

Простые операции с git

Оглавление

Общая информация	1
Скачивание проекта с github (клонирование репозитория)	2
Шаг 1. Поиск адреса репозитория	2
Шаг 2. Клонирование (скачивание) репозитория	6
Сохранение работы (фиксация изменений, commit)	8
Загрузка на github (проталкивание изменений, push).....	13
Авторизация на github.....	15
Использование логина и пароля	16
Обновление папки на рабочем столе (затягивание изменений из github, pull).....	20

Общая информация

Для работы по руководству требуется учетная запись на сайте <https://github.com> .
Изображения актуальны на 17 декабря 2023г.

Руководство было подготовлено с использованием программ следующих версий.

Название программы	Версия	Адрес для скачивания
Git	2.42.0	https://git-scm.com
TortoiseGIT	2.15.0	https://tortoisegit.org/download/

Скачивание проекта с github (клонирование репозитория)

Клонировать репозиторий нужно в случаях:

- Вы первый раз работаете на данном компьютере
- Вы запутались при использовании git в старом репозитории и хотите попробовать заново

При этом вы создадите локальный репозиторий.

Шаг 1. Поиск адреса репозитория

Адрес своего репозитория

Откройте ссылку <https://github.com> в веб-браузере.

Перейдите по ссылке «Sign in». На картинке обведено **красным**.

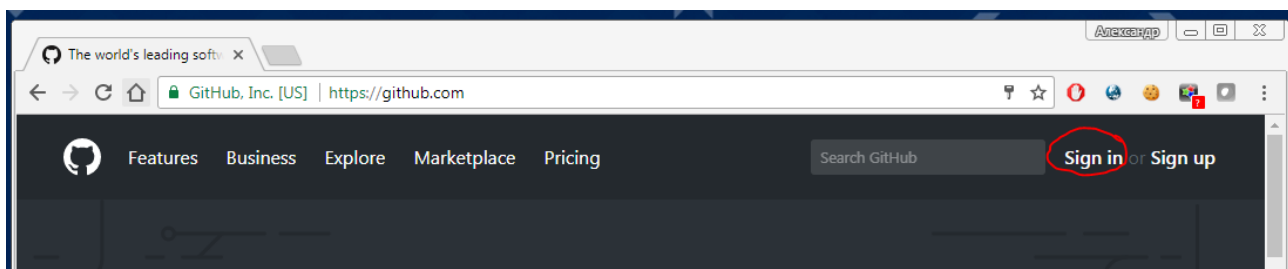


Рисунок 1.1 Расположение ссылки «Sign in».

Введите имя пользователя или email, указанный при регистрации, и пароль.

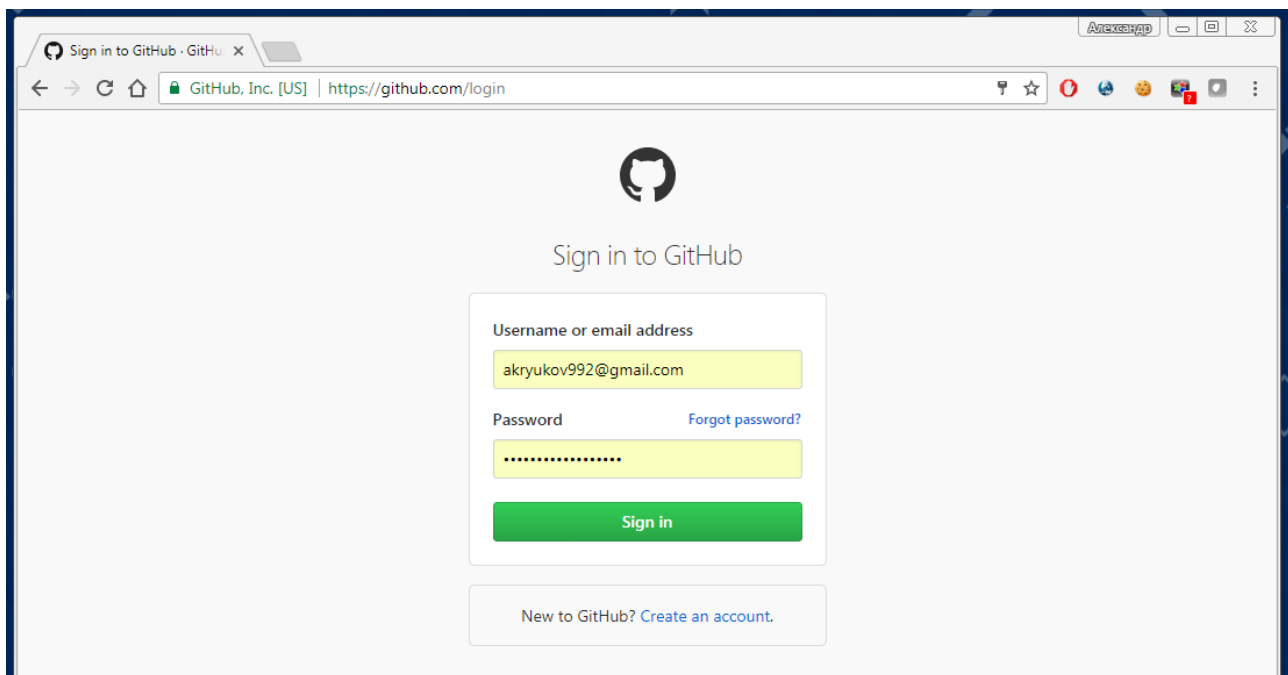


Рисунок 1.2 Форма для входа

Нажмите на кнопку «Sign in». Если пароль введен верно, то откроется главная страница, изображенная на рисунке 1.3.

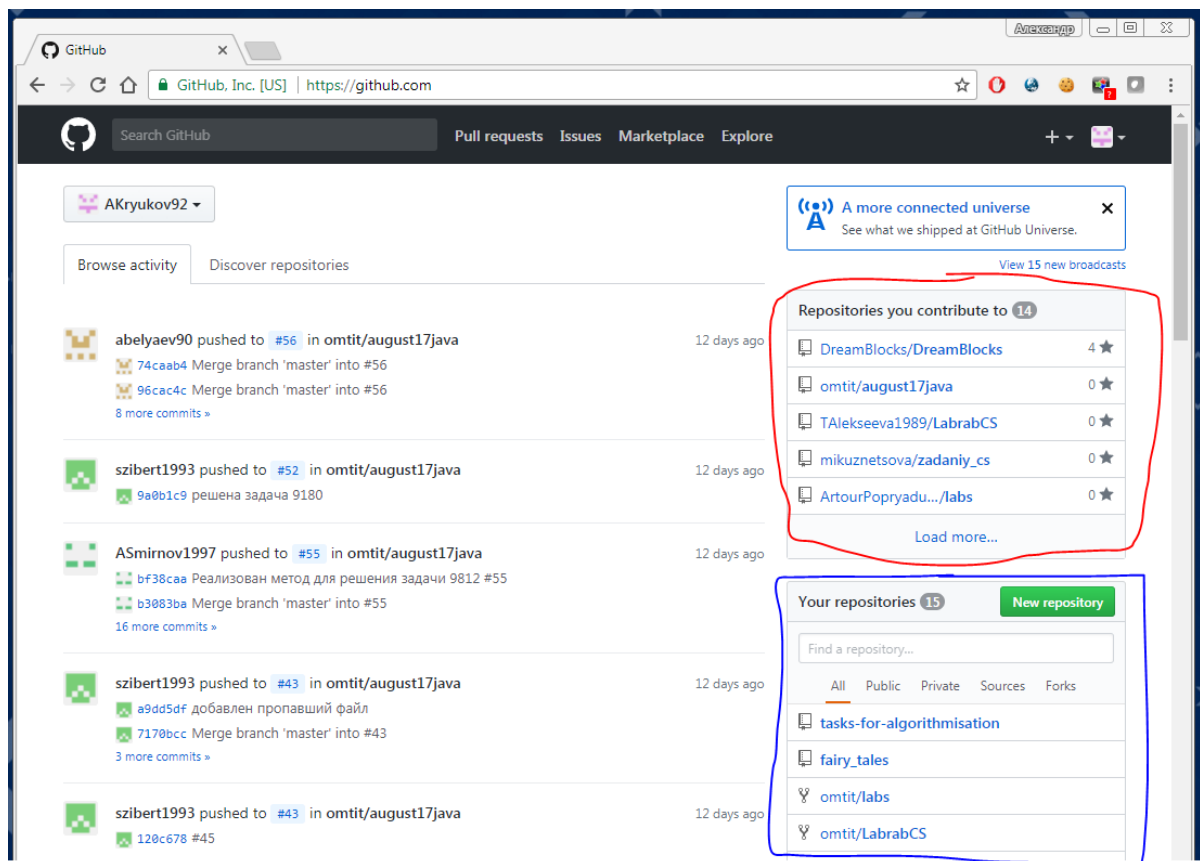


Рисунок 1.3 Главная страница github.com

На рисунке 1.3 **красным** обведен список ссылок на репозитории, в которых вы что-то меняли. **Синим** обведен список ссылок на репозитории, которые вы создали сами.

Внизу каждого списка есть кнопка «Load more...», которая откроет список полностью.

Найдите в этих списках ссылку на нужный репозиторий.

Адрес репозитория другого пользователя

Откройте ссылку <https://github.com> в веб-браузере.

Найдите строку поиска. Если вы еще не авторизовались на сайте, то поиск будет справа (рис. 1.4).

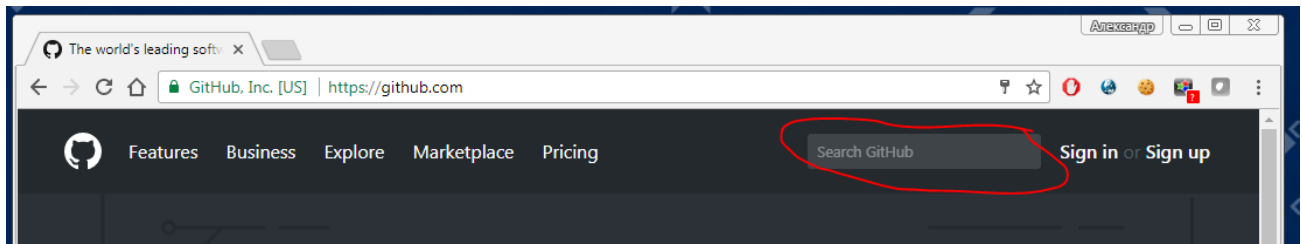


Рисунок 1.4 Расположение формы поиска для неавторизованного пользователя

Если уже авторизовались – слева (рис. 1.5).

