

Лабораторная работа №4.2: Основы работы с Git, GitHub и командная разработка

Цель работы:

- получить базовые навыки командной работы с помощью системы контроля версий Git.

Работа с удалённым репозиторием

Зарегистрируйтесь на [GitHub](#) и создайте новый репозиторий. Имя репозитория на GitHub может отличаться от имени локального репозитория.


Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * IntroVert / Repository name * helloworld ✓

Great repository names are short and memorable. Need inspiration? How about [solid-octo-bassoon?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

Для загрузки локального репозитория на GitHub используются команды `add` `origin` (назначение адреса удаленного репозитория) и `push` (непосредственно загрузка):

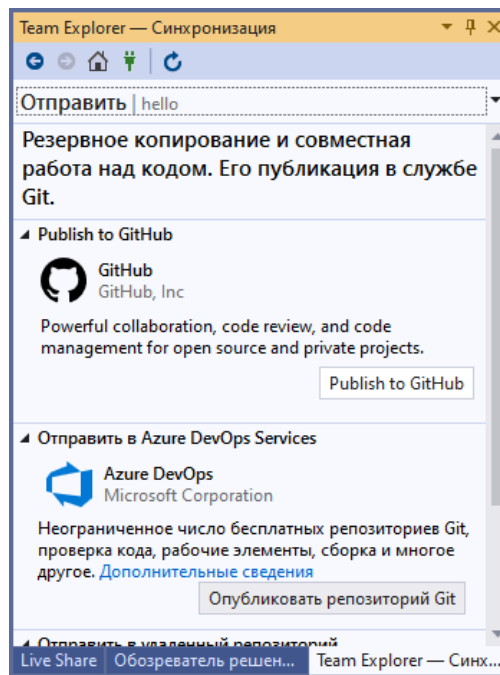
```
MINGW64:/e/Docs/method/TRPO/src/helloworld
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git remote add origin https://github.com/h-Anim/helloworld

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)

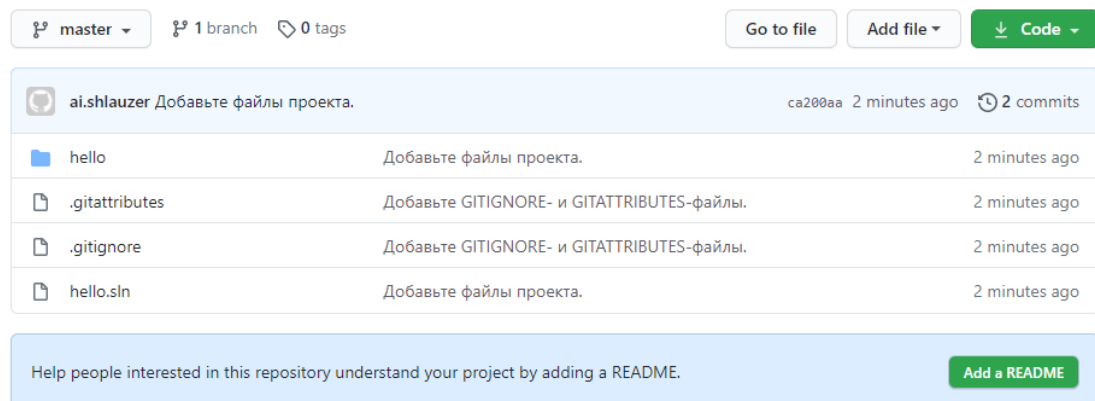
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git push -u origin master
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 5.52 KiB | 942.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To https://github.com/h-Anim/helloworld
 * [new branch] master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

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

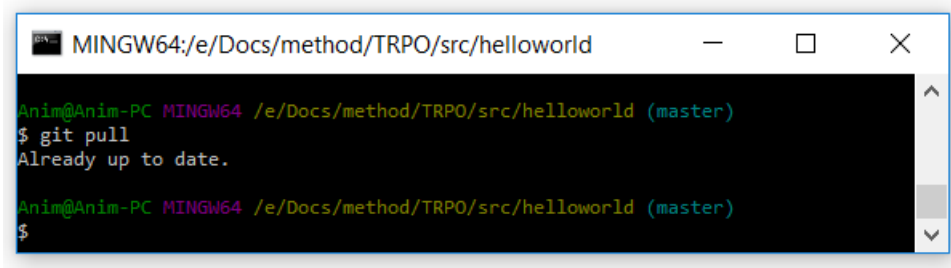
Для проектов Visual Studio есть альтернативный вариант загрузки репозитория на GitHub. В Team Explorer есть функция «Publish to GitHub»:



После загрузки локального репозитория на GitHub он будет выглядеть следующим образом:



Для получения изменений проекта используется команда pull:



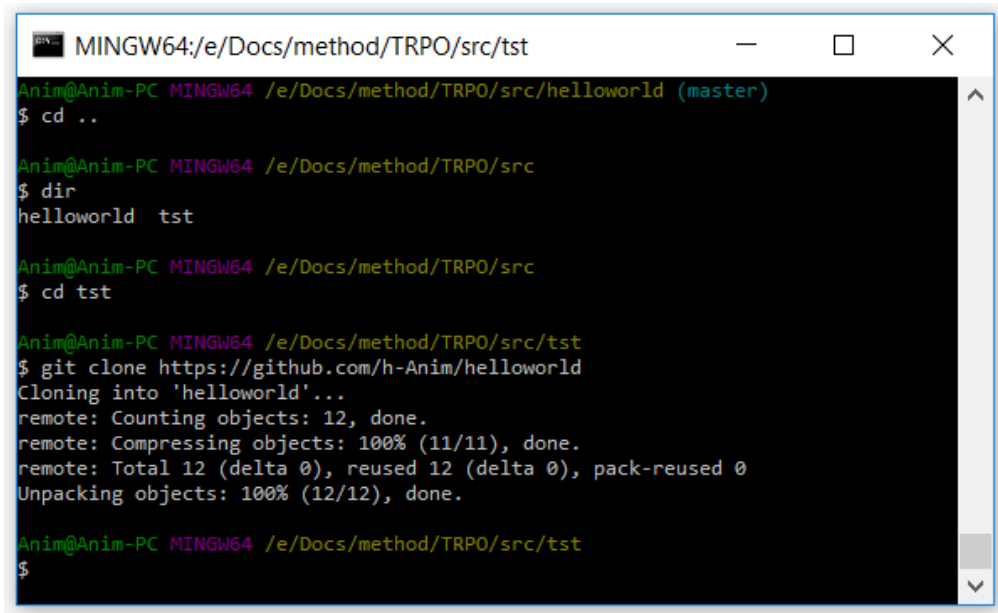
Поскольку репозиторий был только что загружен, он не содержит отличий от локальной версии.

Перед вызовом `git pull` необходимо убедиться, что в локальном репозитории нет незафиксированных изменений. Если таковые имеются, нужно поступить одним из предложенных способов:

- зафиксировать изменения (`git commit`)
- спрятать изменения (`git stash`)
- удалить изменения (`git reset --hard`)

Загрузка существующего репозитория

Можно создать локальный репозиторий на основе уже существующего удалённого репозитория. Для этого используется команда `git clone <url>`:



