Практическая работа №6

Знакомство с распределённой системой управления версиями Git

**Указание к выполнению: результат работы принимается в виде отправленных в репозиторий файлов.**

Теоретическая часть:

Система управления версиями (от англ. Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение.

Такие системы наиболее широко используются при разработке программного обеспечения для *хранения исходных кодов* разрабатываемой программы и *обеспечения взаимодействия* с другими разработчиками.

Существуют два основных способа построения таких систем:

1. Централизованные — subversion.
2. Распределенные — git.

Изображение выглядит как счетчик

Автоматически созданное описание

Рисунок 1 — Основные способы построения систем управления версиями

*Централизованные системы,* как Subversion, CVS и Perforce используют единственный сервер, содержащий все версии файлов, и некоторое количество клиентов, которые получают файлы из этого централизованного хранилища. Применение таких систем являлось стандартом на протяжении многих лет.

В *распределённых системах*, таких как Git, Mercurial, Bazaar или Darcs, клиенты не просто скачивают снимок всех файлов (состояние файлов на определённый момент времени) — они полностью копируют репозиторий. В этом случае, если один из серверов, через который разработчики обменивались данными, отключится, любой клиентский (локальные) репозиторий может быть скопирован на другой сервер для продолжения работы. Каждая копия репозитория является полным бэкапом всех данных.

Практическая часть:

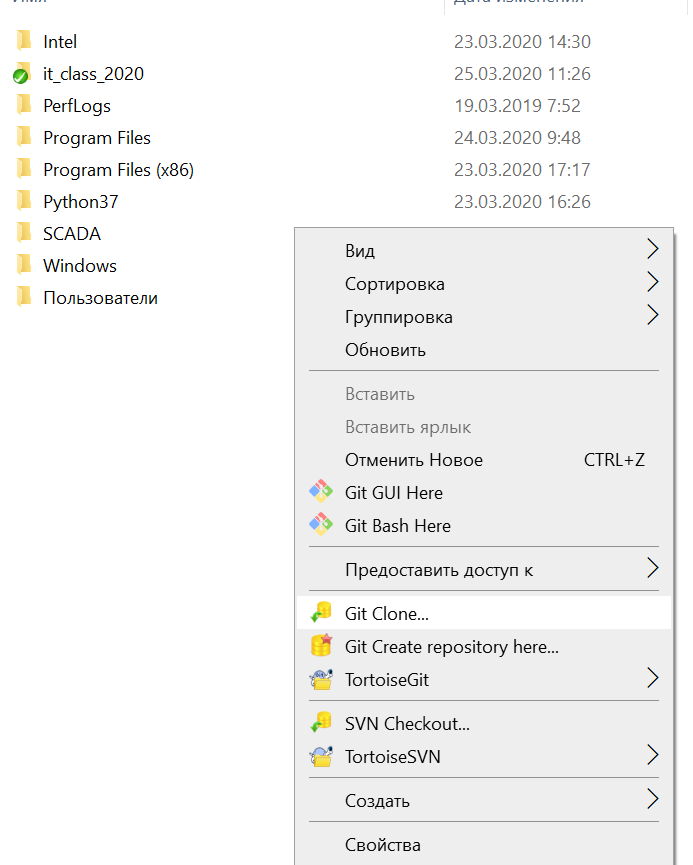
Необходимо создать свою локальную копию (local repository) репозитория ИТ класса и организовать себе рабочее пространство (working copy), отработать основные команды.

1. Удаленный репозиторий будет храниться на сервисе GitHub. Необходимо зарегистрироваться на <https://github.com/join>.