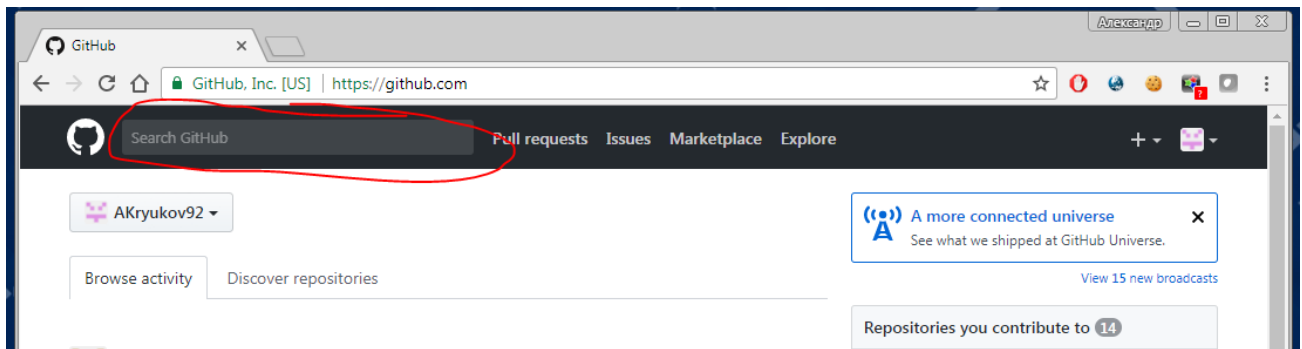


Рисунок 1.5 Расположение формы поиска для авторизованного пользователя

В строке поиска введите имя пользователя и нажмите «Enter». Результат поиска изображен на рисунке 1.6.

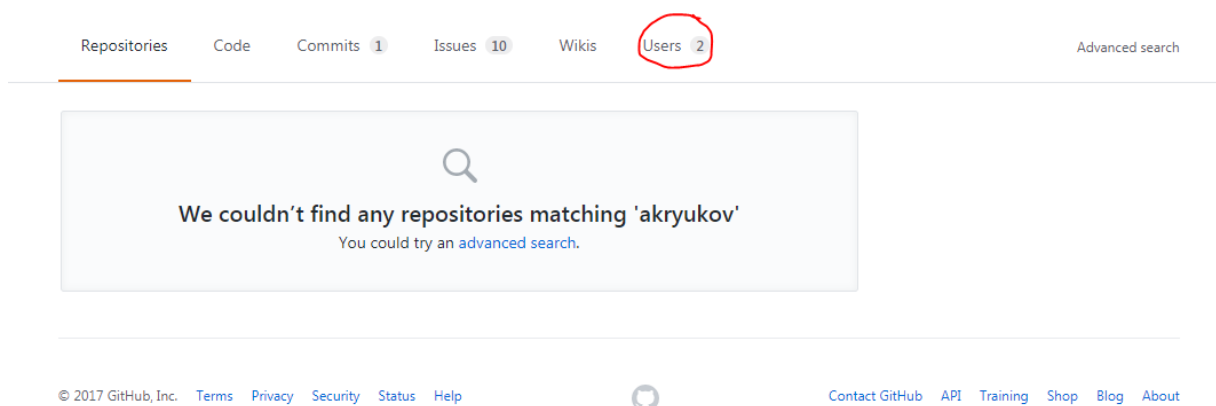


Рисунок 1.6 Результаты поиска

Перейдите на вкладку «Users» (рис. 1.7).

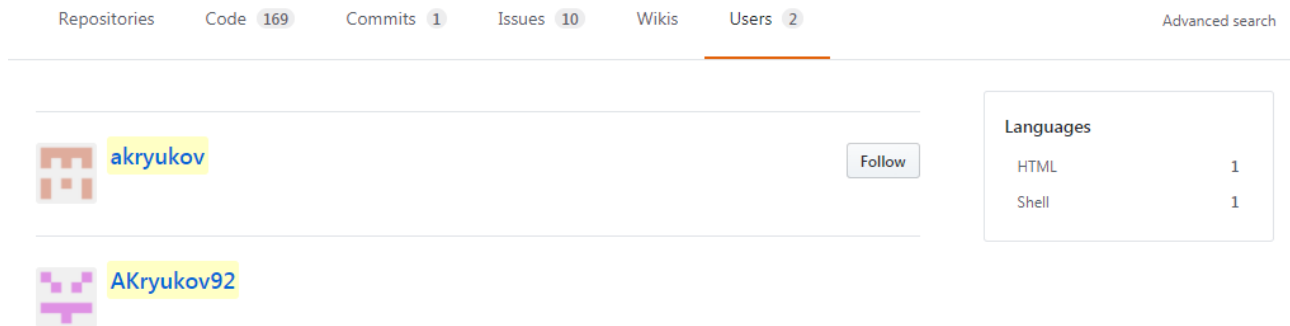


Рисунок 1.7 Вкладка «Users»

Во вкладке «Users» (рис.1.7) нажмите на имени нужного пользователя, чтобы открыть его профиль (рис. 1.8).

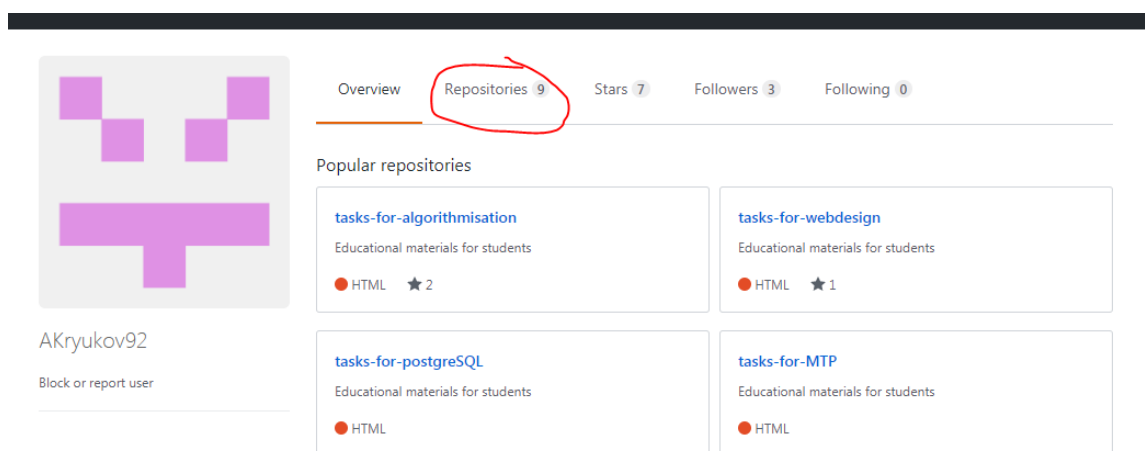


Рисунок 1.8 Профиль пользователя

В профиле пользователя нажмите на ссылку «Repositories». Она отмечена красным на рисунке 1.8. Откроется полный список репозиторий пользователя (рис.

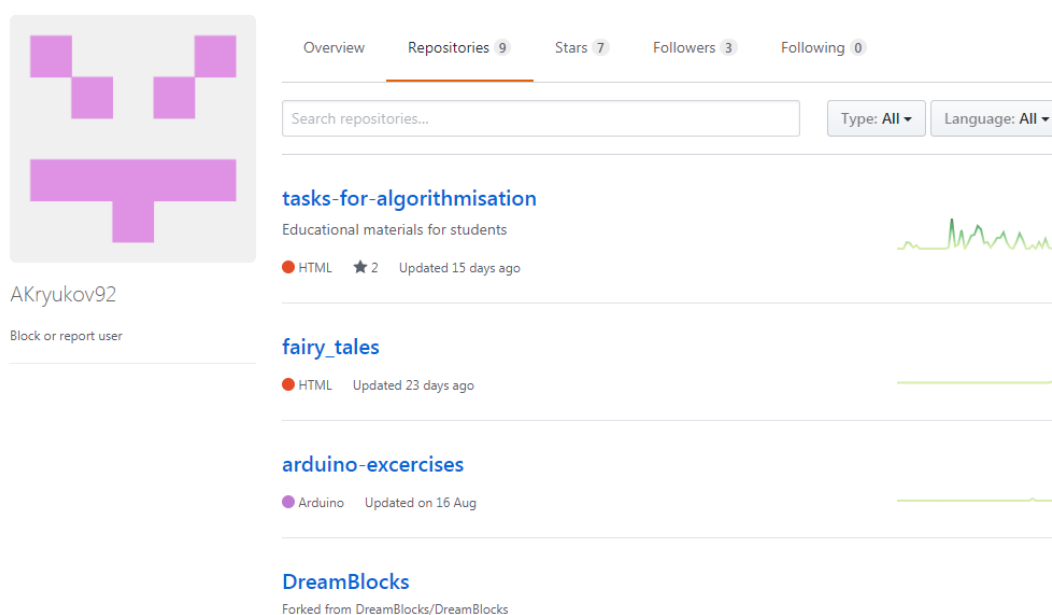


Рисунок 1.9. Вкладка «Repositories»

В появившемся списке нажмите ссылку на нужный репозиторий.

Шаг 2. Клонирование (скачивание) репозитория

При нажатии на названии репозитория в каком-либо списке, откроется страница этого репозитория. Пример на рисунке 2.1. В примере репозиторий называется «fairy_tales».

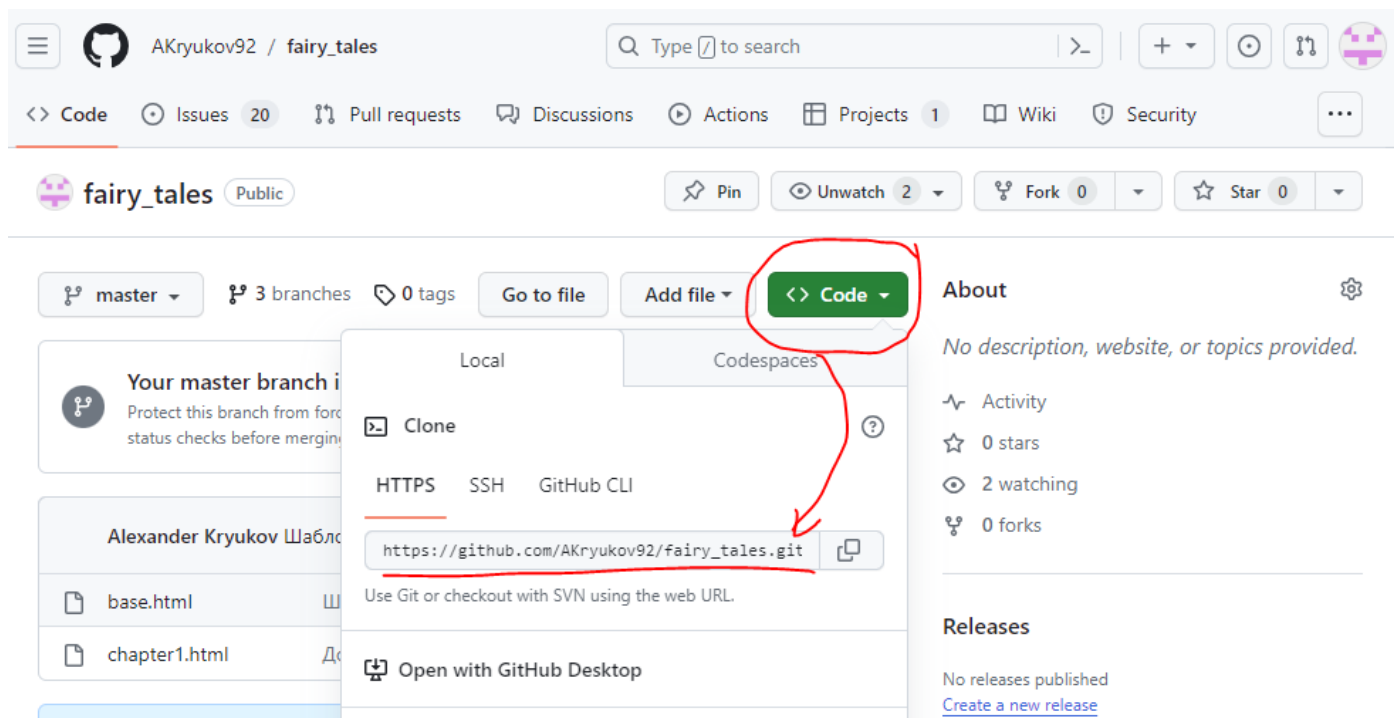


Рисунок 2.1 Страница репозитория

На странице репозитория нажмите на кнопку «Code». В появившемся меню, нужно копировать адрес, который выглядит примерно так:

https://github.com/AKryukov92/fairy_tales.git

На рабочем столе:

1. Удалите или переименуйте папку, название которой совпадает с названием репозитория. В данном примере папка называется «fairy_tales».
2. Нажмите правую кнопку мыши на свободном месте рабочего стола и выберите пункт меню «Git Clone...».