```

MINGW64:/e/Docs/method/TRPO/src/tst
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ cd ..

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src
$ dir
helloworld  tst

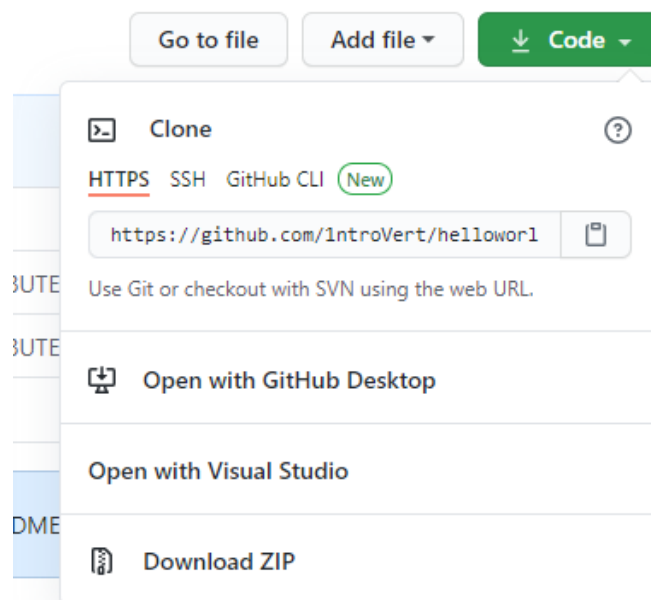
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src
$ cd tst

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/tst
$ git clone https://github.com/h-Anim/helloworld
Cloning into 'helloworld'...
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 0), reused 12 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), done.

Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/tst
$
  
```

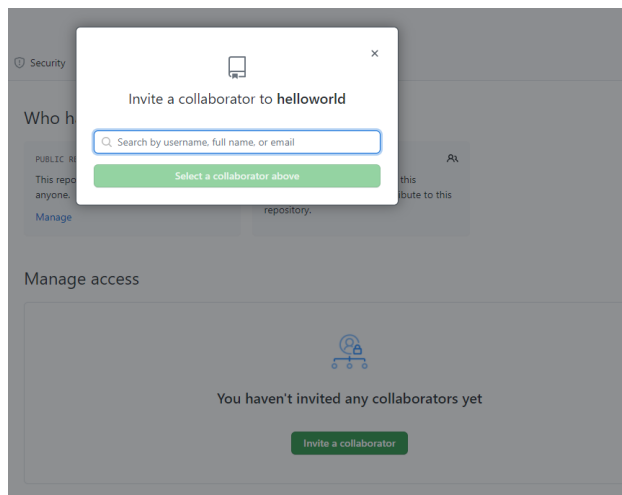
После этого репозиторий скачается в текущую папку.

Также можно загрузить репозиторий и открыть проект автоматически с помощью функции «Open with Visual Studio» или скачать в виде архива с помощью функции «Download ZIP».



Добавление разработчиков на GitHub

Для добавления разработчика к работе над репозиторием перейдите в раздел «Settings», выберите пункт «Manage access» и нажмите на кнопку «Invite a collaborator».



В появившемся поле ввода можно добавить пользователя по его имени на GitHub, полному имени или адресу электронной почты. После чего добавленный пользователь может клонировать и синхронизировать общий репозиторий, а так же совершать коммиты 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 извлечения и применения изменений из удалённого репозитория 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.

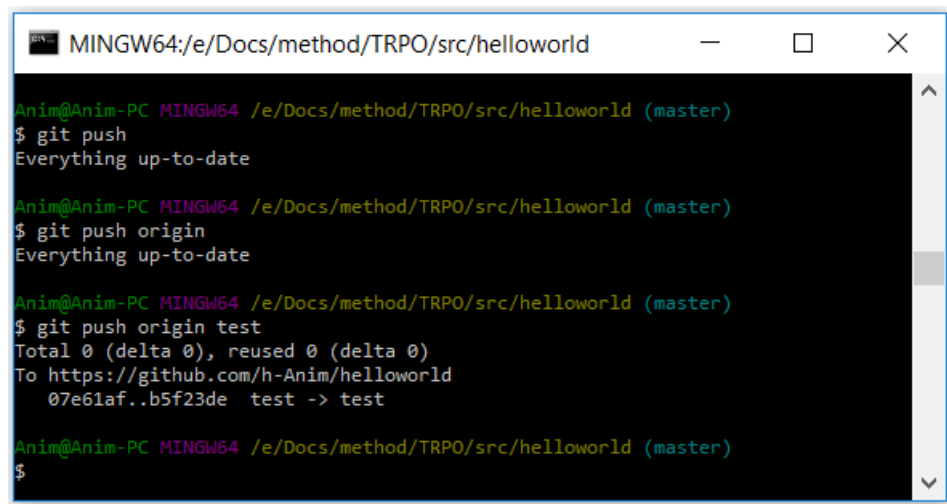
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$
```

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.

Также есть возможность 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 репозитории.

