

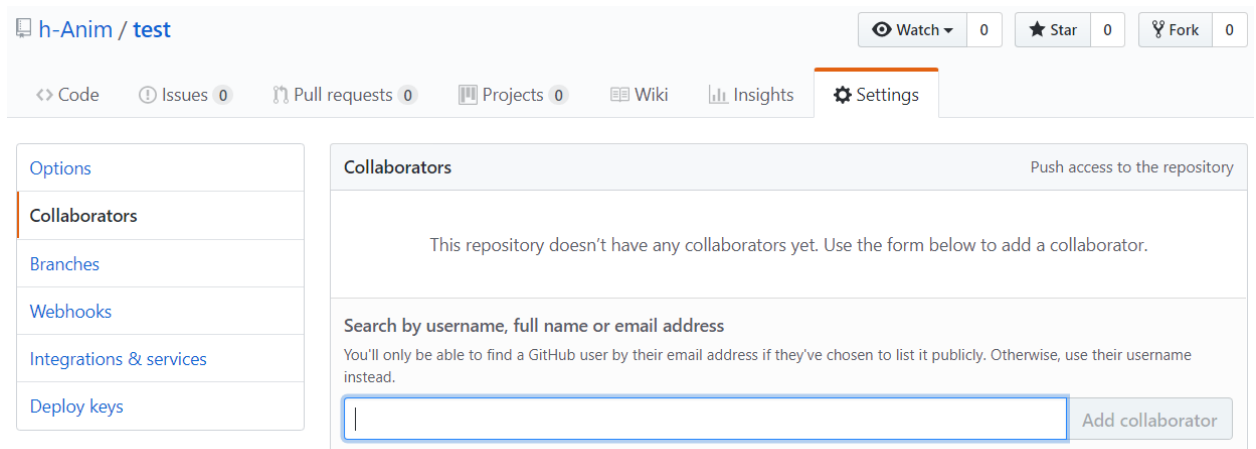
## Лабораторная работа №3: Основы работы с Git: командная разработка

### Цель:

Целью данной работы является получение базовых навыков командной работы в системе контроля версий git

### Добавление разработчиков на GitHub:

Для того, что бы добавить разработчика к работе над репозиторием, перейдите в раздел **Settings** и выберите пункт **Collaborators**:



В появившемся поле ввода, вы можете добавить пользователя по его имени на GitHub, или адресу электронной почты. После чего, добавленный вами пользователь может клонировать и синхронизировать общий репозиторий, а так же, совершать commit непосредственно в общий репозиторий.

### Получение изменений:

Для **получения** изменений из удалённого репозитория, используется команда `git fetch`:

```
MINGW64:/e/Docs/method/TRPO/src/helloworld

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git fetch

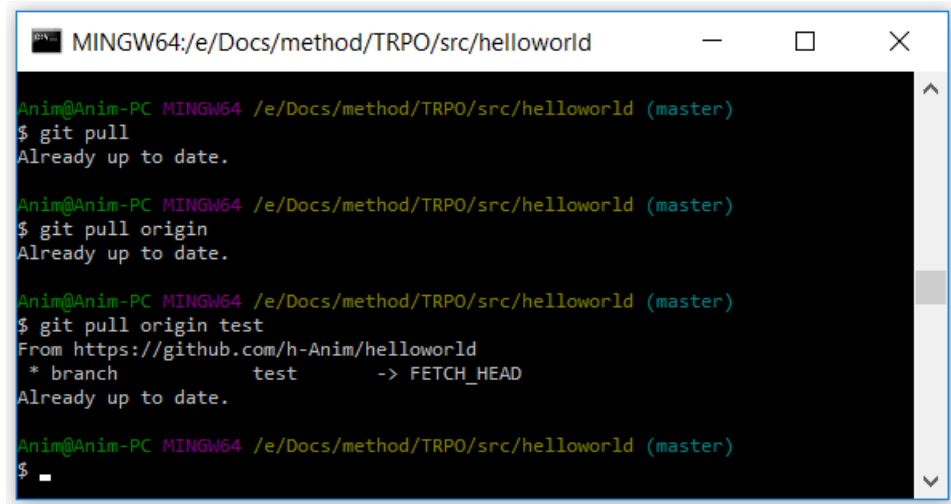
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git fetch origin

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git remote -v
origin  https://github.com/h-Anim/helloworld (fetch)
origin  https://github.com/h-Anim/helloworld (push)

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$
```

Команда может вызываться как без параметров, так и с указанием репозитория, с которым происходит синхронизация.