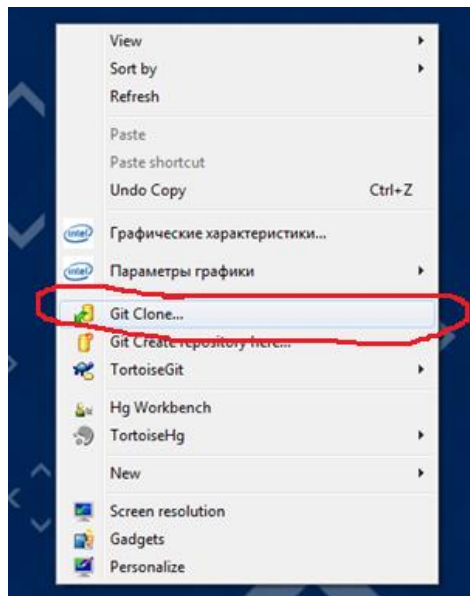
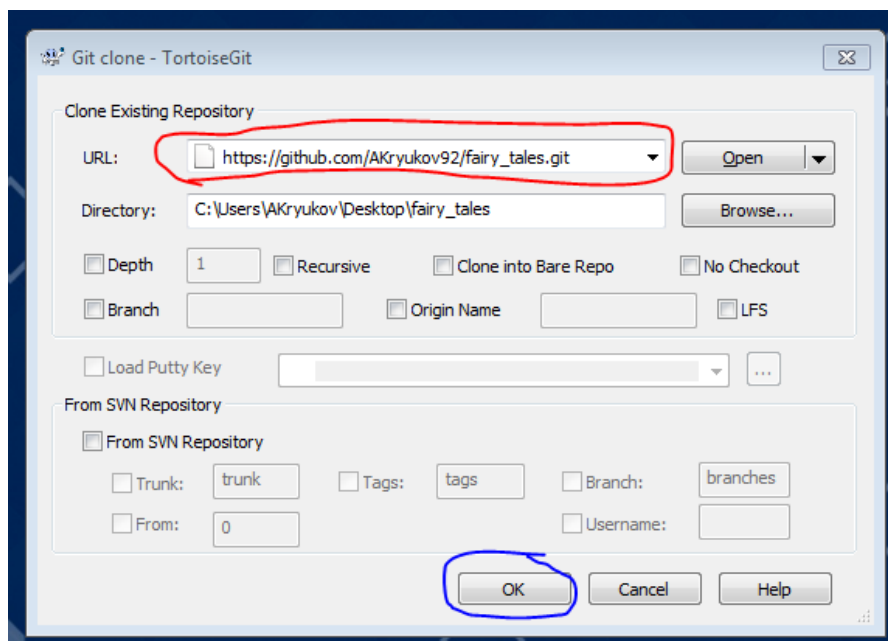


Рисунок 2.2 Пункт меню «Git Clone...»

При этом должен появиться диалог клонирования репозитория (рис. 2.3). В нем:

1. Проверьте что в поле URL (обведено красным) находится адрес репозитория, который вы хотите клонировать.
2. Нажмите «OK» (обведено синим).



Рисуно 2.3 Диалог клонирования репозитория

Должно появиться окно, отражающее процесс клонирования как на рисунке 2.4.

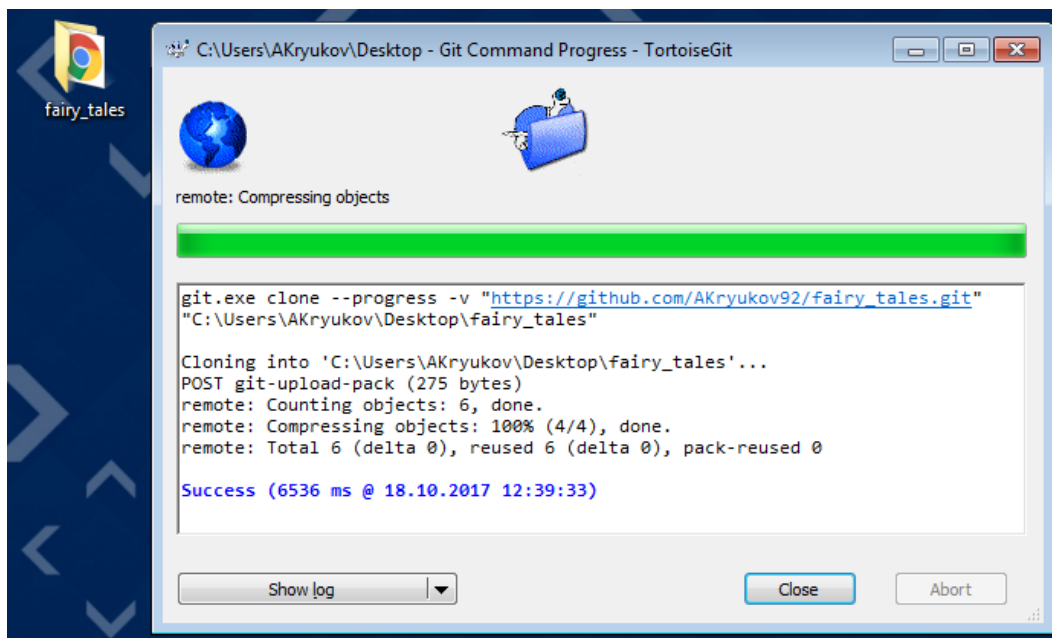


Рисунок 2.4 Окно процесса клонирования

Все прошло успешно, если в текстовом поле этого окна в конце написано “**Success**”. Если надпись **красная**, то нужно звать на помощь более опытных пользователей git.

На рабочем столе должна появиться папка – локальный репозиторий, название которой совпадает с названием репозитория на github.

Сохранение работы (фиксация изменений, commit)

Сохранять работу можно после того, как в папке на рабочем столе (в локальном репозитории) были добавлены или изменены какие-либо файлы.

Сохранять можно любые файлы, включая:

- Файлы с кодом “Program.cs”, “task0000.java”, “task0000.html”.
- Файлы MS Word “Письменные задания к ЛР1.docx”.
- Файлы конфигурации “lab01.sln”, “task0000.csproj”, “checkstyle-idea.xml” и т.п.

Пустые папки не сохраняются.

Если файл уже был сохранен в репозитории и вы его изменили, то он будет отмечен красным кругом с восклицательным знаком. Все добавленные файлы отмечаются синим кругом с вопросительным знаком.

На рисунке 3.1 изображен репозиторий «fairy_tales». В нем файл «base.html» был изменен, а файл «chapter1.html» был добавлен.

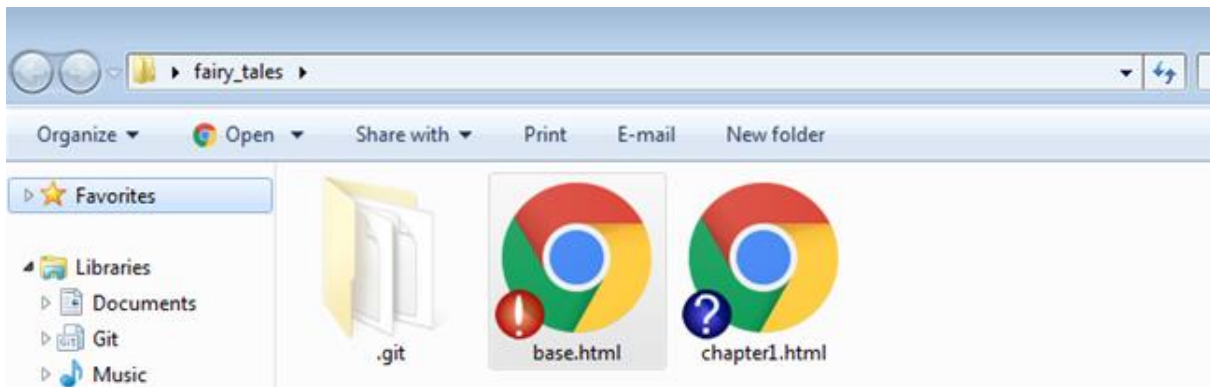


Рисунок 3.1 Внешний вид измененных и добавленных файлов

Нажмите правую кнопку мыши на папке, а затем выберите пункт меню «Git Commit -> “master”...».

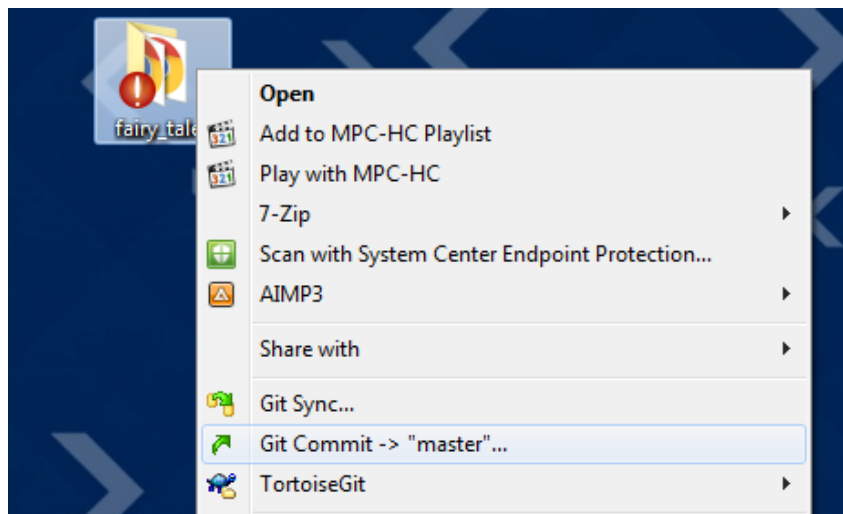


Рисунок 3.2 Пункт меню «Git Commit->”master”» на рабочем столе

Другой вариант – нажмите правую кнопку мыши на свободном месте при открытии папки-локального репозитория в проводнике (рис. 3.3).