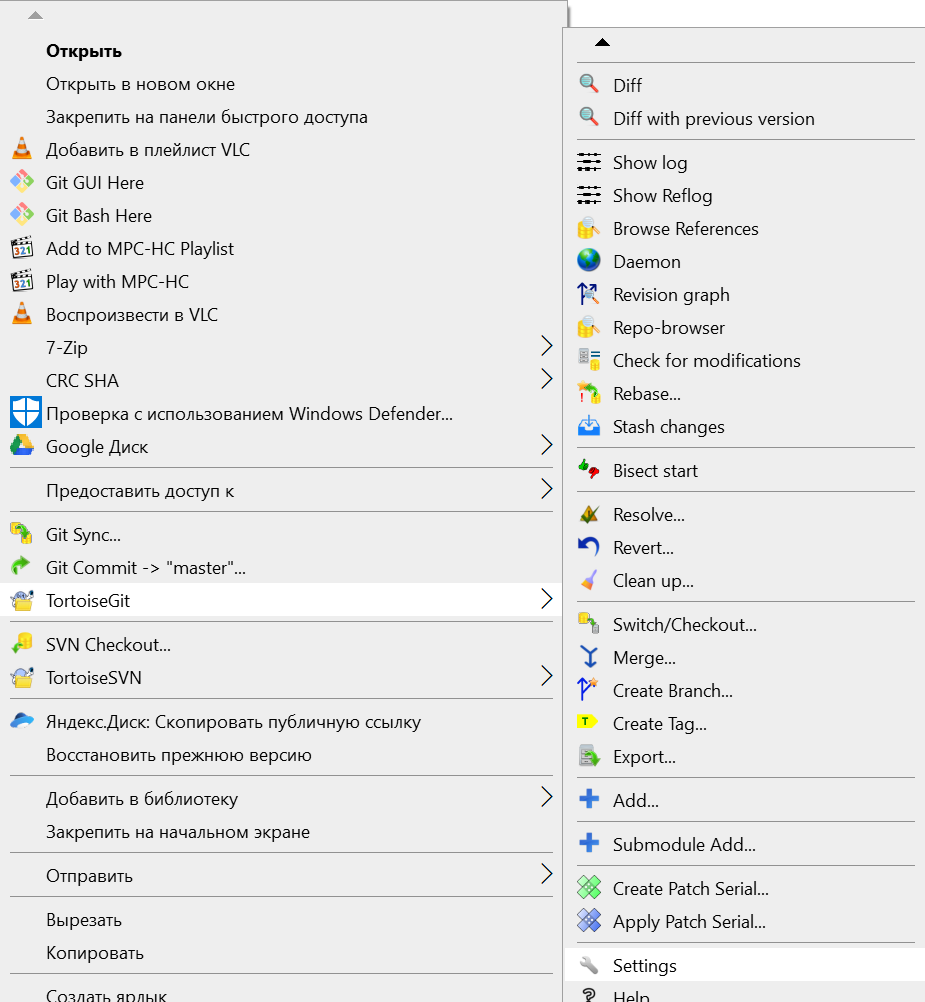
После регистрации прислать на почту [fomin.is@1561.ru](mailto:fomin.is@1561.ru) свой логин, для установки доступа к репозиторию. На почту, указанную при регистрации, придет приглашение, после его подтверждения можно продолжать работу.

1. Скачиваете и устанавливаете TortoiseGit версии 2.10.0, доступный по ссылке <https://tortoisegit.org/download/>

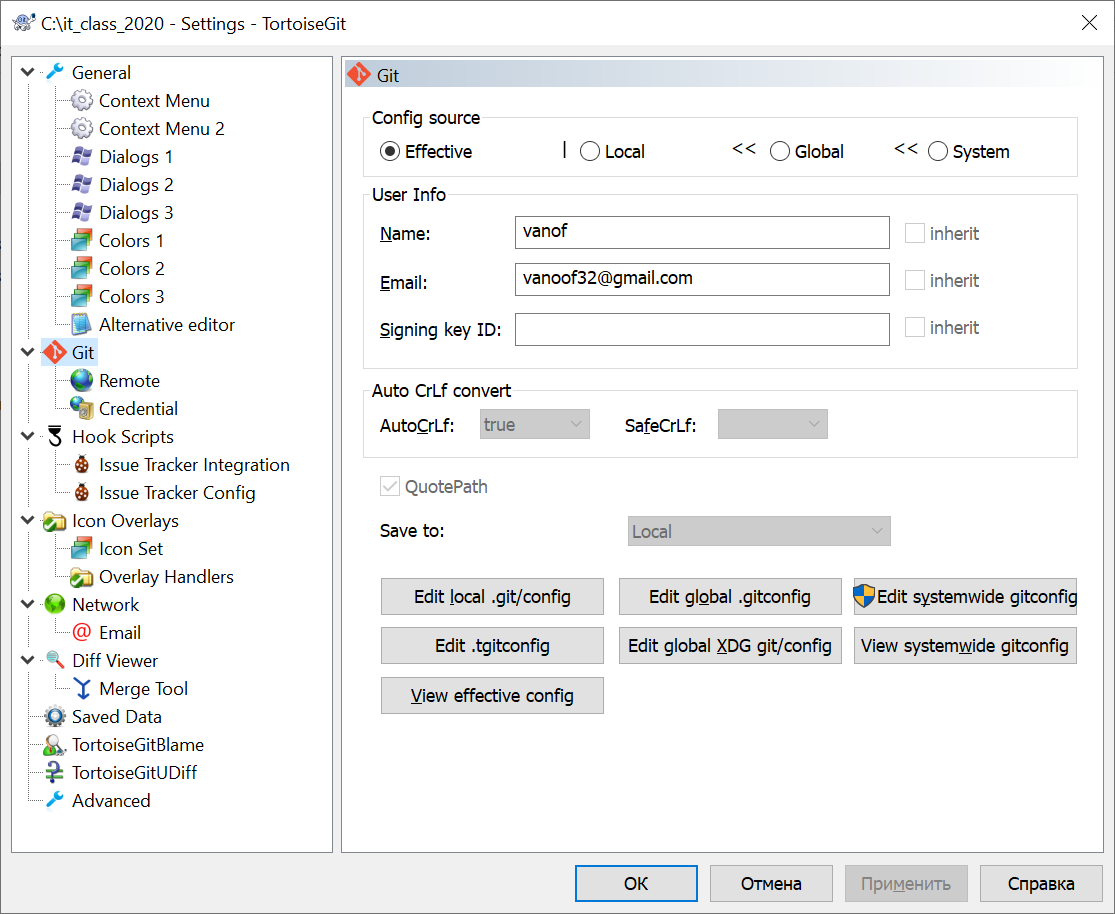
TortoiseGit предоставляет графический интерфейс для работы с репозиториями, встраиваемый в контекстное меню операционной системы:



1. Зайти в настройки TortoiseGit – Settings.

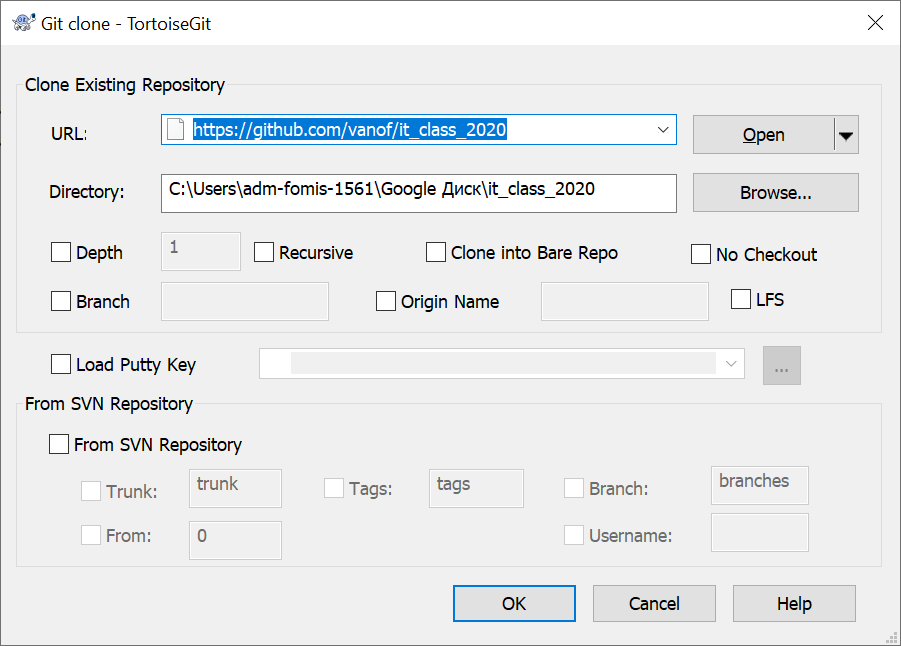


1. Установить в разделе Git name (логин) и email:

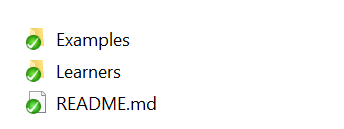


1. Для создания локальной копии репозитория, выполняете команду «Git Clone…».