Для того, что бы **извлечь** изменения из удалённого репозитория и **применить** их к текущей ветви, используется команда `git pull`:



```
MINGW64:/e/Docs/method/TRPO/src/helloworld

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git pull
Already up to date.

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git pull origin
Already up to date.

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git pull origin test
From https://github.com/h-Anim/helloworld
 * branch      test      -> FETCH_HEAD
Already up to date.

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$
```

В первом случае, изменения будут получены из текущего установленного удалённого репозитория для текущей ветви. Во втором случае, изменения будут получены из `origin`. В третьем случае, изменения будут получены из `origin` для ветви `test`.

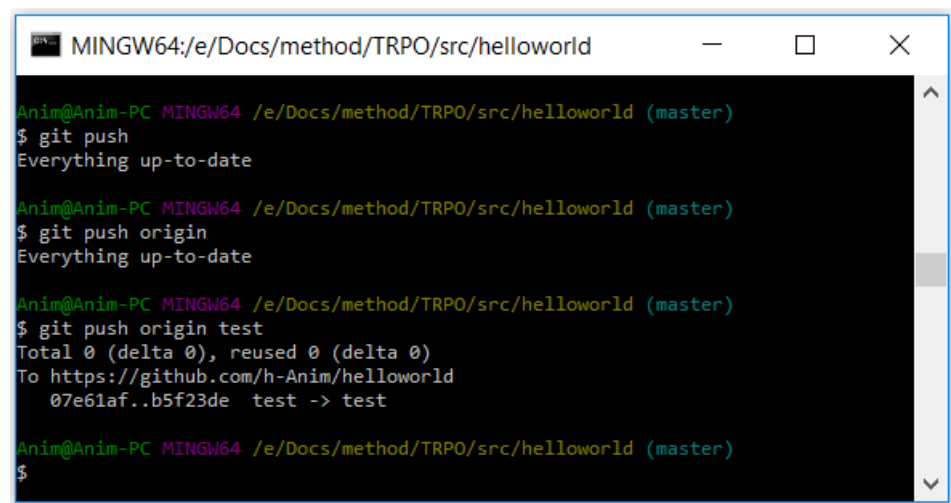
#### Примечание:

Вы можете создать копию ветви, существующей в удалённом репозитории, используя команду: `git checkout -b local_name origin/remote_name`

Где `local_name` имя ветви в локальном репозитории, а `remote_name` имя ветви в удалённом.

#### Отправление изменений:

Для того, что бы **отправить** локальные изменения в удалённый репозиторий, используется команда `git push`:



```
MINGW64:/e/Docs/method/TRPO/src/helloworld

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git push
Everything up-to-date

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git push origin
Everything up-to-date

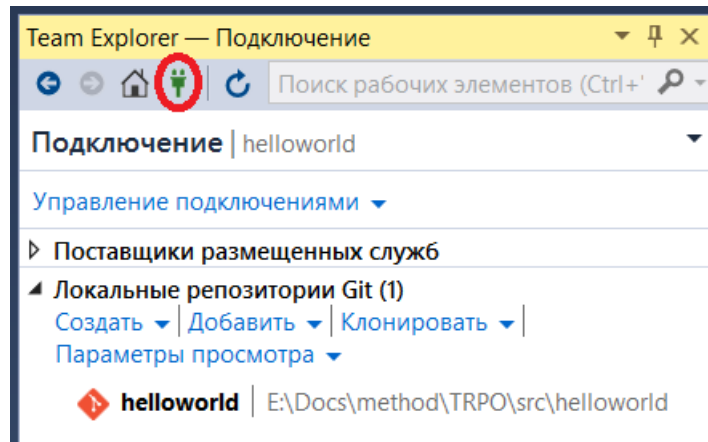
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git push origin test
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/h-Anim/helloworld
 07e61af..b5f23de test -> test

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$
```

В первом случае, изменения будут отправлены в текущий установленный удалённый репозиторий для текущей ветви. Во втором случае, изменения будут отправлены в `origin`. В третьем случае, изменения ветви `test` будут отправлены в `origin`.