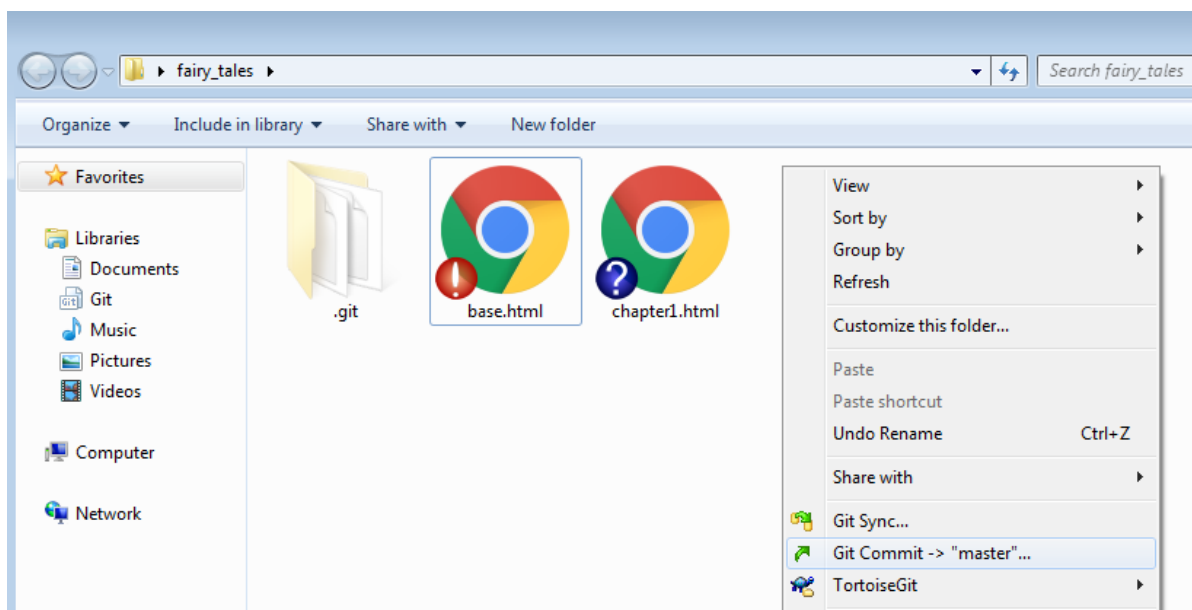


Рисунок 3.3 Пункт меню «Git Commit->”master”» в репозитории

Если вы делаете первый коммит в репозиторий, то у вас может появиться окно как на рисунке 3.4.

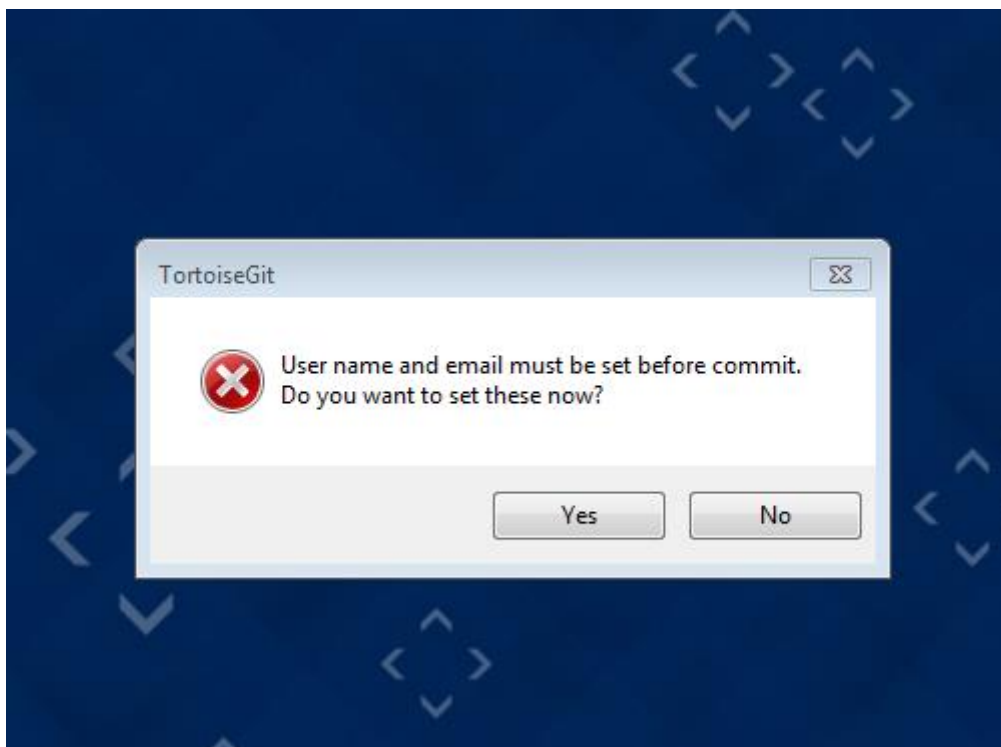


Рисунок 3.4 Диалоговое окно для уточнения автора изменений

Вам нужно уточнить имя пользователя и email. Эта информация используется для обозначения автора изменений при фиксации.

Нажмите на «yes». При этом должно открыться окно с настройками TortoiseGIT.

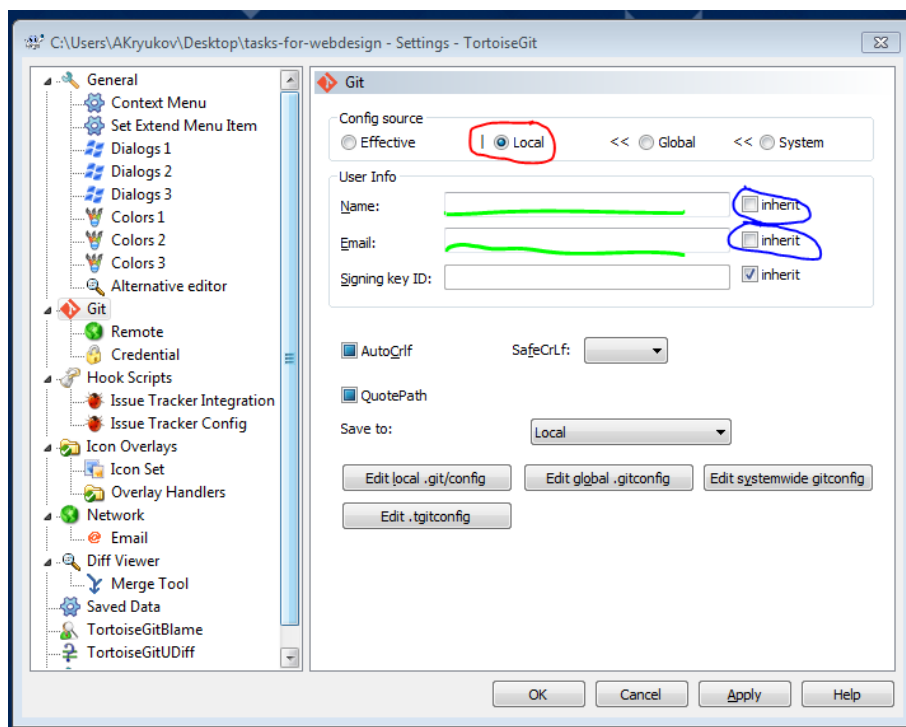


Рисунок 3.5 Окно настроек TortoiseGIT

В окне настроек:

1. Выберите переключатель «Local». На рисунке он отмечен **красным**.
2. Снимите отметки с галочек «inherit» напротив полей «Name» и «Email». Они отмечены **синим**.
3. Укажите имя пользователя на гитхабе в поле «Name» и адрес почты (не пароль) в поле «Email». Эти поля отмечены **зеленым**.

При этом откроется окно коммита, которое изображено на рисунке 3.6.

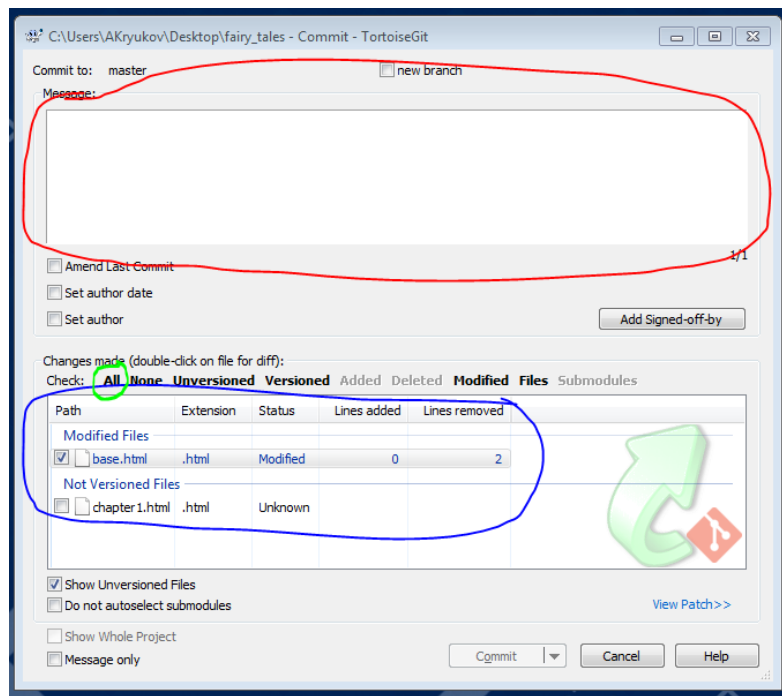


Рисунок 3.6 Окно коммита

В окне коммита:

1. Введите описание изменений в области, отмеченной **красным** на рисунке 3.6. Туда нужно писать номера заданий и лабораторных, которые были выполнены с момента последнего сохранения.
2. Отметьте файлы, которые нужно сохранить в списке файлов. На рисунке 3.6 он отмечен **синим**. Если вам нужны все файлы, а их много, тогда нажмите на слово “all”, которое на рисунке 3.6 обведено **зеленым**. При этом галочки проставятся напротив всех файлов.

Нажмите на кнопку «Commit». На рисунке 3.7 она отмечена **красным**.

