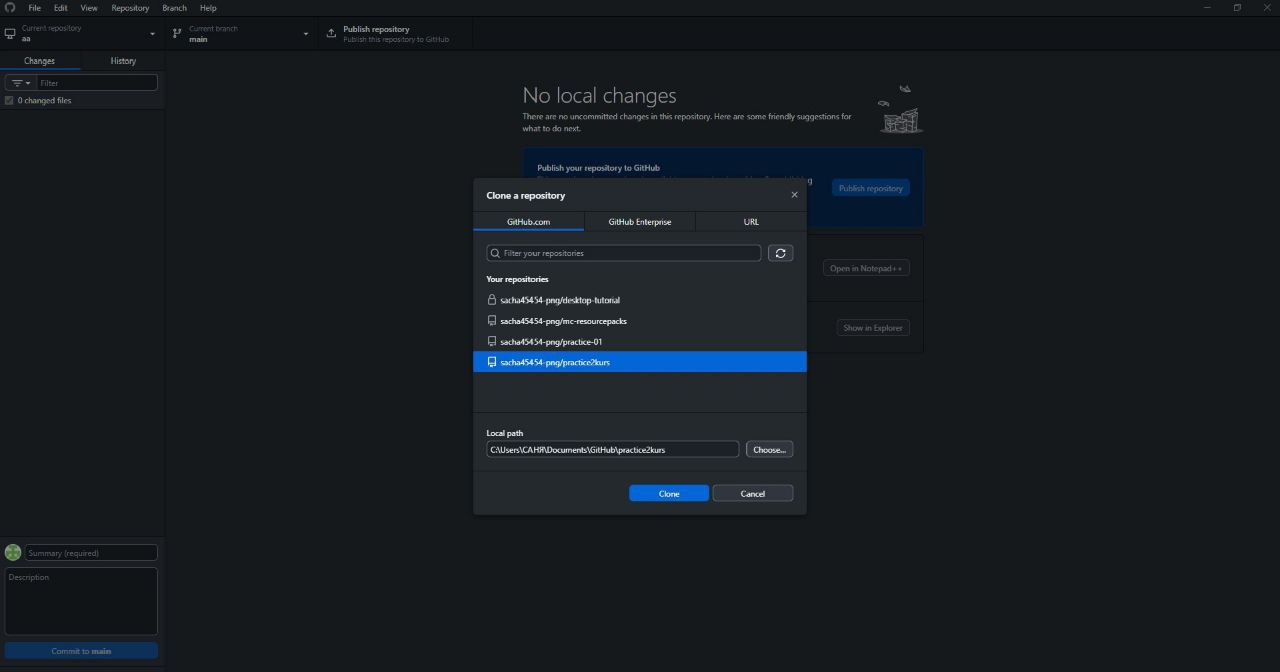
1. Скачан клиент GitHub Desktop с сайта <https://desktop.github.com/>
2. Выполнена авторизация через GitHub-аккаунт.
3. Настроены имя пользователя и email для коммитов.

**3. Основные действия (команды Git)**

3.1 Клонирование репозитория

* В терминале это git clone <url>
* В GitHub Desktop: File → Clone repository… → вставляем ссылку на GitHub → выбираем папку.