## Интеграция Git с Visual Studio 2017:

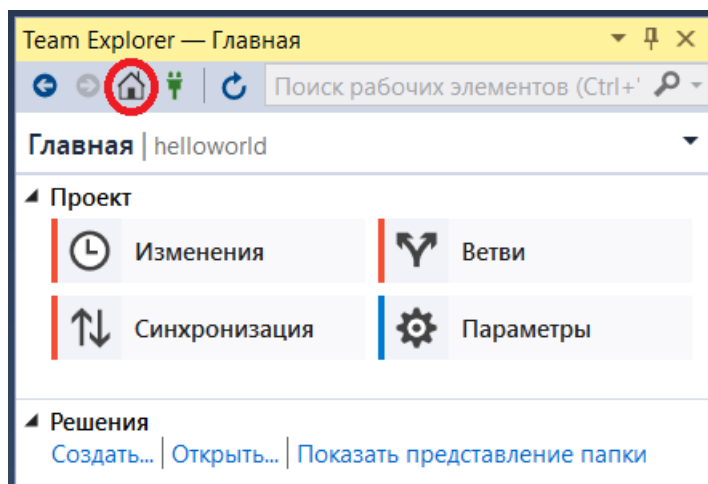
Для того, что бы начать работу с системой контроля версий в Visual Studio, перейдите к окну **Team Explorer** и нажмите **Управление подключениями**:



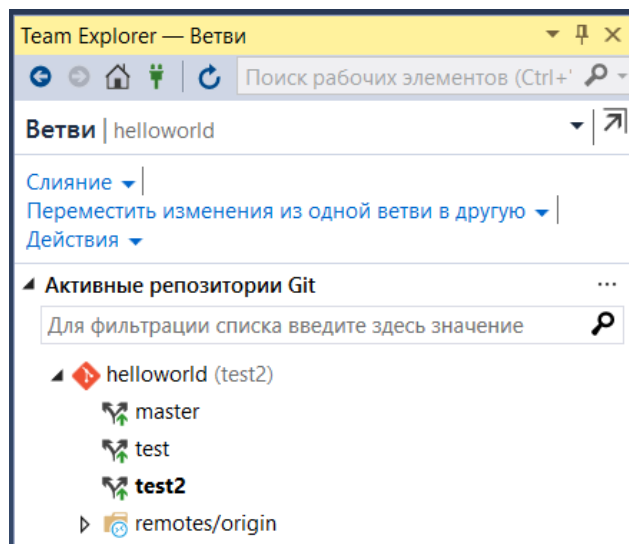
Если окно Team Explorer отсутствует, добавить его можно через Вид -> Team Explorer.

В появившемся окне, вы можете создать новый, локальный, репозиторий, добавить уже существующий, или создать новый на основе удалённого.

Для того, что бы начать работу с проектом, перейдите в раздел **Главная**:

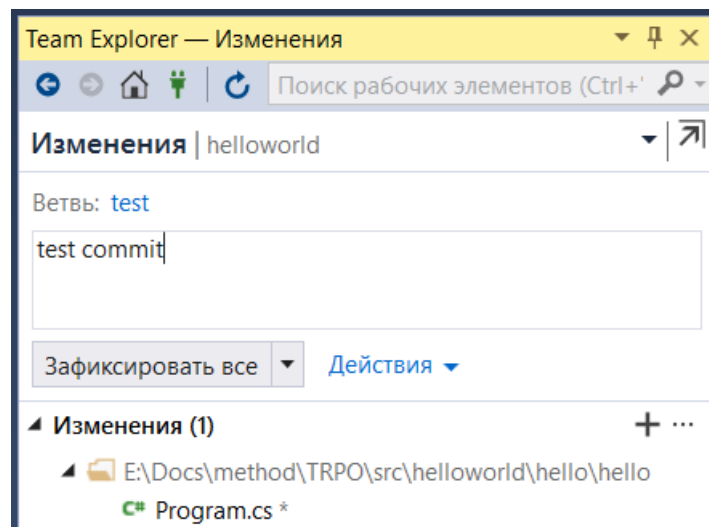


В разделе **Ветви**, будут указаны существующие ветви проекта:



Для того, что бы **переключиться** на нужную ветвь, можно дважды кликнуть на неё левой кнопкой мыши, либо нажать на нужную ветвь правой кнопкой мыши и выбрать **Извлечь**, в появившемся контекстном меню.