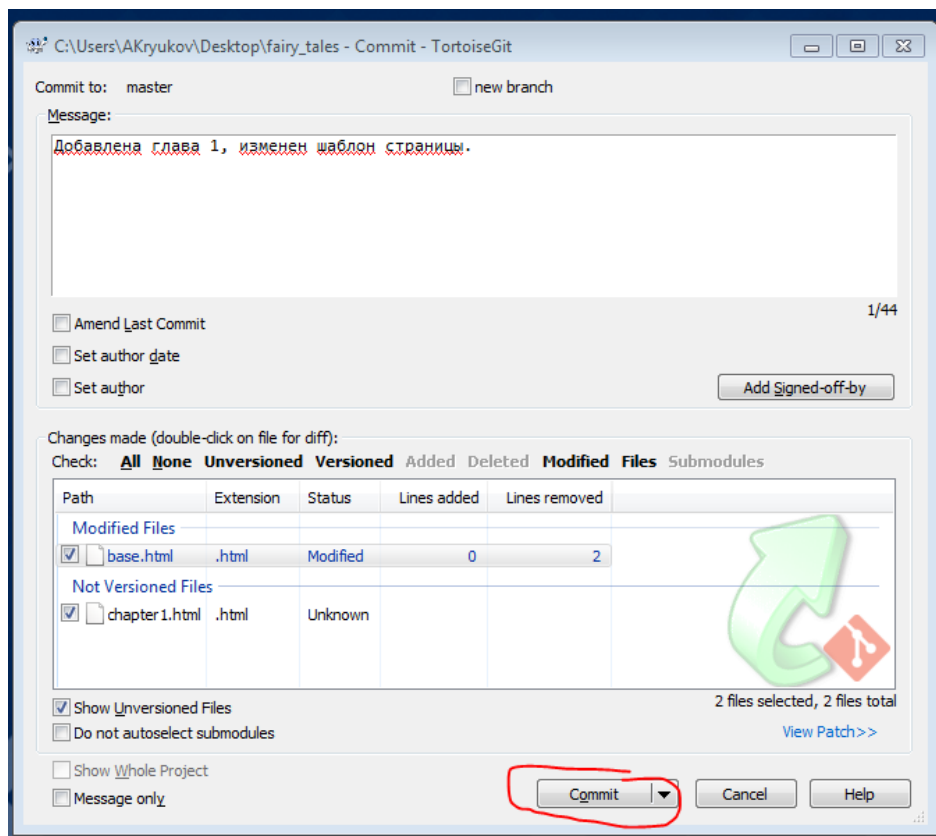


Рисунок 3.7 Расположение кнопки «Commit»

Появится окно (рис. 3.8), отражающее процесс сохранения (фиксации) изменений.

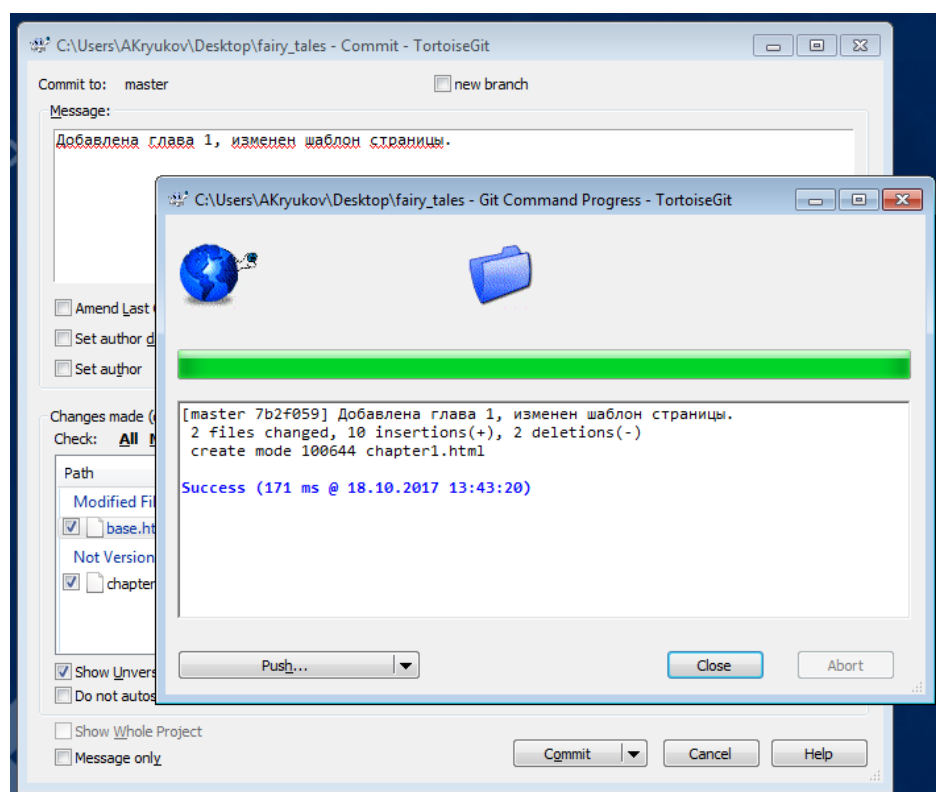


Рисунок 3.8 Окно процесса фиксации изменений

После всех этих действий, изменения будут сохранены только в локальном репозитории. На github они еще не загружены. Инструкция по загрузке на github находится ниже.

Загрузка на github (проталкивание изменений, push)

Загружать на github можно только те изменения, которые вы сохранили (зафиксировали, сделали commit).

Если просто положить файл в папку-локальный репозиторий, то он не будет загружаться на github.

Пустые папки не будут загружаться на github.

Если вы сделали commit, но не отметили файл галочкой, то он не будет загружаться на github.

Чтобы загрузить изменения на github:

1. Нажмите правую кнопку мыши на папке-локальном репозитории.
2. Выберите меню TortoiseGit .
3. Нажмите Push (см. рис. 4.1).

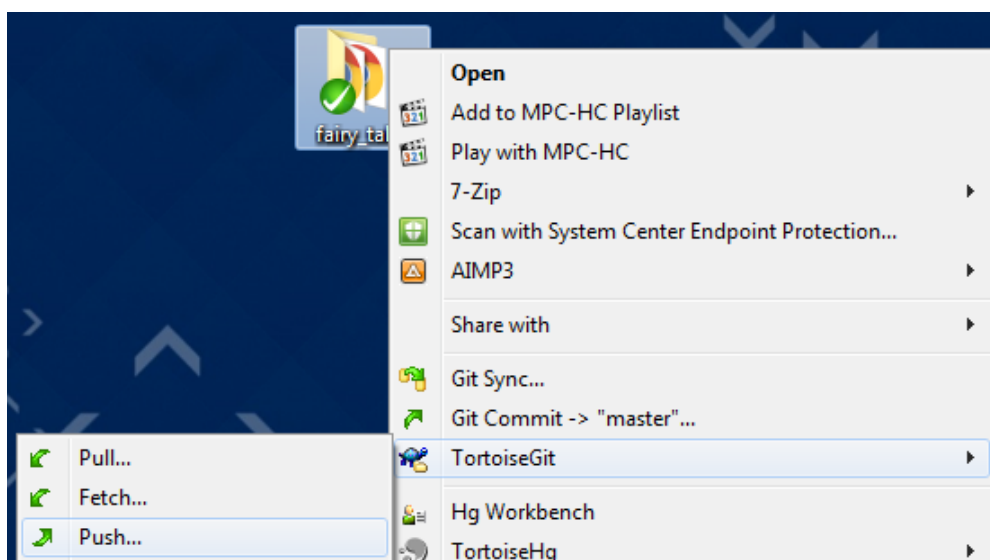


Рисунок 4.1 Пункт меню «TortoiseGit», «Push» на рабочем столе

Другой вариант – нажмите правую кнопку мыши на свободном месте при открытии папки-локального репозитория в проводнике (см. рис. 4.2).

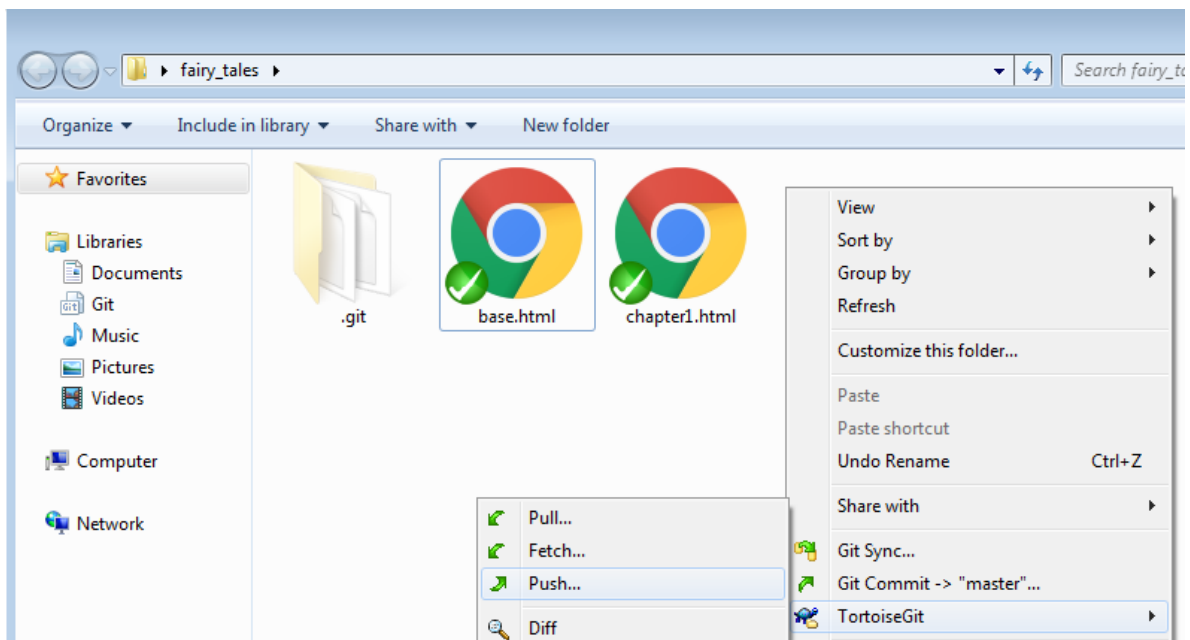


Рисунок 4.2 Пункт меню «TortoiseGit», «Push» в локальном репозитории

Третий вариант – нажмите на кнопку “Push” в окне процесса фиксации изменений (рис. 4.3)