Отправка изменений

Для отправки локальных изменений в удалённый репозиторий используется команда `git push`:



```
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

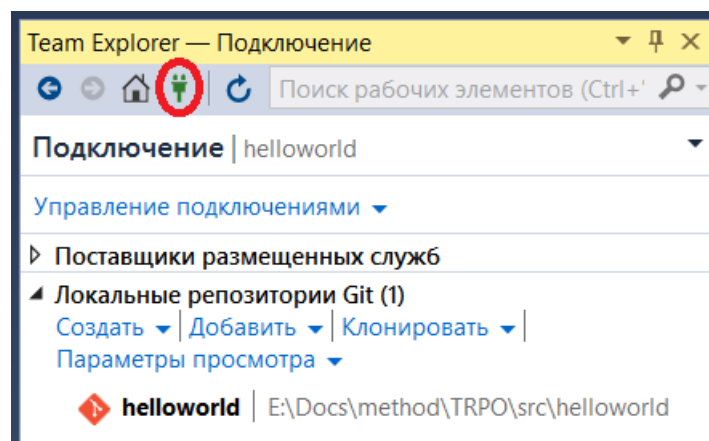
Anim@Anim-PC MINGW64 /e/Docs/method/TRPO/src/helloworld (master)
$ git push origin test
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/h-Anim/helloworld
    07e61af..b5f23de  test -> test

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 2019

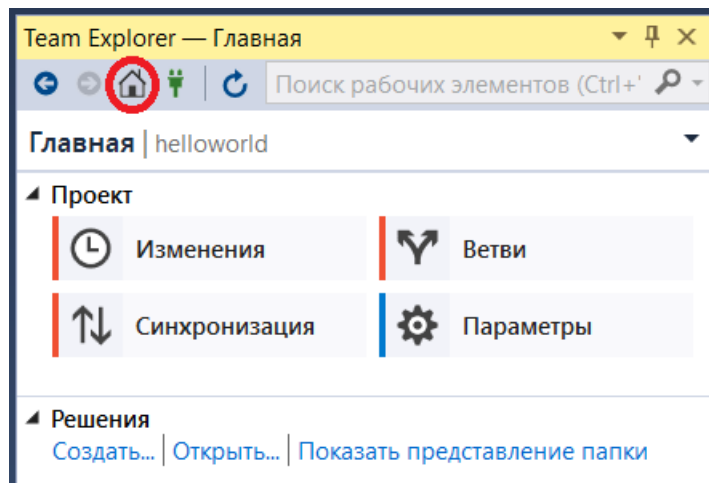
Чтобы начать работу с системой контроля версий в Visual Studio, перейдите к окну «Team Explorer» и нажмите на кнопку «Управление подключениями»:



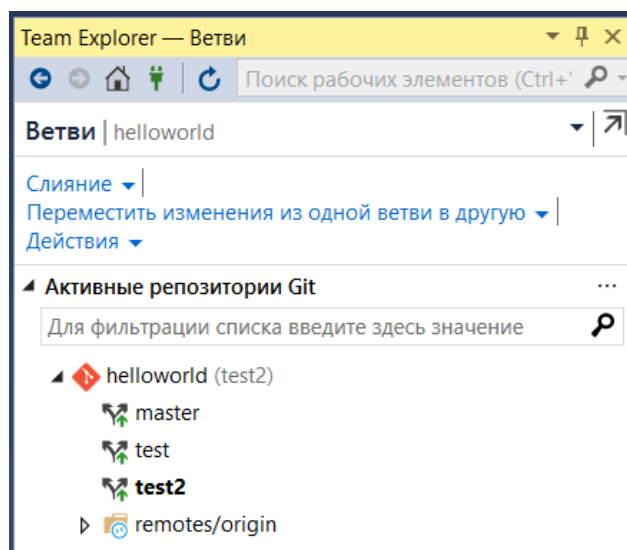
Если окно Team Explorer отсутствует, открыть его можно через меню «Вид -> Team Explorer».

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

Чтобы начать работу с проектом, перейдите в раздел «Главная»:

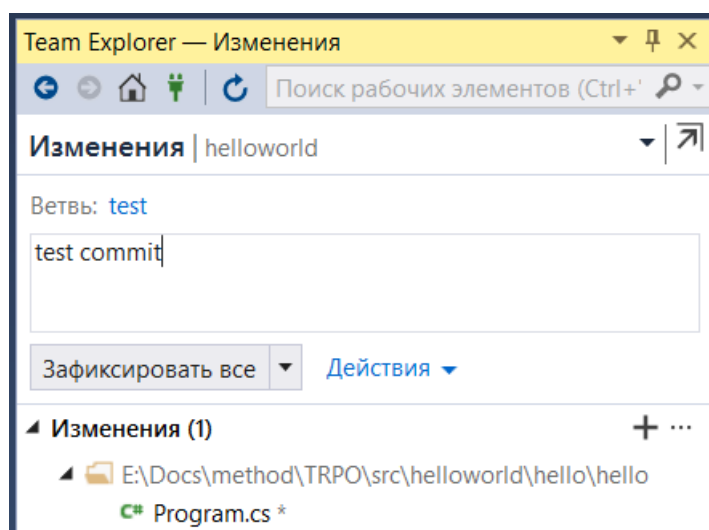


В разделе «Ветви» будут указаны существующие ветви проекта:



Чтобы 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.

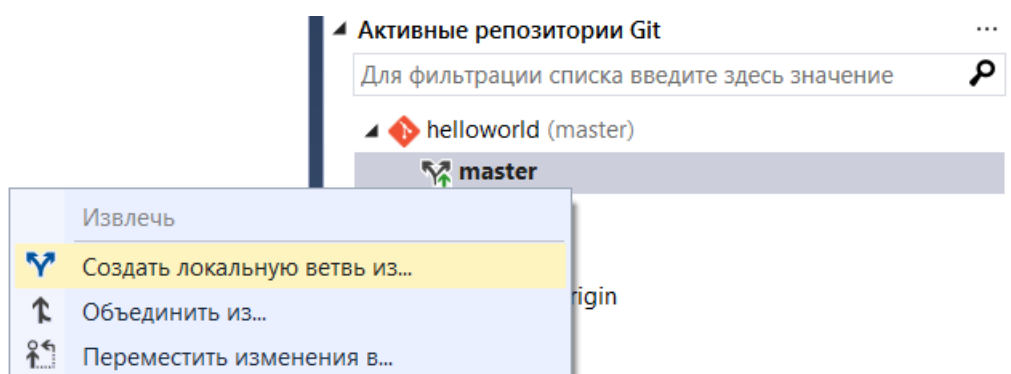
Для фиксации изменений, сделанных в ветви, вернитесь в раздел «Главная» и выберите «Изменения»:



Нажав на «Действия» и выбрав «View History», можно увидеть историю изменений проекта:

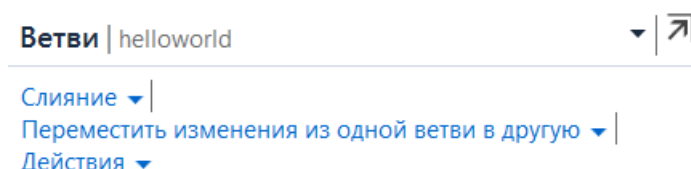
Диаграмма	Идентификатор	Автор	Дата	Сообщение