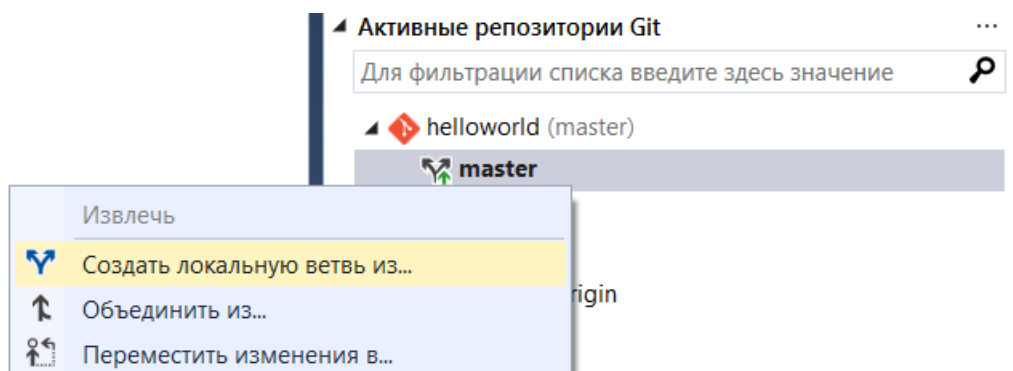
Для применения изменений сделанных в ветви, вернитесь в раздел **Главная** и выберите **Изменения**:



Нажав на **Действия** и выбрав **Посмотреть историю**, можно увидеть журнал изменений проекта:

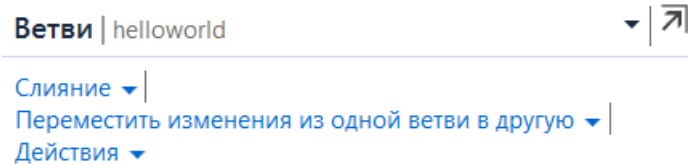
Диаграмма	Идентификатор	Автор	Дата	Сообщение
▲ Локальный журнал				
●	dc754415	h-Anim	25.02.2018 23:44:09	test commit
●	b5f23de2	h-Anim	19.02.2018 0:46:35	upload
●	2377b315	h-Anim	19.02.2018 0:42:22	upload
●	07e61af2	h-Anim	19.02.2018 0:19:37	test3
●	d7e025c5	h-Anim	19.02.2018 0:18:17	test2
●	2bd5f877	h-Anim	19.02.2018 0:13:36	fix
●	cecb699a	h-Anim	19.02.2018 0:00:48	test commit
●	72d61717	h-Anim	19.02.2018 0:01:27	test commit2
●	10a340b7	h-Anim	18.02.2018 22:35:28	commit to test branch
●	1f4a628a	h-Anim	14.02.2018 22:38:04	first project

Для создания новой ветви, вернитесь в раздел ветвей, нажмите **правой кнопкой мыши** на ветвь, от которой хотите начать ветвление и выберите **Создать локальную ветвь из:**

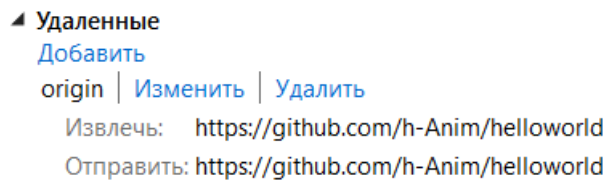


После чего, укажите имя новой ветви, и нажмите **Создать ветвь**.