Устанавливаете URL удаленного репозитория <https://github.com/vanof/it_class_2019>

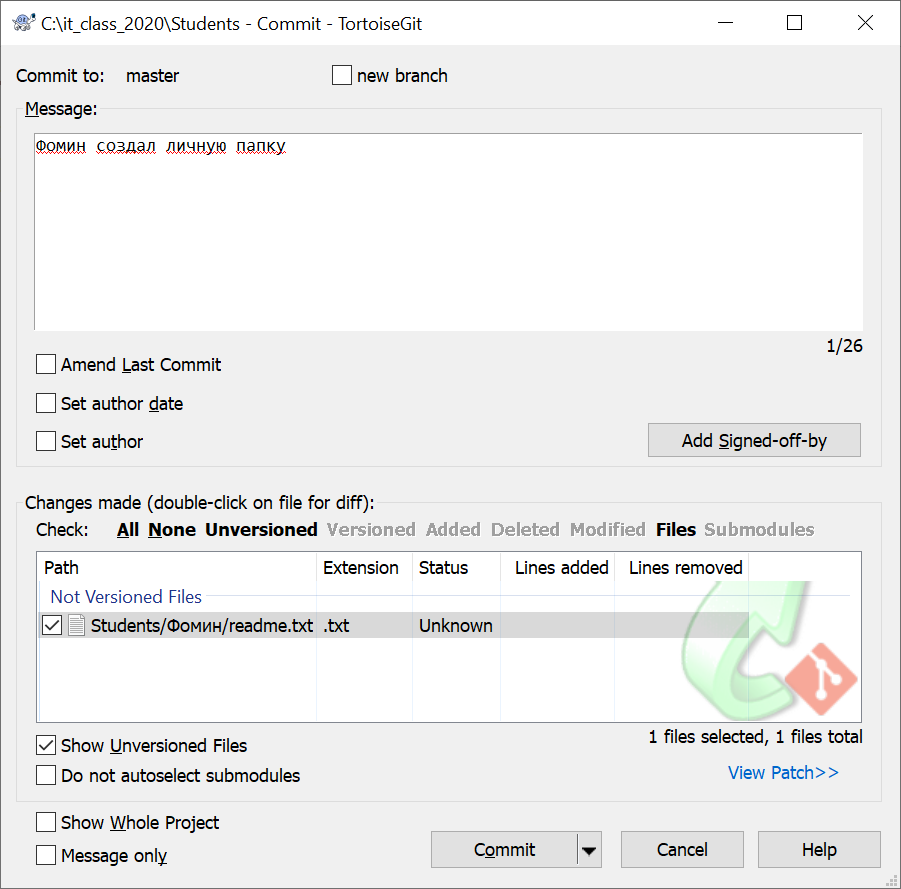


При успешном завершении вы получите сообщение «Success» с временем выполнения — локальная копия создана.

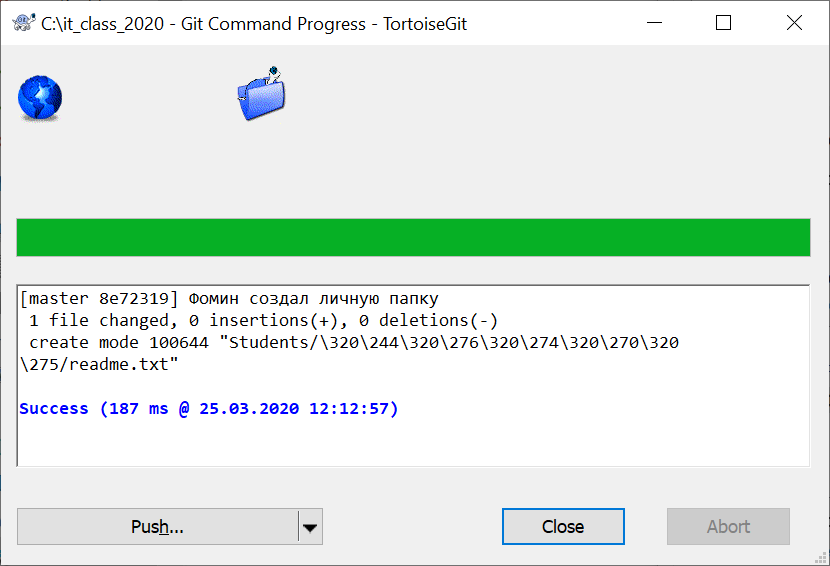


В папке Examples хранятся различные примеры программного кода, необходимые нам для обучения. Ознакомьтесь с ними.