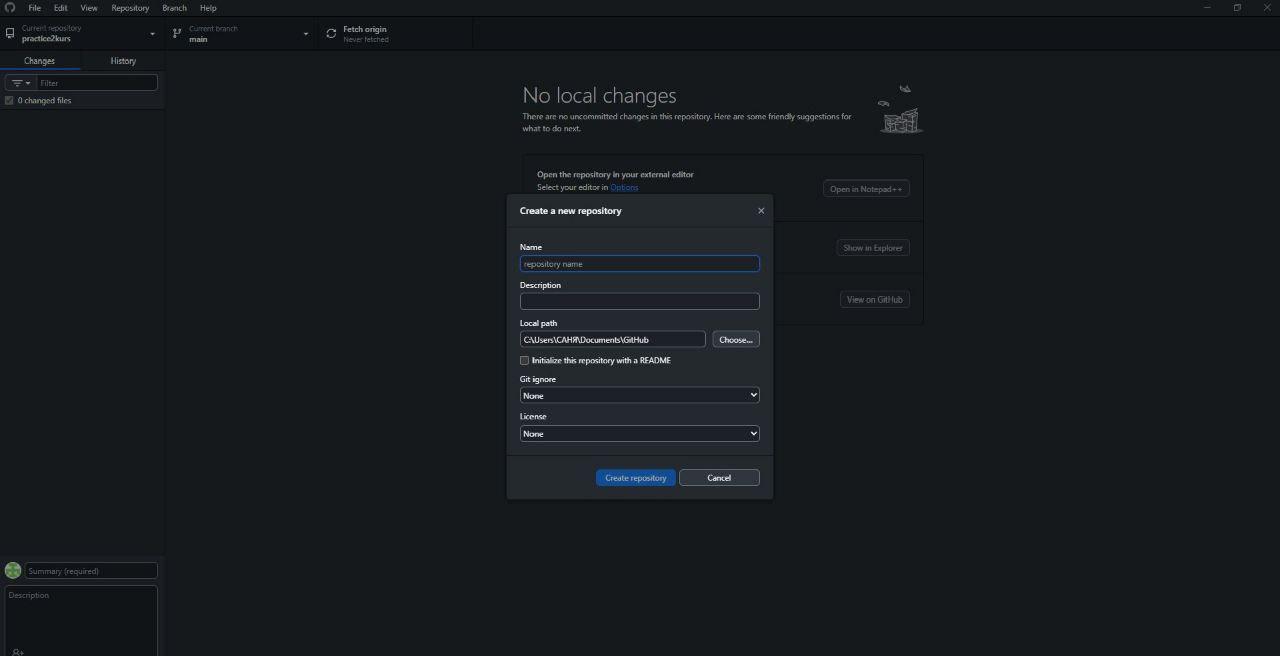
Комментарий: удобно выбирать между «Your repositories» или вставить любую ссылку вручную.



**3.2 Создание нового репозитория**

* В терминале команда: git init
* В GitHub Desktop: File → New repository… → указываем имя проекта, описание и путь.

Особенность: сразу можно сделать репозиторий приватным/публичным, добавить .gitignore и README.



**3.3 Добавление файлов и коммит**

* В терминале:

Bash

git add .

git commit -m "Первый коммит"

* В GitHub Desktop: файлы автоматически отображаются в окне. Достаточно отметить галочками нужные изменения, написать сообщение коммита и нажать Commit to main.

Особенность: визуально чётко видно, какие строки кода были изменены (diff).

**3.4 Отправка изменений в удалённый репозиторий (Push)**

* В терминале: git push origin main
* В GitHub Desktop: кнопка Push origin (появляется в верхней панели).

**3.5 Получение изменений (Pull)**

* В терминале: git pull origin main
* В GitHub Desktop: кнопка Fetch origin → Pull origin, чтобы подтянуть изменения с GitHub.

**3.6 Работа с ветками (Branch)**

* В терминале:

Bash

git branch new-feature

git checkout new-feature

* В GitHub Desktop: сверху нажимаем на выпадающий список веток → New branch… → вводим имя → создаём.

Особенность: переключение происходит мгновенно, видно активную ветку.

**3.7 Pull Request**

Эта функция тесно связана с GitHub:

* После создания ветки и коммитов нажимаем кнопку Create Pull Request.
* Автоматически открывается веб-интерфейс GitHub для обсуждения.

**4. Особенности GitHub Desktop**

Плюсы:

* Очень простой порог входа, подходит для новичков;
* Визуальная работа с изменениями (удобно смотреть diff);
* Быстрая интеграция с GitHub (push/pull/PR в два клика);
* Есть интеграция с GitHub Actions.

Минусы:

* Ограниченный функционал (например, нет тонких настроек rebase и merge конфликтов, как в Sourcetree или GitKraken);
* Привязан в первую очередь к GitHub (с GitLab/Bitbucket нужно больше танцев).

**5. Заключение**

В ходе исследования были изучены и продемонстрированы основные функции GitHub Desktop: клонирование, создание репозитория, добавление файлов, коммиты, push/pull, работа с ветками и pull request.  
Этот инструмент удобен для студентов и начинающих разработчиков благодаря наглядности интерфейса, но для продвинутой работы с Git в дальнейшем рекомендуется изучить консольные команды или более мощные клиенты (Sourcetree, GitKraken).