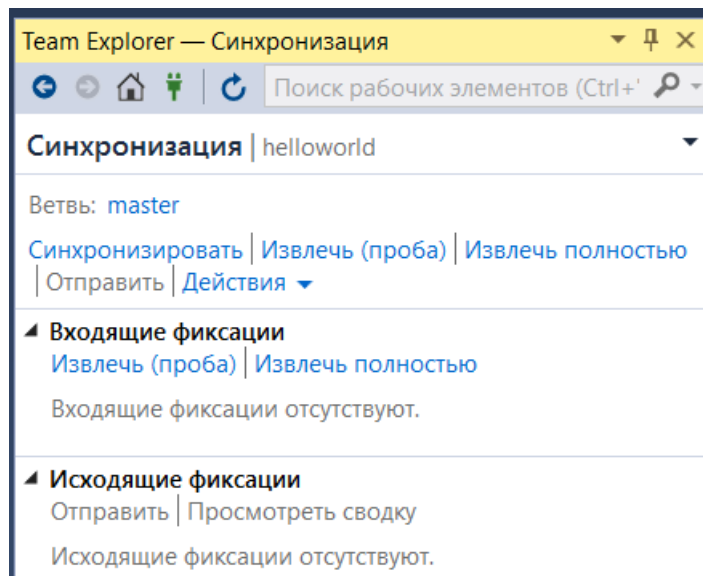
Для слияния и перемещения ветвей, используются соответствующие опции раздела Ветви:



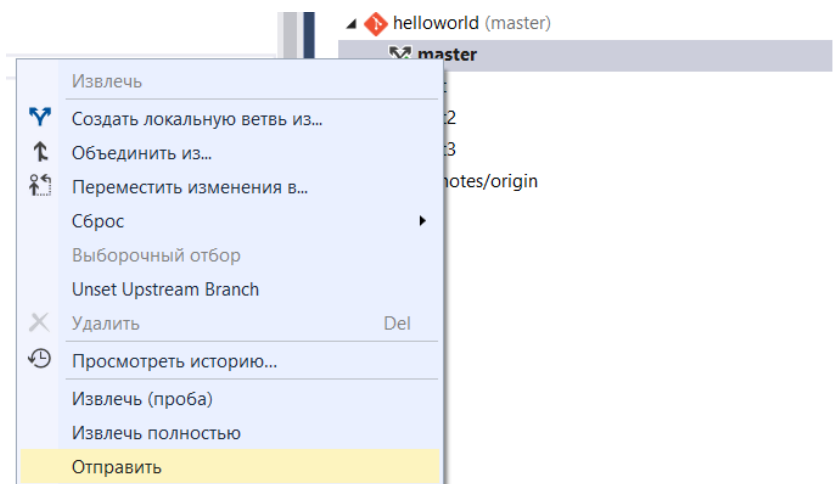
Для того, что бы назначить удалённый репозиторий, перейдите в раздел **Главная**, а затем, **Параметры**:



Для того, что бы **Извлечь** (pull) или **Отправить** (push) данные в удалённый репозиторий, перейдите на **Главную**, а затем в раздел **Синхронизация**:



Либо нажмите правой кнопкой мыши на ветвь в разделе **Ветви** и выберите **Извлечь полностью/Отправить**:



## Задания:

1. Создать репозиторий на GitHub.
2. Предоставить доступ к репозиторию ещё одному пользователю.

Используя средства для работы с Git встроенные в Visual Studio 2017:

1. Создать локальный репозиторий.
2. Создать в локальном репозитории консольное приложение C#, выводящее в консоль меню выбора, состоящее из следующих пунктов:
  - ввести A
  - ввести B
  - выполнить операцию “+”
  - выполнить операцию “-”
  - выполнить операцию “\*”
  - выполнить операцию “/”
3. Синхронизировать локальный репозиторий, с созданным на GitHub.
4. Создать ветви для реализации функционала каждого из пунктов меню в разных директориях/на разных компьютерах.
5. Реализовать функции меню, каждую в своей ветви.
6. Синхронизировать локальные репозитории через репозиторий на GitHub.
7. Создать ветвь “final” и объединить в ней все ветви проекта.
8. Объединить ветви “final” и “master”.
9. Синхронизировать финальную версию проекта с репозиторием на GitHub.

## Список литературы:

Более подробную информацию о git можно получить в книге:

Pro Git by Scott Chacon: <https://git-scm.com/book/ru/v2/> (на русском языке)

Git and VSTS: <https://docs.microsoft.com/en-us/vsts/git/gitquickstart?tabs=visual-studio>

Интерактивное обучение git: <https://try.github.io/levels/1/challenges/1> (на английском языке)