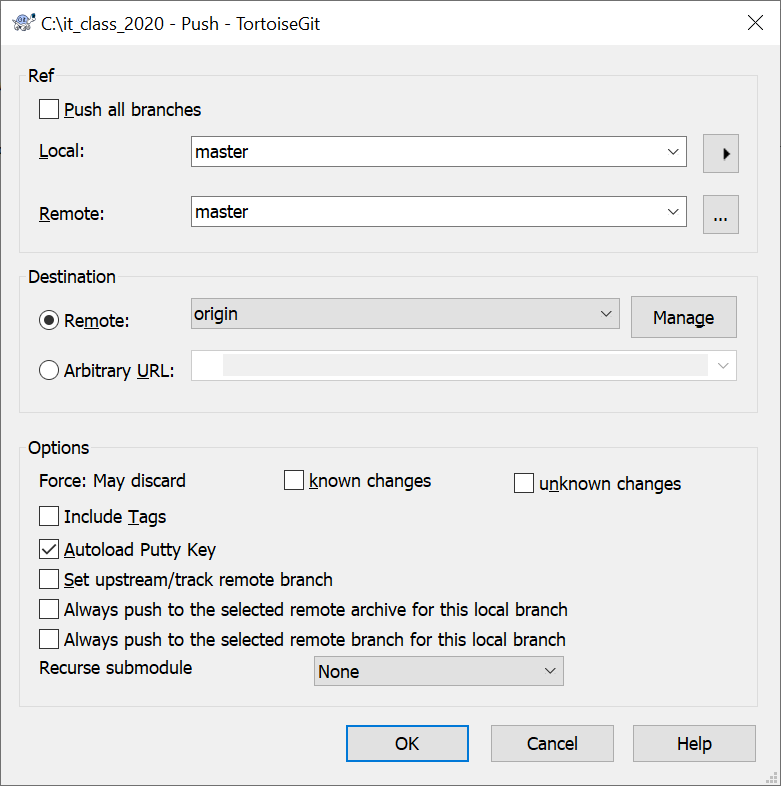
1. В папке Students создайте папку, названную своей фамилией, и разместите там пустой файл readme.txt. Так вы начнете создавать структуру папок для дальнейшей работы.
2. Внесем изменения (*commit*) в локальный репозиторий. По умолчанию в git он называется master. В папке Students выполните команду «Git Commit -> «master». Напишите развернутый комментарий и выберите файл для отправки.



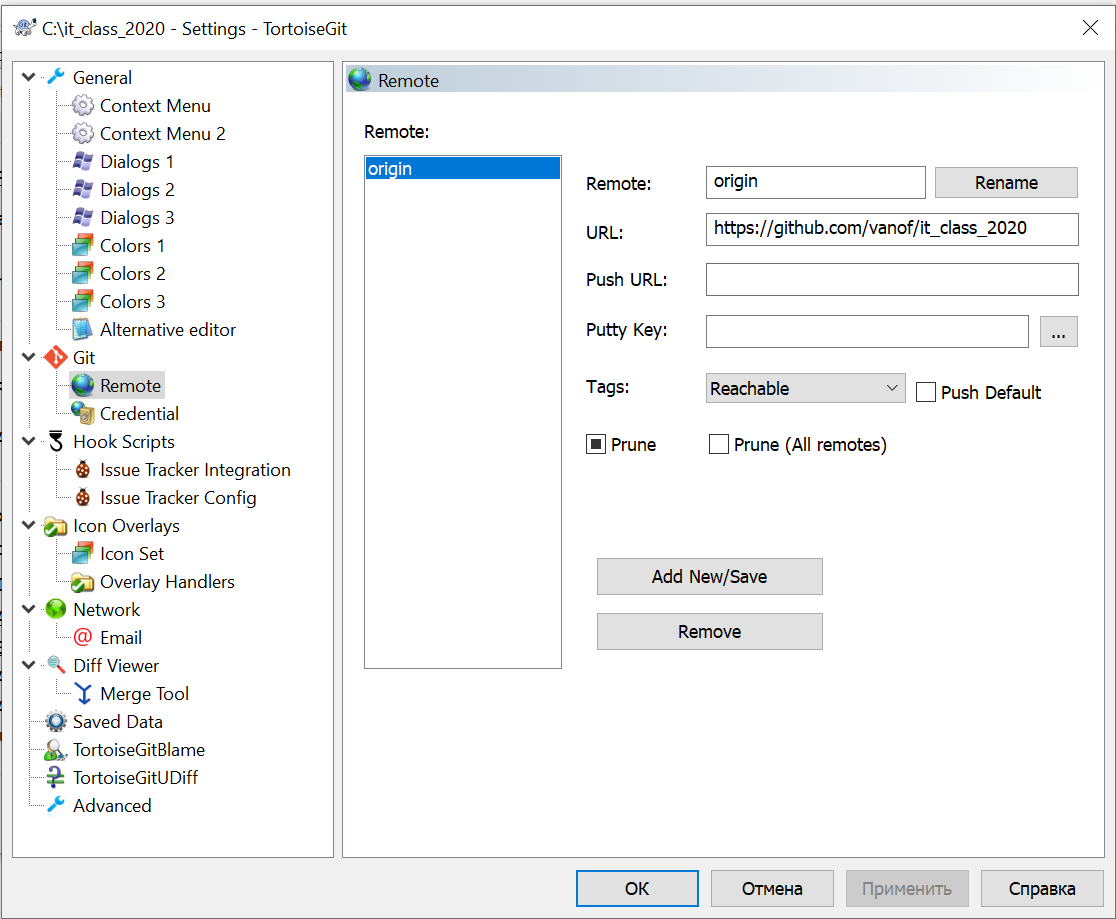
При успешном завершении вы получите сообщение «Success» с временем выполнения — изменения в локальную копию внесены. Интерфейс программы сразу же предложит вам совершить *push* — передать изменения на удаленный репозиторий.