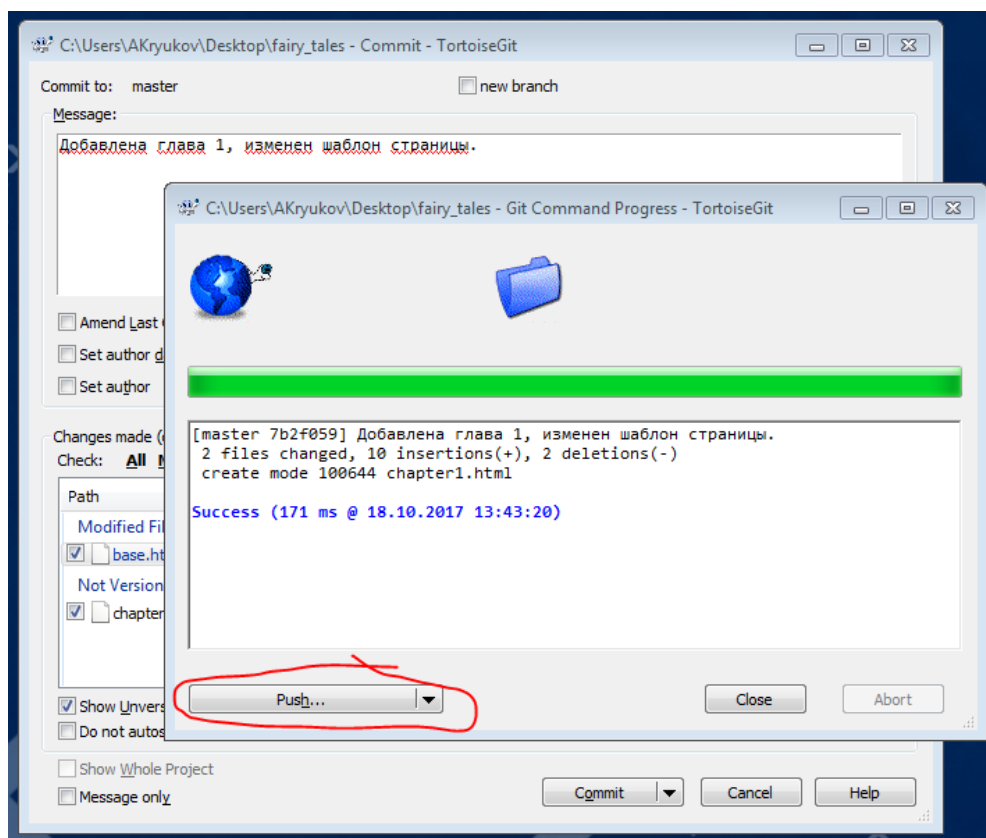


Рисунок 4.3 Расположение кнопки «Push» в окне фиксации изменений

После любого из этих действий появится окно загрузки изменений (проталкивания изменений, push) на github (рис. 4.4).

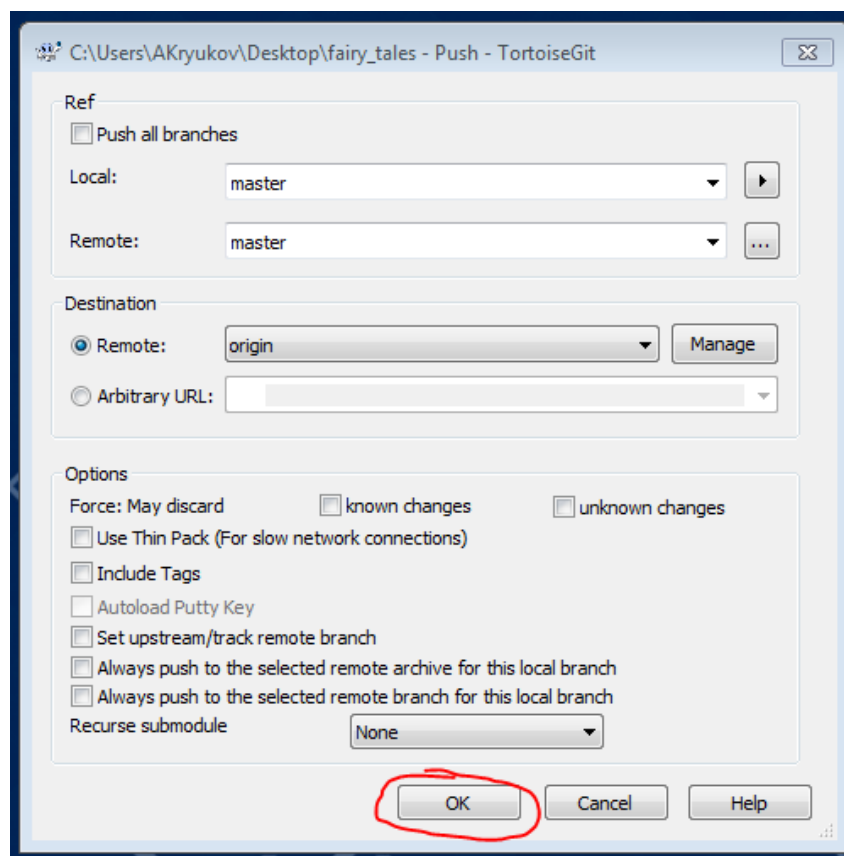


Рисунок 4.4 Окно проталкивания изменений

Нажмите кнопку “OK”. На рисунке 4.4 она отмечена **красным**.

Авторизация на github

Должно появиться окно процесса загрузки (проталкивания) изменений на github. При этом TortoiseGit предложит вам ввести имя вашего пользователя на github и пароль. Этот диалог может выглядеть как на рисунке 4.5 и как на рисунке 4.6.

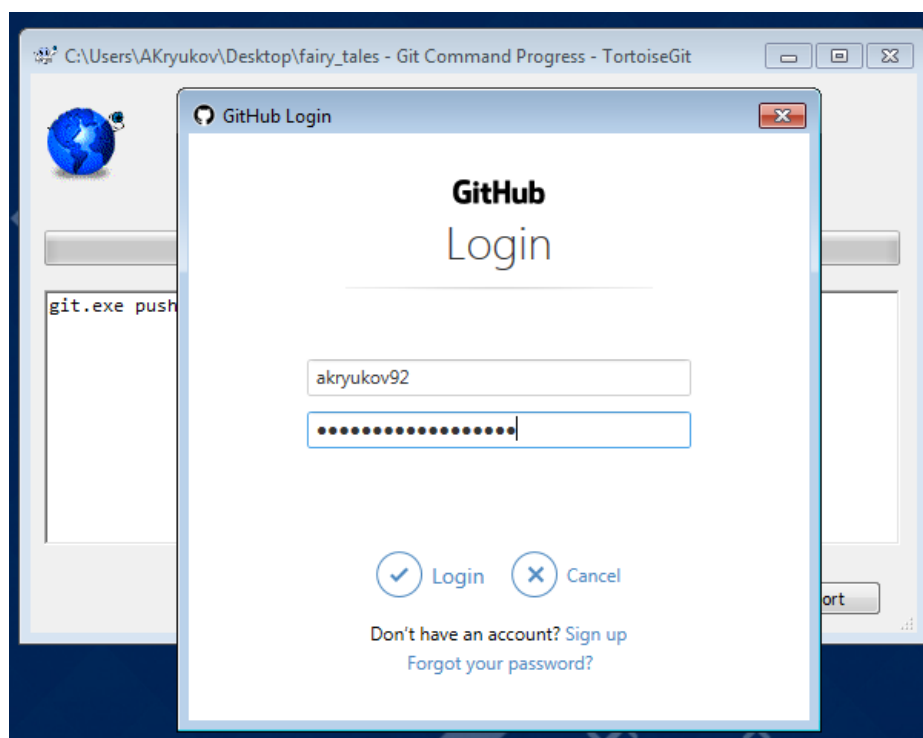


Рисунок 4.5 Окно ввода пароля github

Использование логина и пароля

Авторизация может выглядеть как на рисунке 4.6.

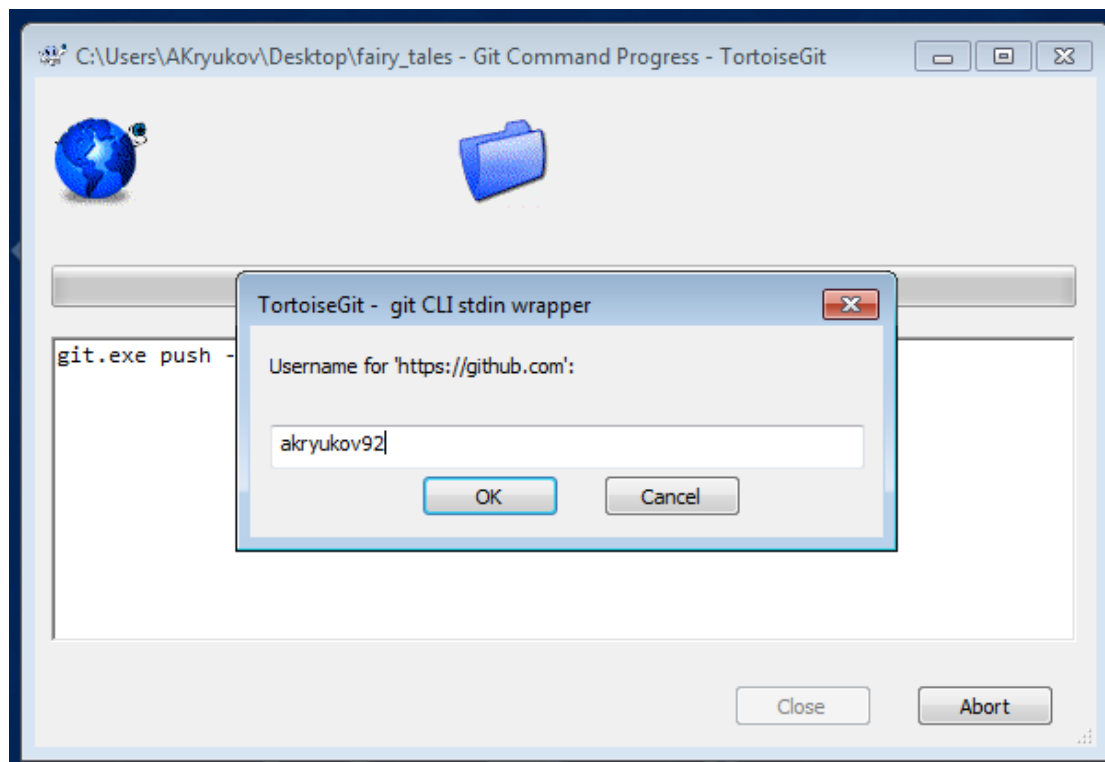


Рисунок 4.6 Окно ввода имени пользователя

В этом случае вам нужно сгенерировать пароль на github.

Перейдите по ссылке <https://github.com/settings/tokens/new> У вас должна открыться страница создания пароля как на рисунке 4.7.

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

2023-11-17

What's this token for?

Expiration *

30 days The token will expire on Sun, Dec 17 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events

Рисунок 4.7 Страница создания пароля

На странице создания пароля напишите текущую дату в поле Note и поставьте галочку напротив «repo». После этого прокрутите до конца страницы и нажмите кнопку «Generate token» как на рисунке 4.8

<input type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Generate token Cancel

Рисунок 4.8 Кнопка Generate token.