Локальный журнал				
	dc754415	h-Anim	25.02.2018 23:44:09	test commit
	b5f23de2	h-Anim	19.02.2018 0:46:35	upload
	2377b315	h-Anim	19.02.2018 0:42:22	upload
	07e61af2	h-Anim	19.02.2018 0:19:37	test3
	d7e025c5	h-Anim	19.02.2018 0:18:17	test2
	2bd5f877	h-Anim	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»:

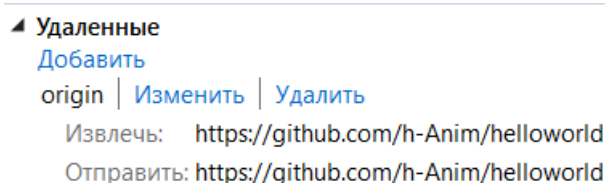


После чего укажите имя новой ветви и нажмите «Создать ветвь».

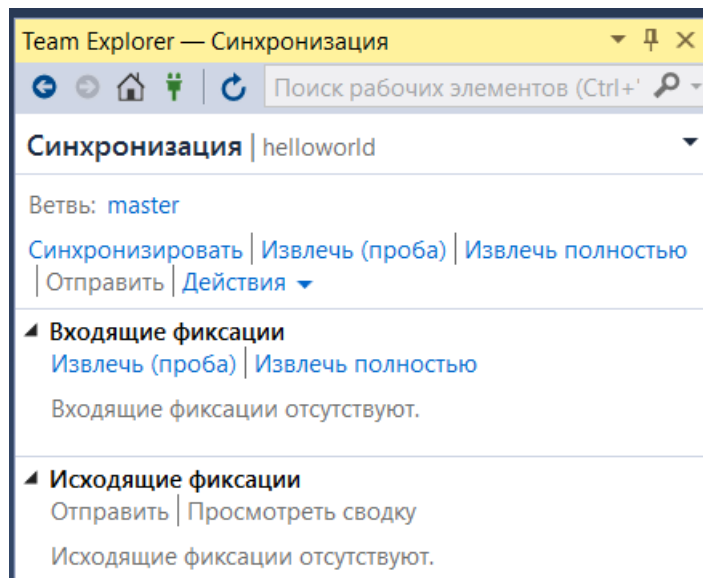
Для слияния ветвей используется соответствующая опция раздела «Ветви»:



Чтобы назначить адрес удалённого репозитория, вернитесь в раздел «Главная», а затем перейдите в раздел «Параметры», пункт «Параметры репозитория»:



Чтобы извлечь (pull) или отправить (push) данные в удалённый репозиторий, перейдите на главную, а затем в раздел «Синхронизация»:



Задание:

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

Используя средства для работы с Git, встроенные в Visual Studio 2019:

1. Создать локальный репозиторий.
2. Создать в локальном репозитории консольное приложение, выводящее в консоль меню выбора, состоящее из следующих пунктов:
 - ввести А
 - ввести В
 - 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:

1. Pro Git by Scott Chacon: <https://git-scm.com/book/ru/v2/> (на русском языке)
2. Интерактивное обучение git: <https://try.github.io/levels/1/challenges/1> (на английском языке)