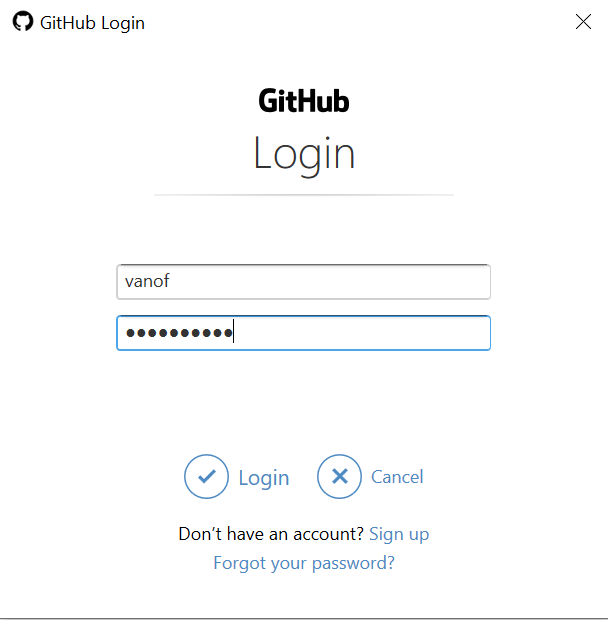
1. Внесем изменения в удаленный репозиторий. Нажимаете «Push…».



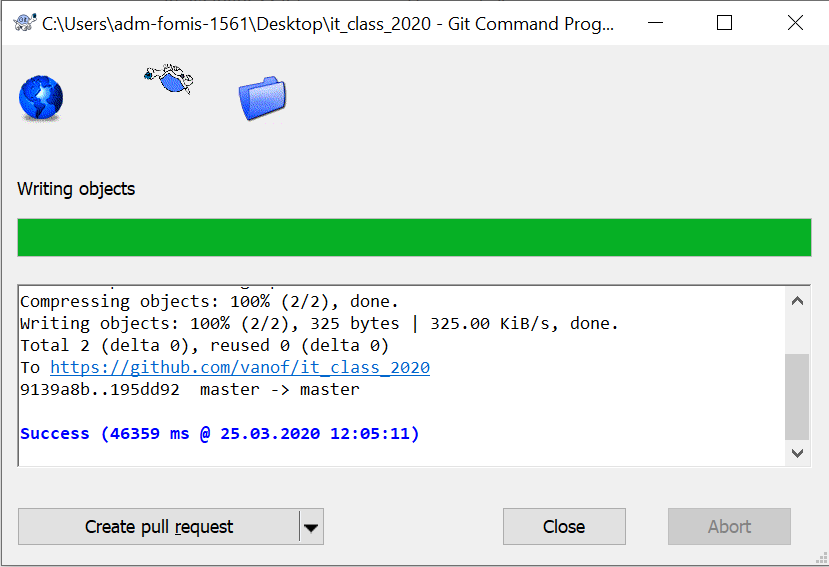
Нажимаете кнопку «Manage», выбираете в списке Remote удаленный репозиторий origin и проверяете его URL, он должен совпадать с https://github.com/vanof/it\_class\_2019



Запускаете команду *push*. На этом этапе удаленный репозиторий попросит вас авторизоваться:



И после авторизации произойдет закачка изменений на удаленный репозиторий.



На этом работа с репозиторием закончена. Результат выполнения практической работы — ваша личная папка, куда вы будете выкладывать программный код.