После этих действий откроется страница с паролем для репозитория. На рисунке 4.9 пароль обведен красным. Пароль выглядит примерно как «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»

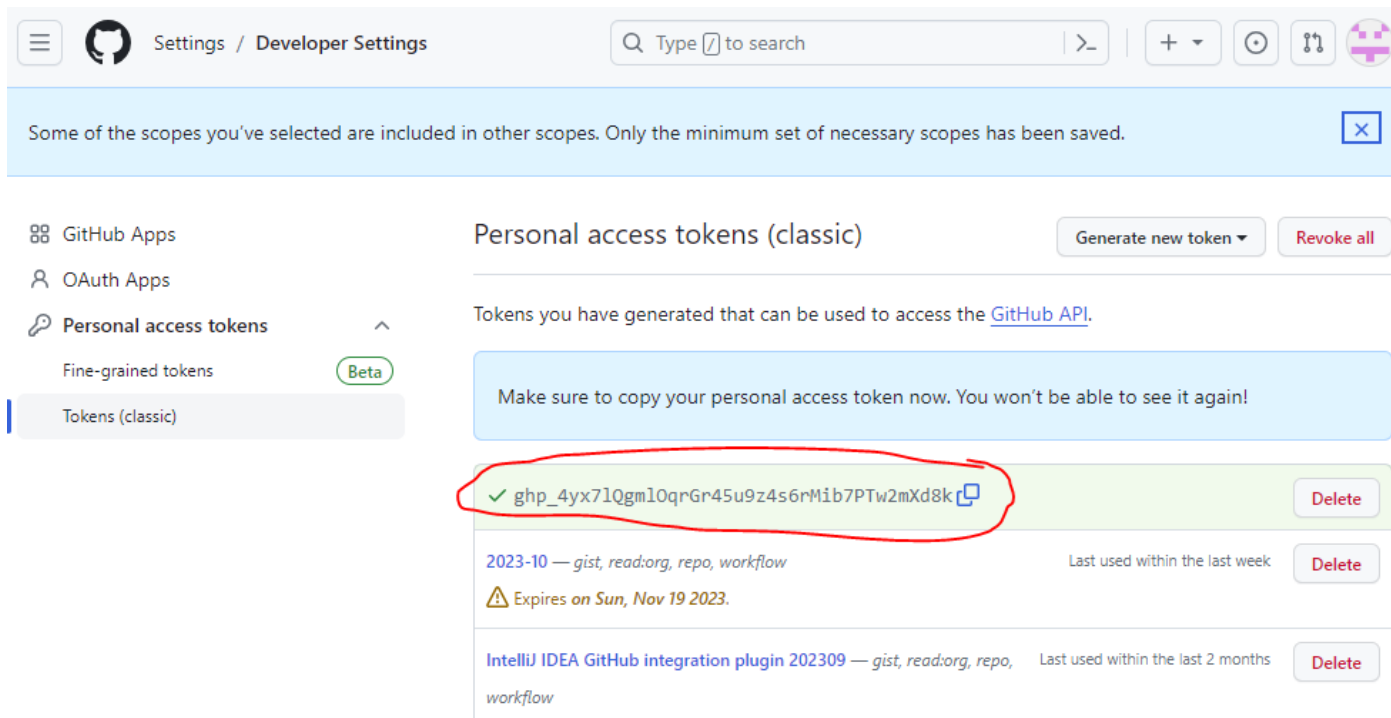


Рисунок 4.9 Страница с новым паролем.

Скопируйте пароль и сохраните его в файле на флешке (не в репозитории). Он будет действовать 30 дней, после чего нужно повторить эти действия заново.

Теперь переключитесь на окно ввода имени пользователя (рисунок 4.6). Заполните поле именем пользователя на github. После нажатия «ОК» появится еще одно окно (рис. 4.10). В него вставьте сгенерированный пароль.

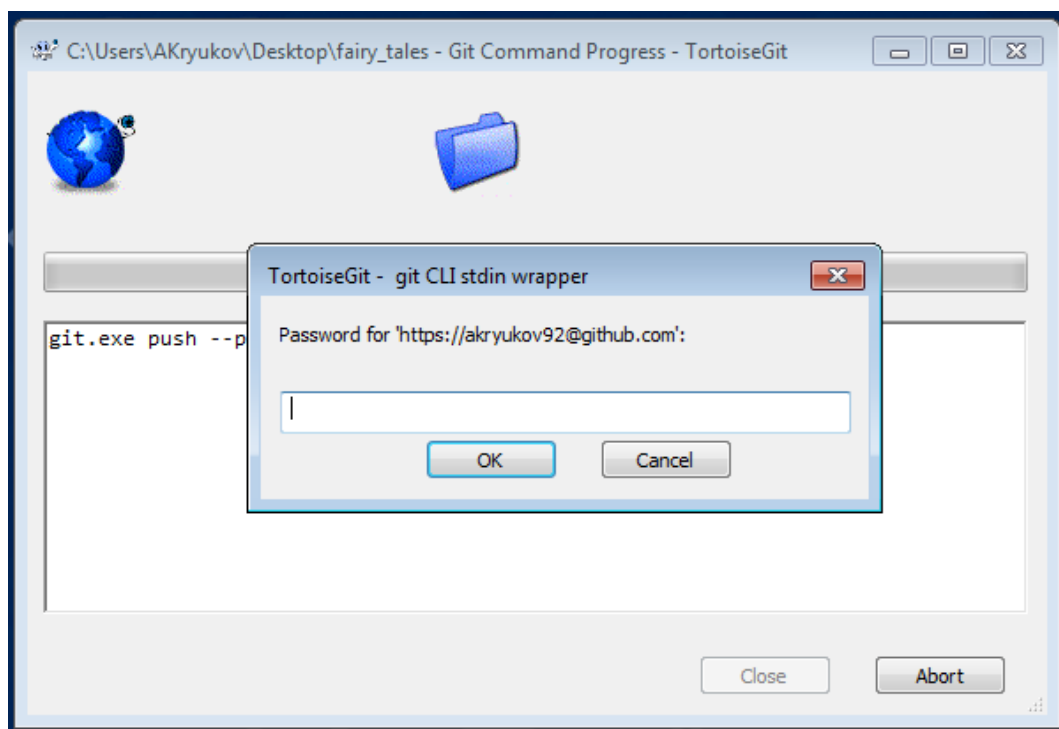


Рисунок 4.10 Окно ввода пароля пользователя

После успешной авторизации, диалог загрузки изменений будет выглядеть как на рисунке 4.11.

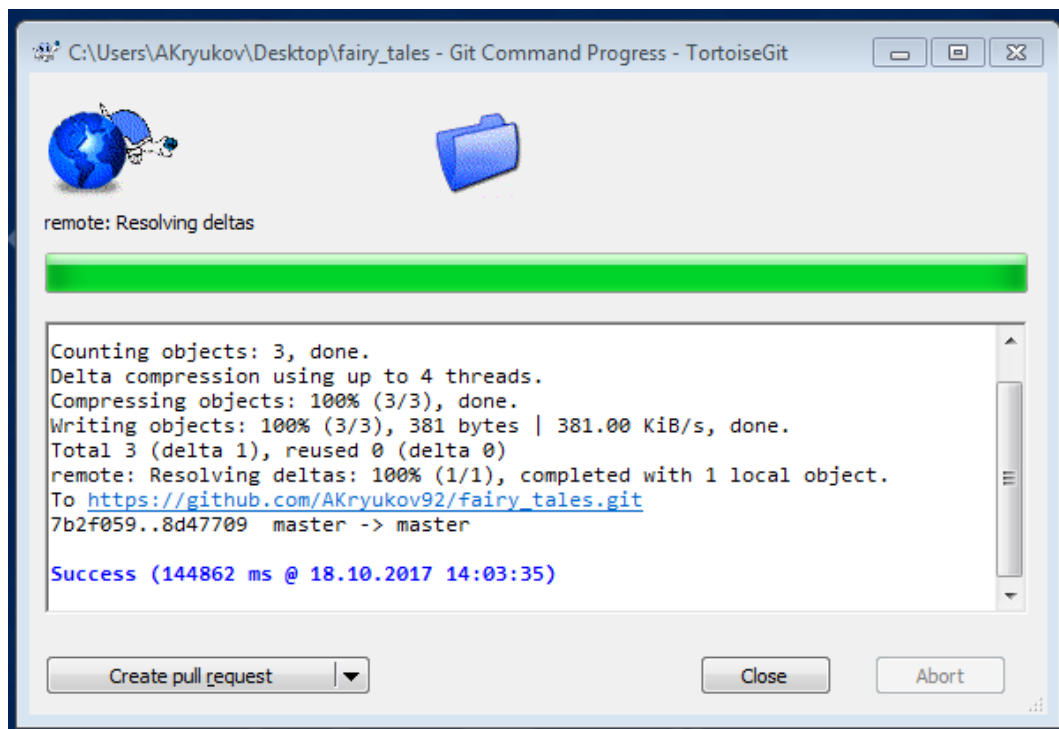


Рисунок 4.11 Окно процесса проталкивания изменений на github.com

Синяя надпись **Success** значит, что все в порядке. Если там появится красная надпись, попробуйте еще раз и проверьте правильность ввода имени пользователя и пароля.

Если имя и пароль верные, а надпись красная, то сохраняйте папку (локальный репозиторий) на флешку и зовите на помощь более опытных пользователей git.

Обновление папки на рабочем столе (затягивание изменений из github, pull)

Вам нужно обновлять папку с проектом на рабочем столе, если вы с другого компьютера загружали изменения на github.

Например:

Во вторник вы **в классе** скачали проект (клонировали репозиторий), сохранили файлы в нем (зафиксировали изменения, commit) и загрузили эти изменения на github (протолкнули изменения, push).

В среду вы **дома** скачали проект, сохранили еще какие-нибудь изменения и снова загрузили их на github.

А в четверг вы пришли **в класс** и вам нужно продолжить работу с папкой (локальным репозиторием), которая уже есть у вас на рабочем столе.

Чтобы начать обновление:

1. Нажмите правую кнопку мыши на папке-локальном репозитории.
2. Выберите меню TortoiseGit.
3. Нажмите Pull (см. рис. 5.1).

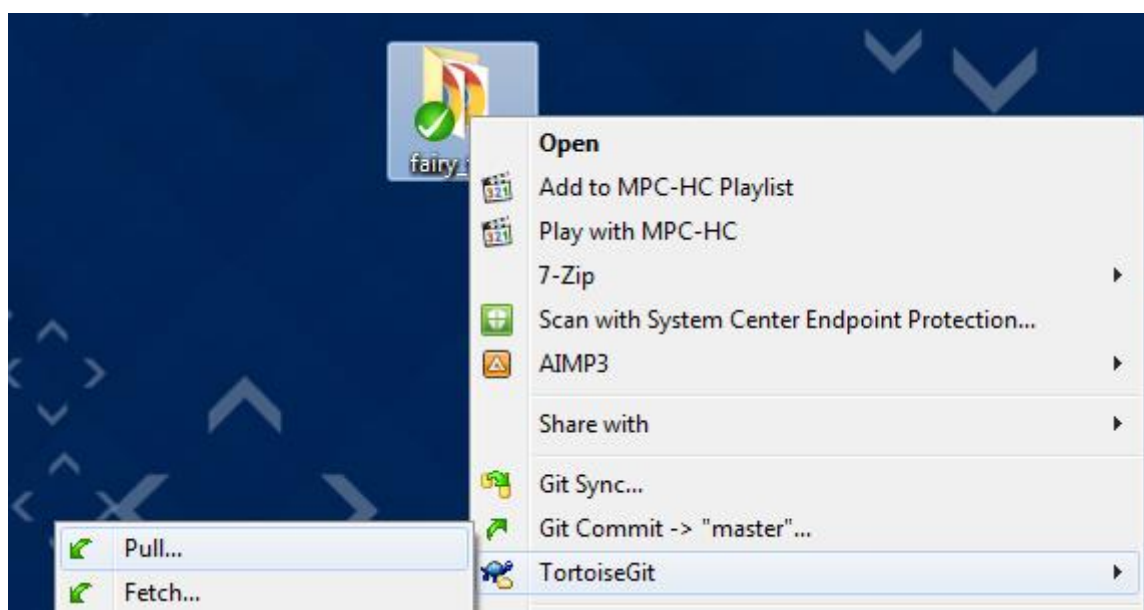


Рисунок 5.1 Пункт меню «TortoiseGit» «Pull» на рабочем столе

Другой вариант - нажмите правую кнопку мыши на свободном месте при открытии папки-локального репозитория в проводнике (см. рис. 5.2).

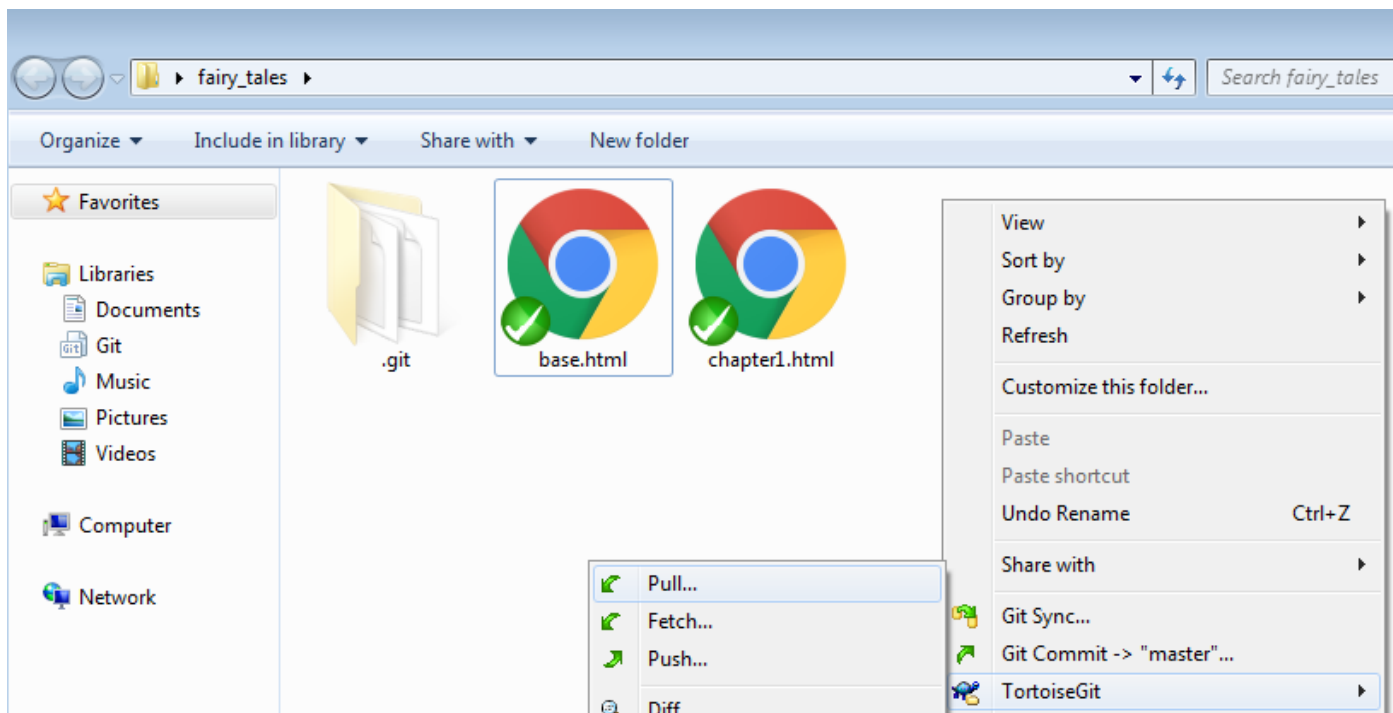


Рисунок 5.2 Пункт меню «TortoiseGit» «Pull» в локальном репозитории

При этом появится окно обновления (затягивания изменений, pull) проекта (рис. 5.3).

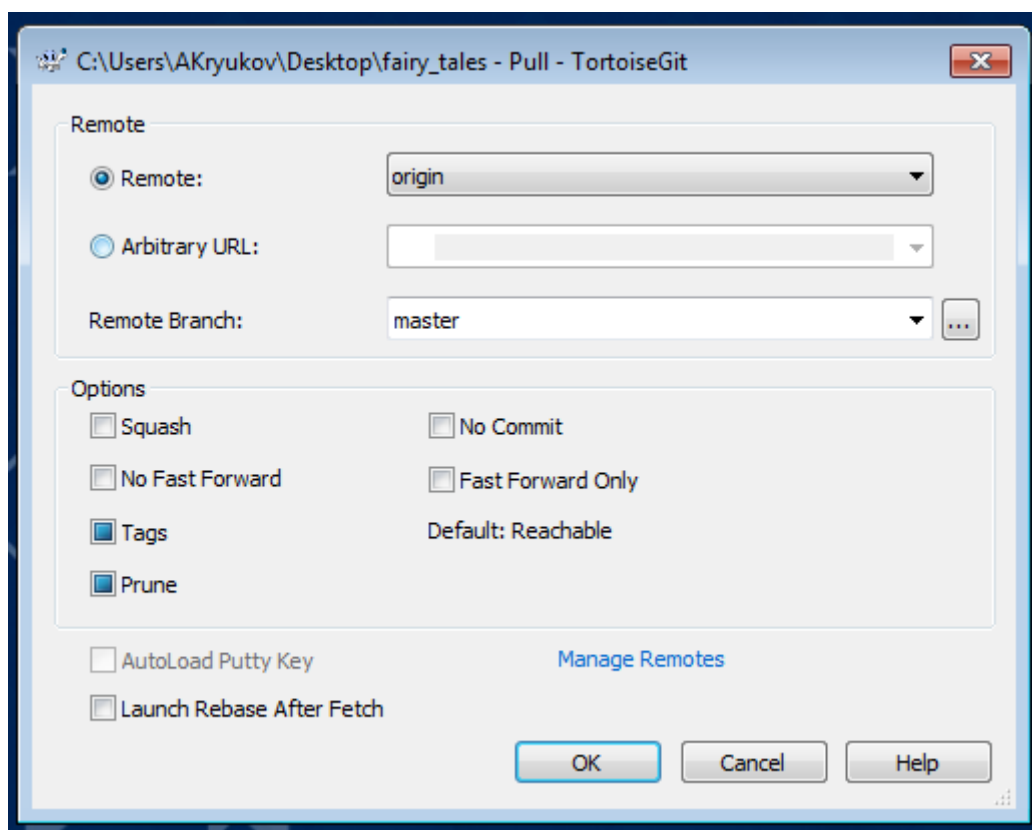


Рисунок 5.3 Окно затягивания изменений

Нажмите кнопку «ОК».

Появится окно процесса обновления (затягивания изменений) проекта (рис. 5.4).

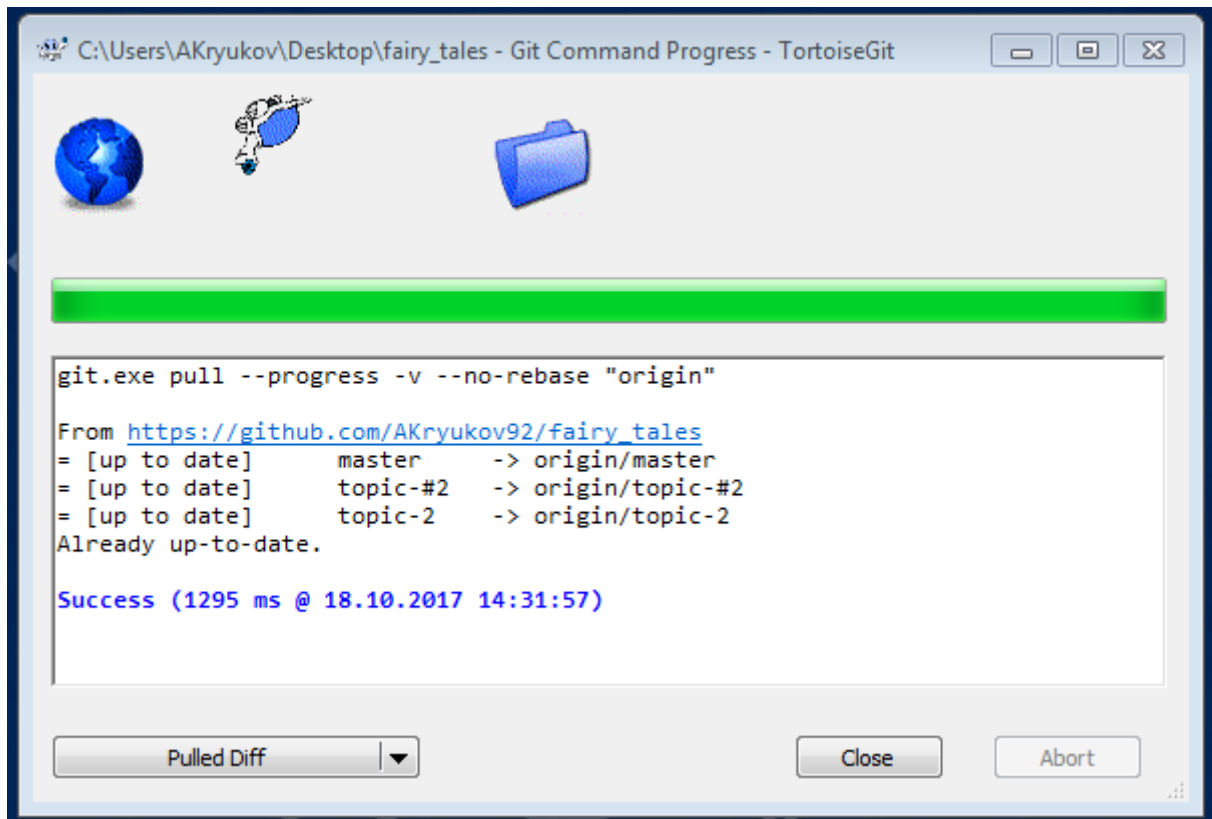


Рисунок 5.4 Окно процесса обновления

Синяя надпись **Success** значит, что проект был обновлен успешно.

Черная надпись «Already up to date» над надписью «Success» значит что версия в папке такая же, как на github.

Если внизу надпись **красная**, зовите на помощь более опытного пользователя git.

В качестве временного решения, вы можете переименовать эту папку и заново клонировать проект.